Temporal Contrastive Pre-Training for Sequential Recommendation

Changxin Tian*
tianchangxin@ruc.edu.cn
School of Information, Renmin University of China

Zihan Lin*
zhlin@ruc.edu.cn
School of Information, Renmin University of China

Shuqing Bian*
bianshuqing@ruc.edu.cn
School of Information, Renmin University of China

Jinpeng Wang
wangjinpeng@meituan.com
Meituan Group

Wayne Xin Zhao†
batmanfly@gmail.com
Gaoling School of Artificial Intelligence, Renmin University of China

ABSTRACT

Recently, pre-training based approaches are proposed to leverage self-supervised signals for improving the performance of sequential recommendation. However, most of existing pre-training recommender systems simply model the historical behavior of a user as a sequence, while lack of sufficient consideration on temporal interaction patterns that are useful for modeling user behavior.

In order to better model temporal characteristics of user behavior sequences, we propose a **T**emporal **C**ontrastive **P**re-training method for **S**equential **Rec**ommendation (**TCPSRec** for short). Based on the temporal intervals, we consider dividing the interaction sequence into more coherent subsequences, and design temporal pre-training objectives accordingly. Specifically, TCPSRec models two important temporal properties of user behavior, *i.e., invariance* and *periodicity*. For invariance, we consider both global invariance and local invariance to capture the long-term preference and short-term intention, respectively. For periodicity, TCPSRec models coarse-grained periodicity and fine-grained periodicity at the subsequence level, which is more stable than modeling periodicity at the item level. By integrating the above strategies, we develop a unified contrastive learning framework with four specially designed pre-training objectives for fusing temporal information into sequential representations. We conduct extensive experiments on six real-world datasets, and the results demonstrate the effectiveness and generalization of our proposed method.

CCS CONCEPTS

• **Information systems → Recommender systems**.

KEYWORDS

Sequential Recommendation, Pre-training, Contrastive Learning

ACM Reference Format:
Changxin Tian, Zihan Lin, Shuqing Bian, Jinpeng Wang, and Wayne Xin Zhao. 2022. Temporal Contrastive Pre-Training for Sequential Recommendation. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management (CIKM '22), October 17–21, 2022, Atlanta, GA, USA*. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3511808.3557468

1 INTRODUCTION

Nowadays, recommender systems have become progressively essential to improve user experience and increase business revenue [25, 26] for various online platforms. In order to make appropriate recommendations, it needs to accurately infer user preference based on sequential user behaviors [33], which is called *sequential recommendation*. In the literature, various models from Markov chains [11, 24], recurrent neural networks [13, 34] and self-attentive networks [16, 20] have been proposed for this task.

More recently, based on the powerful Transformer architecture [30], pre-training based approaches [36, 39, 45] have been shown effective to improve the performance of sequential recommendation. The core idea is to leverage self-supervised signals for improving the model training, based on the correlations from side information or the behavior sequence itself. However, most of existing pre-training approaches in recommender systems model the historical behavior of a user as an interaction sequence, and then reuse or slightly modify the pre-training objectives of language modeling (*e.g.*, masked item prediction). They mainly capture item-level correlation from the entire sequence by fitting the historical sequences, while lack of sufficient consideration on temporal patterns that are specifically useful for modeling user behavior.

To illustrate this, in Figure 1, we present a snapshot of the interaction sequence from a user. From this figure, we can observe the interaction behaviors naturally form several coherent subsequences (separated by a long time interval), and thus simply modeling the sequence as a whole may lose important temporal characteristics of user behavior. Besides, it is interesting to observe that user behaviors are highly affected by temporal information, and such behaviors might recur periodically, *e.g.,* entertainment-based behavior usually

*This work was done during internship at Meituan.
†Corresponding author. Xin Zhao is also with Beijing Key Laboratory of Big Data Management and Analysis Methods and Beijing Academy of Artificial Intelligence.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
CIKM '22, October 17–21, 2022, Atlanta, GA, USA
© 2022 Association for Computing Machinery.
ACM ISBN 978-1-4503-9236-5/22/10…$15.00
https://doi.org/10.1145/3511808.3557468

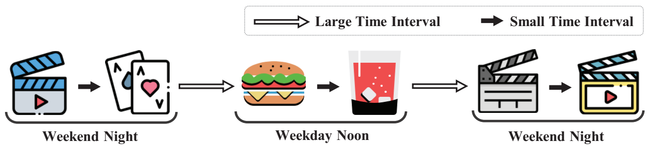


Figure 1: The interaction sequence from a sample user. The entire sequence naturally forms several coherent subsequences corresponding to different time periods.

happens on weekends. Indeed, it has been reported that such finedgrained temporal characteristics are very important for improving sequential recommender systems [8, 18, 32, 38].

In order to enhance the pre-training performance, we aim to better model temporal characteristics of user behavior sequences by developing an improved pre-training approach. Motivated by the observation in Figure 1, we consider dividing the interaction sequence into more coherent subsequences, and design temporal pre-training objectives accordingly. Specifically, we consider modeling two important temporal properties, *i.e., invariance* and *periodicity*. Concretely, *invariance* means that the performance tastes of a user are relatively stable over a time period (*e.g.,* Jack often purchases fast food and movie tickets in a year); and *periodicity* means that the performance tastes of a user are periodically recurrent (*e.g.,* Jack purchases movie tickets every weekend night). A number of studies have shown that these two kinds of temporal properties are essential to modeling sequential user behaviors [5, 18, 32, 38]. However, to the best of our knowledge, no work has studied how to utilize or characterize them in a pre-training framework.

To address the aforementioned issue, we propose a **T**emporal **C**ontrastive **P**re-training method for **S**equential **Rec**ommendation (denoted as **TCPSRec**). We extend the previous pre-training approaches by explicitly modeling invariance and periodicity in user behaviors. These two kinds of temporal correlations are modeled by specially designed self-supervised objectives. For invariance, we consider both *global invariance* within a sequence and *local invariance* within a subsequence, in order to capture the long-term preference and short-term intention, respectively. For periodicity, TCPSRec models both *coarse-grained periodicity* and *fine-grained periodicity* at the subsequence level rather than the item level. The intuition behind this is that subsequences correspond to the user's intentions over a period of time rather than a specific behavior, which is more stable for periodicity modeling. Considering the above motivations, we develop a unified contrastive learning framework with four specially designed pre-training objectives for global invariance, local invariance, coarse-grained periodicity, and finegrained periodicity, respectively. To conduct effective contrastive learning, a key point of our approach is that we construct contrastive objectives based on the temporal patterns rather than simple random sampling, which is more efficient and meaningful for sequential recommendation. Specifically, we take items in the same sequence and those in the same subsequence as weak positives and strong positives, respectively, to model the invariance; and then take subsequences with the same coarse-grained periodic features and those with the same fine-grained periodic features as weak positives and strong positives, respectively, to model the periodicity. By optimizing these four pre-training objectives, we can effectively capture the temporal interaction patterns in the learned sequential representations. Finally, the pre-trained parameters will be fed into sequential recommender systems and then fine-tuned according to the recommendation task. With our proposed pre-training method, extensive experiments on several large datasets show that it can effectively improve the performance of sequential recommender systems.

Our contributions can be summarized threefold:

• To the best of our knowledge, we present the first temporal contrastive pre-training approach by modeling invariance and periodicity, for sequential recommender systems, which is beyond token-level prediction as in language modeling.

• We propose four specially designed contrastive learning objectives to better model temporal characteristics of user behaviors at both item and subsequence levels.

• Extensive experiments are conducted on one industry dataset and five public datasets, demonstrating the effectiveness and generalization of our proposed method.

2 PRELIMINARY

In this section, we introduce the background for this work.

Task Definition. We consider the sequential interaction scenario between users and items, where users interact with the items at different timestamps. Formally, given user set $\mathcal{U} = \{u\}$ and item set $\mathcal{I} = \{i\}$, the observed user-item interaction data is denoted as a set $\{(u, i, t)\}$, where the instance (u, i, t) represents that user u interacts with item i at time t. According to the interaction timestamps, for a user u, we sort the interactions to form a chronological interaction sequence $S_u = [i_1, i_2, \cdots, i_n]$, where n is the number of user u's interactions, and i_p is the item at the p-th interaction. Considering the time intervals between any two adjacent interactions are different, we divide the entire interaction sequence into subsequences when a time interval is larger than a threshold Δ (*e.g.,* an hour). Thus, the interaction sequence S_u can be written as $S_u = [z_1^u, z_2^u, \cdots, z_{n'}^u]$, where n' ($n' \leq n$) is the number of user u's interaction subsequences and z_p^u is the p-th interaction subsequence of user u. For convenience, we drop the index of u from the subsequences. Based on the above notations, given the user interaction sequence S_u, the task of sequential recommendation aims to predict the most possible item that the user u will interact with at the $(n + 1)$-th step. Note that following the previous setting [3, 16, 36, 45], the timestamp data is used only to determine the the sequential order of interactions when performing the recommendation.

Base Model. For sequential recommendation, we adopt the Transformer [30] architecture as the base model to model user interaction sequences [16], consisting of an embedding layer, multi-head self-attention layers, and a point-wise feed-forward layer. In the embedding layer, an item embedding matrix $\mathbf{E} \in \mathbb{R}^{|\mathcal{I}| \times d}$ is adopted to project high dimensional one-hot item representations to low dimensional dense vectors. Besides, we incorporate a learnable position encoding matrix $\mathbf{P} \in \mathbb{R}^{n \times d}$ to represent the position information of sequence. Given an n-length item sequence, the input embedding matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$ of the sequence can be obtained by looking up these two embedding matrices and summing them. Then, the multi-head self-attention layers extract information selectively from different representation subspaces as follows:

$$\begin{aligned}\text{MultiHead}\left(\mathbf{H}^{(l)}\right) &= [head_1, head_2, \ldots, head_h]\,\mathbf{W}^O,\\ head_i &= \text{Attention}\left(\mathbf{H}^{(l)}\mathbf{W}_i^Q, \mathbf{H}^{(l)}\mathbf{W}_i^K, \mathbf{H}^{(l)}\mathbf{W}_i^V\right),\end{aligned} \tag{1}$$

where h is the number of heads, $\mathbf{W}_i^Q$, $\mathbf{W}_i^K$ and $\mathbf{W}_i^V$ are learnable projection matrixs for the i-th head. $\mathbf{H}^{(l)}$ is the input for the l-th layer, where $\mathbf{H}^{(0)}$ is set to $\mathbf{X}$. And the attention function is implemented as scaled dot-product attention [16]. Furthermore, the point-wise feed-forward layer is applied to endow the model with non-linearity. The output of each multi-head self-attention layer is processed separately as follows:

$$\begin{aligned}\mathbf{F}^{(l)} &= \left[\text{FFN}(\mathbf{F}_1^{(l)})^\top; \cdots; \text{FFN}(\mathbf{F}_n^{(l)})^\top\right],\\ \text{FFN}(x) &= (\text{ReLU}\,(\boldsymbol{x}\mathbf{W}_1 + \boldsymbol{b}_1))\,\mathbf{W}_2 + \boldsymbol{b}_2,\end{aligned} \tag{2}$$

where $\mathbf{W}_1, \boldsymbol{b}_1, \mathbf{W}_2$ and $\boldsymbol{b}_2$ are trainable parameters. In the final layer of the Transformer module, we denote the output at step t as $\boldsymbol{m}_t$, which is the final representation of the item at step t. Finally, we use the output $\boldsymbol{m}_t$ to calculate the user's preference score for the item i at the step $(t+1)$ as follows:

$$\text{R}\,(i_{t+1} = i \mid i_{1:t}) = \boldsymbol{e}_i^\top \boldsymbol{m}_t, \tag{3}$$

where $\boldsymbol{e}_i$ is item i's embedding from the item embedding matrix $\mathbf{E}$. Then, the obtained ranking scores can be used for recommendation.

3 METHODOLOGY

In this section, we introduce the proposed **T**emporal **C**ontrastive **P**re-training method for **S**equential **Rec**ommendation (**TCPSRec**).

3.1 Overview

The proposed TCPSRec consists of four pre-training objectives, which are developed in the unified form of contrastive learning. Specifically, TCPSRecmodel two temporal properties of user behavior (*i.e., invariance* and *periodicity*) at item level and subsequence level, respectively. The model overview is illustrated in Figure 2.

For invariance, it means that the preferences of a user are stable over a time period. In order to comprehensively model the invariance, we consider both global invariance within a sequence and local invariance within a subsequence to capture the long-term preference and short-term intention, respectively. For periodicity, it means that the intentions of a user are regularly recurrent. In this work, we model both coarse-grained periodicity and fine-grained periodicity at the subsequence level, respectively. The fine-grained periodicity reflects more detailed and complex periodic characteristics than coarse-grained periodicity. By optimizing these four contrastive objectives, we can effectively capture temporal interaction patterns in sequential representations.

3.2 Item Level Invariant Modeling

The invariance of user behavior means that the user's preferences or intentions are relatively stable over a period of time. In this section, we present two item-level pre-training objectives to model global invariance and local invariance, which correspond to the long-term preference and short-term intention, respectively.

3.2.1 ***Global Invariant Modeling.*** Intuitively, the long-term preferences of a user are reflected in all of his/her historical behaviors, *e.g.,* Jack often purchases fast food and movie tickets in a year. We call this property the global invariance of user behavior, which is a global property reflected by the entire behavior sequence. According to the global invariance, items that the same user interacts with should be related to each other, *i.e.,* items in the same interaction sequence should be related to each other.

To model global invariance, the basic idea is to enforce the relatedness between items within each interaction sequence. In the form of contrastive learning, for an item, we take items in the same sequence as its positives and other items as negatives, as shown in Figure 2(a). However, the time complexity of the above approach is prohibitive because each item has multiple positives for contrastive learning. Thus, for a target item, we take the sequence to which it belongs as its positive in implementation, where the representation of the whole sequence can be obtained by mean pooling operation. Based on InfoNCE [29], we propose the global invariant modeling (GIM) objective to minimize the distance between them:

$$\mathcal{L}_{GIM} = \sum_{u\in\mathcal{U}}\sum_{i\in S_u} -\log\frac{\exp(\boldsymbol{e}_i^\top \boldsymbol{e}_{S_u}/\tau)}{\sum_{u'\in\mathcal{U}}\exp(\boldsymbol{e}_i^\top \boldsymbol{e}_{S_{u'}}/\tau)}, \tag{4}$$

where τ is the temperature hyper-parameter, S_u is the interaction sequence of a user u, and $\boldsymbol{e}_{S_u} = \sum_{i\in S_u}\boldsymbol{e}_i/|S_u|$ is the representation of the sequence S_u with length $|S_u|$.

3.2.2 ***Local Invariant Modeling.*** In real-world scenarios, users often interact with highly similar items in a short period, *e.g.,* clicking on different types of movie tickets intensively when they want to see a movie. This is the local invariance within each behavior subsequence, reflecting the user's short-term intentions. Generally speaking, based on the local invariance, one item should be more related to items in the same interaction subsequence than other non-subsequence items in the same sequence. However, such local correlations have often been neglected in previous studies.

Therefore, we further model local invariance within a subsequence. Specifically, for an item, we take items in the same interaction subsequence as its positives and the remaining items in the same sequence as negatives to model the local invariance, as shown in Figure 2(b). Considering the efficiency issue, for a target item, we further propose to take the corresponding subsequence as its positive. Formally, given the representation of subsequence obtained by mean pooling operation, the local invariant modeling (LIM) objective can be defined as follows:

$$\mathcal{L}_{LIM} = \sum_{u\in\mathcal{U}}\sum_{z_u\in S_u}\sum_{i\in z_u} -\log\frac{\exp(\boldsymbol{e}_i^\top \boldsymbol{e}_{z_u}/\tau)}{\sum_{z'_u\in S_u}\exp(\boldsymbol{e}_i^\top \boldsymbol{e}_{z'_u}/\tau)}, \tag{5}$$

where $\boldsymbol{e}_{z_u} = \sum_{i\in z_u}\boldsymbol{e}_i/|z_u|$ is the representation of the subsequence z_u and $|z_u|$ is the length of the subsequence z_u. By treating some of the positives in the GIM objective (defined in Eq. (4)) as negatives in the LIM objective (defined in Eq. (5)), we can capture the relative correlation of global invariant and local invariant.

3.3 Subsequence Level Periodicity Modeling

The periodicity of user behaviors means that the performance tastes of a user are regularly recurrent. Here, we present two pre-training

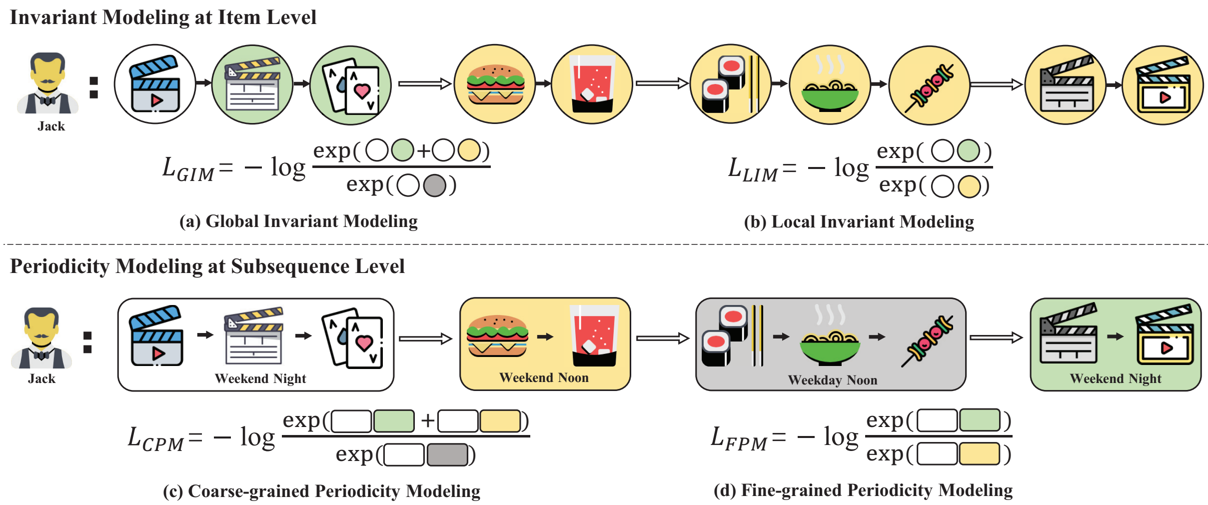


Figure 2: The overview of the proposed TCPSRec. In the pre-training stage, TCPSRec consists of four temporal contrastive objectives based on coherent subsequences. To illustrate each contrastive objective, we take the first item/subsequence of the sequence as the target, and then mark its positives and negatives in different colors. In particular, negatives in the Global Invariant Modeling objective (represented by gray circles) are items sampled from other interaction sequences.

objectives at subsequence level to model coarse-grained periodicity and fine-grained periodicity of user behavior, respectively.

3.3.1 ***Coarse-grained Periodicity Modeling.*** Generally speaking, the periodicity means that the similar user intention recurs regularly through the timeline. Instead of modeling item-level periodicity, it is more common to consider *intention periodicity*. For example, a user might watch movies every weekend, but rarely watches the same movie multiple times. In this work, we consider intention as the overall purpose for the activities in a subsequence, and devise subsequence-level contrastive learning objectives to model the periodicity of user behavior.

Since users' behavior often reflects daily or weekly patterns, we enforce the correlations between the subsequences with the same periodic features (*e.g.,* both on weekends or evenings). These two sets of periodic features are denote by $\mathcal{W} = \{Mon, \ldots, Sun\}$ and $\mathcal{D} = \{1, \ldots, 24\}$, respectively. Specifically, we can group the subsequences of the same users into either 7 or 24 coarse-grained periodic clusters, according to the timestamp of the first item in each subsequence. Based on the above periodic clusters, for a subsequence in a sequence, we take subsequences from the same periodic cluster as its positives and other items as negatives, as shown in Figure 2(c). In implementation, we treat the representation of a subsequence and the mean representation of all its positives as a positive pair to improve the efficiency. Formally, the coarse-grained periodicity modeling (CPM) objective can be defined as follows:

$$\mathcal{L}_{CPM} = \sum_{u \in \mathcal{U}} \sum_{z_u \in S_u} \sum_{c \in C_{z_u}} -\log \frac{\exp(\boldsymbol{e}_{z_u}^{\top} \boldsymbol{e}_c^u / \tau)}{\sum_{c' \in C} \exp(\boldsymbol{e}_{z_u}^{\top} \boldsymbol{e}_{c'}^u / \tau)}, \tag{6}$$

where $C = \mathcal{W} \cup \mathcal{D}$ is the set of all coarse-grained periodic clusters, C_{z_u} is the coarse-grained cluster set of subsequence z_u. For a user u, $\boldsymbol{e}_c^u = \sum_{z_u \in S_u} \boldsymbol{e}_{z_u} / n'$ is the mean representation of all subsequences from the periodic cluster a, and n' is the number of user u's interaction subsequences. Note that other periodical patterns, such as one month and a quarter, can also be modeled in a similar way.

3.3.2 ***Fine-grained Periodicity Modeling.*** The above pre-training objectives capture coarse-grained periodicity, while we note that combining features of different periodicity patterns can yield more fine-grained periodic features (*e.g.,* weekend night). Intuitively, fine-grained periodicity can reflect a more accurate modeling of user behavior characteristics. To model the fine-grained periodicity, for a subsequence in a sequence, we take subsequences with the same fine-grained periodic features as positives and the subsequences with the same coarse-grained periodic features as negatives, as shown in Figure 2(d). Similar to Section 3.3.1, we adopt the mean pooling to aggregate multiple positives into one positive for contrastive learning in implementation. Thus, the fine-grained periodicity modeling (FPM) objective can be formally given as follows:

$$\mathcal{L}_{FPM} == \sum_{u \in \mathcal{U}} \sum_{z_u \in S_u} \sum_{a \in \mathcal{A}_{z_u}} -\log \frac{\exp(\boldsymbol{e}_{z_u}^{\top} \boldsymbol{e}_a^u / \tau)}{\sum_{c \in C_{z_u}} \exp(\boldsymbol{e}_{z_u}^{\top} \boldsymbol{e}_c^u / \tau)}, \tag{7}$$

where $\mathcal{A} = \mathcal{W} \times \mathcal{D}$ is the set of all fine-grained periodic clusters obtained by considering the cartesian product of $\mathcal{W}$ and $\mathcal{D}$, and $\mathcal{A}_{z_u}$ denotes the fine-grained periodic features of subsequence z_u. For a user u, $\boldsymbol{e}_a^u$ is the mean of the representations of the subsequences with the fine-grained periodic feature a. Note that similar to the invariant modeling, subequences with the same coarse-grained features but different fine-grained features are treated as positives in CPM objective (defined in Eq. (6)) and negatives in FPM objective (defined in Eq. (7)). Combining Eq. (6) and Eq. (7), we can capture hierarchical periodicity and the corresopnding correlations at the subsequence level.

Table 1: Comparison between the proposed TCPSRec and existing representative pre-training recommendation algorithms. Due to space limitation, "L.P." is the abbreviations for "Level Pre-training".

Methods	Item L.P.	Subsequence L.P.	Sequence L.P.	Scenario	Side Information	Data Augmentation
S^3-Rec [45]	✔	✔	✔	Sequential	Item Attribute	Random Item Masking/Cropping
CP4Rec [36]	✗	✗	✔	Sequential	None	Random Item Masking/Reordering/Cropping
UPRec [35]	✔	✗	✗	Sequential	User Attribute	Random Item Masking
CLUE [4]	✗	✗	✔	Sequential	None	Random Item Masking/Cropping/Dropout Mask
PMGT [21]	✔	✗	✗	Graph	Item Attribute	Random Item Masking/Subgraph Sampling
TCPSRec	✔	✔	✔	Sequential	Temporal Information	Naturally Sequence Dividing

3.4 Training Strategy

As introduced before, we adopt the two-stage training scheme, *i.e.*, pre-training and fine-tuning, to train the recommender system.

Pre-training Stage. Firstly, we jointly optimize the four contrastive objectives to pre-train the recommender system. Formally, the overall pre-training objective is the weighted sum of the above four objectives, which can be defined as follows:

$$\mathcal{L}_{Pre} = \lambda_1 \mathcal{L}_{GIM} + \lambda_2 \mathcal{L}_{LIM} + \lambda_3 \mathcal{L}_{CPM} + \lambda_4 \mathcal{L}_{FPM}, \tag{8}$$

where $\lambda_1, \lambda_2, \lambda_3$ and λ_4 are the hyper-parameters. After pre-training, the parameters will be fed into the neural models for fine-tuning.

Fine-tuning Stage. At this stage, we utilize the pre-trained parameters to initialize the parameters of the recommender system, and then adopt the binary cross-entropy loss for each user u at each time step t to fine-tune the parameters as previous studies [16]:

$$\mathcal{L}_{Rec} = \sum_{u \in \mathcal{U}} \sum_{i \in S_u} -\left(\log \sigma \left(\boldsymbol{e}_i^\top \boldsymbol{m}_t \right) + \sum_{j \notin S_u} \log \left(1 - \sigma \left(\boldsymbol{e}_j^\top \boldsymbol{m}_t \right) \right) \right), \tag{9}$$

where $\boldsymbol{e}_i$ is item i's embedding, as defined in Section 2.

3.5 Comparison and Discussion

In this part, we compare the proposed TCPSRec with existing pre-training recommender systems, and then analyze its complexity.

3.5.1 ***Comparison with Existing Pre-training Recommender Systems.*** Recent studies [4, 19, 21, 27, 35, 39–41, 45] propose to enhance the sequential recommender systems by conducting sequence-level pre-training. Compared with existing methods, our method TCPSRec has two major technical contributions. First, TCPSRec considers temporal correlations at the subsequence level when devising the self-supervised learning tasks at the pre-training stage, while previous works mainly focus on the sequence-level correlation, without considering temporal information. In general, timestamps reflect fined-grained temporal information of interaction, and are much more readily available than attributes of items. Second, existing methods mainly reuse the language modeling techniques for deriving the self-supervised signals, which may lack consideration of the temporal information for the recommendation task. As a comparison, TCPSRec derives self-supervision signals based on the naturally divided behavior subsequences, and constructs pre-training tasks according to the intrinsic temporal properties of user behavior. Besides, our approach is more efficient than augmentation based pre-training methods [36], since we don't introduce any sequence augmentation strategies. We summarize the key differences between the proposed method and several representative pre-training recommender systems in Table 1.

Table 2: Statistics of the datasets

Datasets	Meituan	Yelp	Beauty	Sports	Toys	ML-1M
# Users	22,484	30,494	22,363	35,598	19,412	6,040
# Items	31,303	20,061	12,101	18,357	11,924	3,953
# Actions	588,406	317,078	198,502	296,337	167,597	1,000,000
# Actions / User	26.2	10.4	8.9	8.3	8.6	165.6
# Actions / Item	18.8	15.8	16.4	16.1	14.1	253.0
Sparsity	99.92%	99.95%	99.93%	99.94%	99.93%	95.81%

3.5.2 ***Time and Space Complexity.*** The key points of TCPSRec are the four contrastive objectives defined in Eq. (4)-Eq. (7). Assuming that the average length of user interaction sequences is I and that, on average, each interaction sequence is divided into B subsequences, we sample N negative samples in each proposed contrastive objective to improve efficiency. Then, the time complexity of the four contrastive objectives can be roughly estimated as $O(|\mathcal{U}| \cdot N \cdot d \cdot (I + B))$, where d is the dimension of embedding vector. Besides, TCPSRec plays a role in the pre-training stage and does not introduce additional complexity when fine-tuning on the sequential recommendation task, which is an essential advantage over existing temporal recommendation methods [6, 8, 18, 32, 38].

4 EXPERIMENTS

In this section, we conduct a series of experiments to demonstrate the effectiveness of the proposed TCPSRec.

4.1 Experimental Setup

4.1.1 ***Dataset.*** We adopt one industry dataset and five public recommendation datasets for evaluation:

- **Meituan**[1] is an industry dataset that consists of transaction records from the Meituan platform in Beijing City from May 2021 to October 2021. This dataset includes a variety of items, including restaurants, hotels, movies, and others.
- **Yelp**[2] is a business recommendation dataset in which the businesses in the catering industry are reviewed as the items. In this work, we use the transaction records after January 1st, 2019.
- **Amazon Beauty, Sports and Toys**[3] are obtained from Amazon review dataset [12]. In our experiments, we select three subcategories: "Beauty", "Sports and Outdoors", and "Toys and Games".

[1] https://www.meituan.com
[2] https://www.yelp.com/dataset
[3] https://jmcauley.ucsd.edu/data/amazon/

• **MovieLens-1M** [4] has been widely used in previous studies on recommendation task [10], which contains movie ratings.

Following previous works [3, 36, 45], we group interaction records by user for all datasets and sort them in ascending order by interaction timestamp to obtain the user interaction sequence. Next, we divide the interaction sequence into several subsequences according to the time intervals larger than one hour. To ensure data quality, we keep the 5-core datasets, filtering out items and users with less than five interaction records. The maximum length of the interaction sequence is 100 for Meituan dataset and 50 for other datasets. The statistics of the datasets are summarized in Table 2.

4.1.2 ***Baseline Models.*** We compare the proposed TCPSRec with the following baseline methods to verify its effectiveness:

(1) **PopRec** is a non-personalized approach that recommends the most popular items with the most interactions for each user.

(2) **GRU4Rec** [13] applies GRU to model user interaction sequence for recommendation. In our implementation, we use embedding vectors instead of one-hot vectors to represent the items.

(3) **Caser** [28] is a CNN-based method that captures higher-order patterns for sequential recommendations by applying horizontal and vertical convolution operations.

(4) **SASRec** [16] is a single-directional self-attention based method that uses the multi-head attention mechanism to perform sequential recommendation.

(5) **TiSASRec** [18] is an extension of SASRec, which uses time interval information to improve the self-attention mechanism.

(6) $\mathbf{S^3}$**-Rec**$_{\mathbf{MIP}}$ [45] applies the mutual information maximization principle to fuse the context data and sequence data. Because the attribute information about items is not available in this work, the Mask Item Prediction (MIP) variant is used here.

(7) **CL4SRec** [36] fuses contrastive self-supervised learning with the transformer-based sequential recommendation model.

(8) **ICLRec** [3] models latent intent factors in user interactions and fuses them into a sequential recommendation model through contrastive learning and EM framework.

To ensure fairness, we use the cross-entropy loss function for the next item prediction task in all baseline methods.

4.1.3 ***Evaluation Metrics.*** We adopt the leave-one-out strategy to evaluate the performance of each method. Concretely, for each user interaction sequence, the last item is used as the test data, the item before the last one is used as the validation data, and the remaining data is used for training. Following related works [3, 36, 45], we adopt top-K metrics Hit Ratio (HR@K), Normalized Discounted Cumulative Gain (NDCG@K) to evaluate the performance, where K is set to be 5, 10 and 20. In addition, some previous works [16, 45] use sampling metrics to rank only the relevant items with smaller sets of random items, which may lead to unreliable experimental results [42]. For this reason, we perform a full ranking with all item candidates and report the average results across all the users.

4.1.4 ***Implementation Details.*** For the baseline methods, we either implement them with the provided source code, or reproduce them with `RecBole`[5] [43, 44]. For a fair comparison, we set the embedding size and batch size to 64 and 1024, respectively, and carefully tune the hyperparameters according to the original papers. We adopt the Adam optimizer with its default parameter setting, and apply an early stopping strategy with the patience of 10 epochs. For the proposed TCPSRec, the model learning consists of the pre-training stage and fine-tuning stage, and the learned parameters at the pre-training stage are used to initialize the embedding layers of base model at the fine-tuning stage. we set the number of the self-attention blocks and the attention heads as 2. We tune the learning rate in [2e-5, 2e-3], the temperature τ in {0.05, 0.1, 0.2}, and the pre-trainng epoch in {10, 20, 50}. we set the weights $\lambda_1 = \lambda_2 = \lambda_3 = \lambda_4$, and tune them in {1e-7, 1e-6, 1e-5, 1e-4}.

[4]https://grouplens.org/datasets/movielens/1m/
[5]https://github.com/RUCAIBox/RecBole

4.2 Overall Performance Comparison

The comparison results between TCPSRec and baseline methods are shown in Table 3. This table shows that TCPSRec achieves the best performances on all datasets and outperforms the state-of-the-art methods CL4SRec and ICLRec.

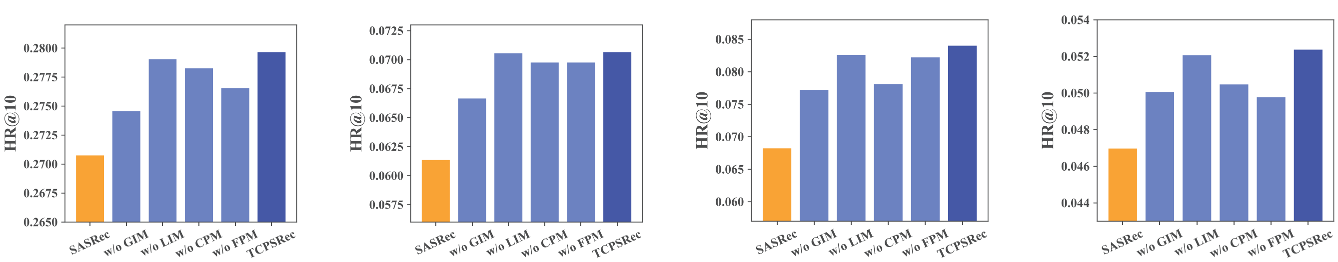
Specifically, the non-personalized method PopRec obtains the worst results among all baseline methods, since the personalized preference of users are neglected. For standard sequential recommendation baseline methods, SASRec performs better than GRU4Rec and Caser, which shows the effectiveness of the attention mechanism for modeling sequential patterns. Nevertheless, the performance improvement of TiSASRec over the original SASRec is neither significant nor inconsistent. However, although TiSASRec considers the time interval information in the sequential recommendation task, its performance improvement over SASRec is not significant or stable. In particular, TiSASRec even performs worse than SASRec models on the Meituan dataset. This finding indicates that simply optimizing the entire model through the final recommendation objective can't fully leverage temporal information.

For sequential recommendation methods using self-supervised learning, overall, CL4SRec and ICLRec perform better than S^3-Rec$_{\text{MIP}}$. A possible reason is that S^3-Rec is designed to fuse sequence data with auxiliary contextual information, such as item category labels. In the absence of contextual information, it may be difficult for S^3-Rec to derive effective self-supervised signals. Both CL4SRec and ICLRec adopt the multi-task learning strategy to leverage the self-supervised signals. These two methods consistently outperform the base model SASRec on six datasets, demonstrating the effectiveness of self-supervised learning for improving the sequential recommendation performance. However, CL4SRec and ICLRec conduct self-supervised signals through data augmentation and clustering respectively, which require a higher time complexity. In addition, though both CL4SRec and ICLRec adopt contrastive learning, neither of them can capture temporal correlations. The experimental results further show the usefulness of temporal information for the sequential recommendation task.

By comparing the proposed TCPSRec with baselines, we conclude that TCPSRec consistently yields the best performance on six datasets. Our approach can effectively fuse temporal information to improve the sequential representations by optimizing four contrastive pre-training objectives. It devises specific pre-training tasks based on both invariance and periodicity, which are more effective to capture sequential behavior characteristics.

Table 3: Performance comparison of different methods on top-N recommendation. The best result is bolded and the runner-up is underlined. "$*$" indicates the statistical significance for $p < 0.01$ compared to the best baseline.

Dataset	Metric	PopRec	GRU4Rec	Caser	SASRec	TiSASRec	S^3-Rec$_{MIP}$	CL4SRec	ICLRec	TCPSRec	Improv.
Meituan	HR@5	0.0192	0.1792	0.1830	0.2110	0.2033	0.2123	0.2158	0.2111	**0.2229***	+3.30%
	HR@10	0.0281	0.2320	0.2433	0.2708	0.2611	0.2713	0.2707	0.2692	**0.2797***	+3.30%
	HR@20	0.0521	0.2918	0.3211	0.3305	0.3301	0.3321	0.3318	0.3301	**0.3424***	+3.20%
	NDCG@5	0.0122	0.1309	0.1349	0.1553	0.1547	0.1546	0.1554	0.1532	**0.1663***	+7.00%
	NDCG@10	0.0150	0.1479	0.1526	0.1745	0.1735	0.1698	0.1751	0.1744	**0.1846***	+5.40%
	NDCG@20	0.0210	0.1629	0.1713	0.1896	0.1888	0.1887	0.1915	0.1899	**0.2004***	+4.60%
Yelp	HR@5	0.0059	0.0315	0.0342	0.0431	0.0444	0.0441	0.0448	0.0455	**0.0493***	+8.35%
	HR@10	0.0097	0.0492	0.0523	0.0614	0.0619	0.0616	0.0620	0.0632	**0.0707***	+11.87%
	HR@20	0.0171	0.0773	0.0821	0.0886	0.0891	0.0886	0.0880	0.0912	**0.1043***	+14.36%
	NDCG@5	0.0039	0.0216	0.0243	0.0326	0.0329	0.0331	0.0334	0.0332	**0.0359***	+7.49%
	NDCG@10	0.0051	0.0273	0.0315	0.0385	0.0388	0.0381	0.0389	0.0383	**0.0428***	+10.03%
	NDCG@20	0.0070	0.0343	0.0399	0.0454	0.0455	0.0442	0.0455	0.0461	**0.0513***	+11.28%
Beauty	HR@5	0.0071	0.0422	0.0453	0.0483	0.0491	0.0489	0.0495	0.0499	**0.0566***	+13.43%
	HR@10	0.0102	0.0640	0.0677	0.0683	0.0694	0.0691	0.0716	0.0732	**0.0841***	+14.89%
	HR@20	0.0173	0.0921	0.0952	0.0970	0.1007	0.0997	0.1030	0.1062	**0.1176***	+10.73%
	NDCG@5	0.0046	0.0299	0.0315	0.0341	0.0336	0.0336	0.0346	0.0341	**0.0352***	+5.70%
	NDCG@10	0.0056	0.0369	0.0389	0.0405	0.0408	0.0406	0.0417	0.0408	**0.0440***	+5.50%
	NDCG@20	0.0074	0.0440	0.0472	0.0477	0.0488	0.0481	0.0496	0.0486	**0.0524***	+5.70%
Sports	HR@5	0.0057	0.0254	0.0274	0.0306	0.0311	0.0286	0.0312	0.0316	**0.0335***	+6.01%
	HR@10	0.0091	0.0387	0.0411	0.0470	0.0498	0.0441	0.0501	0.0510	**0.0524***	+2.75%
	HR@20	0.0161	0.0581	0.0632	0.0695	0.0723	0.0662	0.0725	0.0732	**0.0773***	+5.60%
	NDCG@5	0.0036	0.0167	0.0165	0.0168	0.0172	0.0166	0.0171	0.0174	**0.0184***	+5.74%
	NDCG@10	0.0047	0.0210	0.0201	0.0221	0.0225	0.0217	0.0225	0.0233	**0.0244***	+4.72%
	NDCG@20	0.0064	0.0259	0.0255	0.0278	0.0283	0.0269	0.0281	0.0283	**0.0307***	+8.48%
Toys	HR@5	0.0054	0.0450	0.0466	0.0620	0.0622	0.0602	0.0621	0.0631	**0.0671***	+6.34%
	HR@10	0.0075	0.0645	0.0691	0.0910	0.0906	0.0883	0.0894	0.0913	**0.0967***	+5.91%
	HR@20	0.0135	0.0888	0.0922	0.1221	0.1125	0.1193	0.1217	0.1236	**0.1324***	+7.12%
	NDCG@5	0.0040	0.0324	0.0333	0.0347	0.0345	0.0341	0.0359	0.0351	**0.0378***	+8.05%
	NDCG@10	0.0047	0.0387	0.0391	0.0441	0.0443	0.0432	0.0447	0.0444	**0.0474***	+8.17%
	NDCG@20	0.0062	0.0448	0.0462	0.0519	0.0519	0.0504	0.0529	0.0521	**0.0564***	+8.79%
ML-1M	HR@5	0.0199	0.1745	0.1733	0.1786	0.1791	0.1744	0.1801	0.1784	**0.1863***	+3.44%
	HR@10	0.0363	0.2647	0.2651	0.2699	0.2707	0.2653	0.2711	0.2703	**0.2714***	+0.11%
	HR@20	0.0692	0.3649	0.3655	0.3778	0.3784	0.3707	**0.3798**	0.3788	0.3793	-0.13%
	NDCG@5	0.0126	0.1081	0.1142	0.1173	0.1169	0.1104	0.1185	0.1183	**0.1218***	+2.78%
	NDCG@10	0.0179	0.1379	0.1386	0.1466	0.1463	0.1417	0.1470	0.1465	**0.1492***	+1.50%
	NDCG@20	0.0261	0.1644	0.1654	0.1738	0.1741	0.1721	0.1743	0.1733	**0.1765***	+1.26%

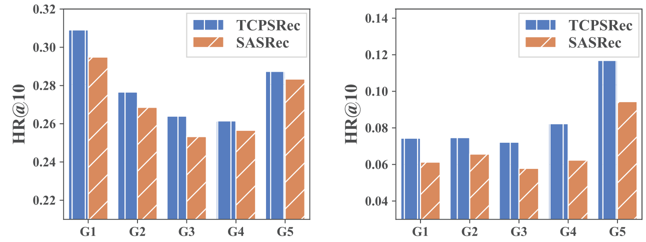


(a) Meituan (b) Yelp (c) Beauty (d) Sports

Figure 3: Ablation study on four recommendation datasets. "*w/o*" indicates that the corresponding objective is removed in the pre-training stage, while the rest objectives are kept.

Table 4: Performance comparison w.r.t. different base models enhanced by our proposed method TCPSRec.

Method	Meituan		Beauty	
	HR@10	NDCG@10	HR@10	NDCG@10
GRU4Rec	0.2320	0.1479	0.0640	0.0369
+TCPSRec	**0.2346**	**0.1511**	**0.0673**	**0.0382**
Caser	0.2433	0.1526	0.0677	0.0389
+TCPSRec	**0.2441**	**0.1553**	**0.0683**	**0.0394**
SASRec	0.2708	0.1745	0.0683	0.0405
+TCPSRec	**0.2797**	**0.1846**	**0.0841**	**0.0440**
TiSASRec	0.2711	0.1735	0.0694	0.0408
+TCPSRec	**0.2786**	**0.1808**	**0.0835**	**0.0433**



(a) Meituan (b) Beauty

Figure 4: Analysis of recommendation for different sparse-level users. $G5$ denotes the group of users with the largest average number of interactions, and $G1$ is the opposite.

4.3 Further Analysis

In this section, we conduct further analysis on TCPSRec .

4.3.1 ***Ablation Study.*** To validate the contribution of each pre-training objective in TCPSRec, we perform an ablation study on four real-world datasets, *i.e.,* Meituan, Yelp, Beauty, and Sports. We present the experimental results in Figure 3, where "*w/o* GIM", "*w/o* LIM", "*w/o* CPM" and "*w/o* FPM" denote the variants by removing the global invariant modeling objective, the local invariant modeling objective, the coarse-grained periodicity modeling objective, and the fine-grained periodicity modeling objective in TCPSRec, respectively. From this figure, we can observe that the performance of TCPSRec significantly decreases when removing any pre-training objective. Meanwhile, all the four variants perform better than the base model SASRec.

Furthermore, the importance of these objectives varies on different datasets. The global invariant modeling objective (GIM) brings more improvement than the other objectives. It is because GIM is more basic than the other two objectives. In addition, the improvements achieved by coarse-grained periodicity modeling (CPM) and fine-grained periodicity modeling (FPM) vary greatly across different datasets, which implies that different periodicities might be needed in different scenarios. Overall, the ablation study indicates that all pre-training objectives contribute to the performance improvement of our approach.

4.3.2 ***Applying the Pre-training Strategy on Other Models.*** With the default settings, we adopt SASRec as the based model in TCPSRec (refer to Section 2). Actually, our proposed pre-training method is generally applicable for other sequential recommender systems, such as GRU4Rec [13], Caser [28], and TiSASRec [18]. Therefore, we conduct experiments to examine whether the proposed pre-training strategy can also improve these sequential recommender systems. We report the results on the industry dataset Meituan and the public dataset Beauty in Table 4. From this table, we can observe that the proposed method can consistently improve the performance of GRU4Rec, Caser, SASRec, and TiSASRec, which further shows the effectiveness of the proposed method. In addition, we find that the proposed temporal pre-training method can also improve the performance of TiSASRec, a time-aware sequential recommender system, suggesting that our approach can better leverage temporal information. The experimental results further indicate the usefulness of temporal information in improving the recommendation performance.

4.3.3 ***Impact of Data Sparsity Levels.*** Traditional sequential recommender systems often suffer from data sparsity issues because they only rely on interaction sequences for training. To study the effect of the sparsity of interaction data on training, we group users according to the sparsity of their interactions and evaluate the performance of TCPSRec on different user groups separately. Concretely, we divide all users into five user groups according to their interaction number, each with the same total number of interactions (ensuring an equal amount of training data) but different numbers of users. Then, we train TCPSRec and SASRec separately on the interaction data of each user group, and compare their recommendation performance on these five user groups. The results are reported in Figure 4, where users in group $G1$ have the sparsest interaction data. It can be observed that TCPSRec outperforms SASRec on all user groups, which implies that the improvement brought by our method is stable and robust for users with different data sparsity levels. In addition, according to the results in Table 3, TCPSRec also achieves more improvement on the relatively sparse Beauty dataset than on the dense Meituan dataset. These findings show that the proposed TCPSRec can effectively alleviate the sparsity issue for sequential recommendation.

4.3.4 ***Impact on the Convergence Speed.*** To examine how pre-training affects the convergence speed, we further compare the loss of TCPSRec and SASRec during training on the recommendation task. Note that SASRec is randomly initialized, while TCPSRec corresponds to SASRec initialized by the pre-trained parameters. We adopt the early stopping strategy in the training process of recommendation tasks, *i.e.,* if HR@10 on the validation data does not increase for ten successive epochs, we consider that the model has converged and stop the training. Figure 5 shows the training process of SASRec and TCPSRec on Meituan and Beauty datasets. This figure shows that the proposed TCPSRec converges faster and achieves lower loss, while the SASRec needs more epochs to achieve similar performance. Specifically, on the Beauty dataset, SASRec requires more than 70 epochs of training to converge, while TCPSRec only needs about 20 epochs of fine-tuning to achieve better performance. The experimental results demonstrate that the proposed TCPSRec can significantly accelerate the convergence of recommendation models and achieve better performance.

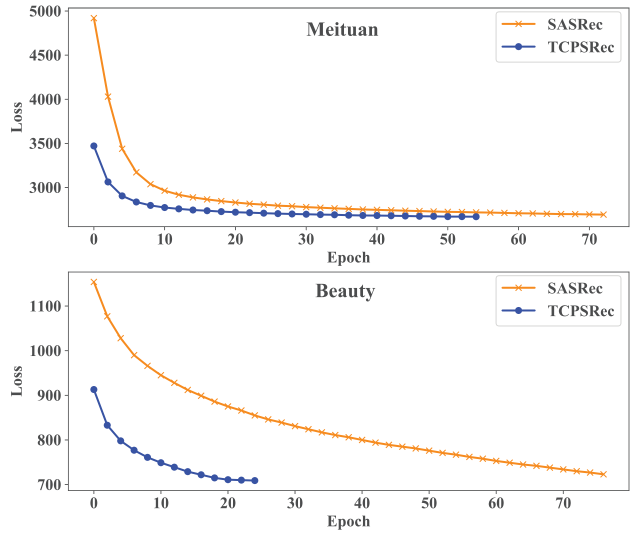


Figure 5: Analysis of convergence speed. SASRec is randomly initialized, while TCPSRec corresponds to the SASRec variant initialized by the pre-trained parameters.

5 RELATED WORK

In this section, we review the related work in two aspects, namely sequential recommendation and pre-training for recommendation.

5.1 Sequential Recommender Systems

Sequential recommender systems have gained success in capturing user preferences from sequential interaction data [33]. Early studies [11, 24] employ the Markov chain method to model the item transitions for capturing sequential relatedness. Subsequently, with the development of deep learning [17], Convolutional Neural Networks (CNN) [28], Recurrent Neural Networks (RNN) [13, 34] and other effective variants [14, 23] are applied to encode the user behavior sequences. Recently, significant attention has been dedicated to Transformer [30] based sequential models [3, 16, 20, 36]. For instance, SASRec [16] applies self-attention networks to model user behavior sequences, which effectively improves the recommendation performance. Furthermore, recent works [3, 36] introduce self-supervised learning into sequential user behavior modeling. Despite the remarkable success, the above methods seldom consider temporal information for modeling historical interactions. More recently, several methods [6, 8, 18, 32, 38] further leverage the temporal information for sequential recommendation. For example, TiSASRec [18] utilizes the time intervals between interactions to improve the self-attention mechanism of SASRec. TASER [38] takes into consideration both absolute time patterns and relative time intervals reflected by timestamps. However, these studies consider timestamps as auxiliary item features [6, 32] or use timestamps to develop time-aware models [8, 18, 38], which inevitably requires additional computational costs in the inference stage. Different from previous studies, our approach devises self-supervised objectives tailored for sequential recommendation, which can effectively capture temporal correlations for improving recommendation via pre-training.

5.2 Pre-training for Recommendation

Since the success of pre-training models in natural language processing [7, 37], the pre-training technique has also gained significant attention in the field of recommender systems [39]. Recently, some works apply the pre-training technique to sequential recommender systems [27, 35, 40, 45] and graph-based recommender systems [9, 21, 31] to better learn the representation of users and items. As for pre-trained sequential recommender systems, most of existing studies adopt Transformer [30] architecture. For example, BERT4Rec [27] and BERT4SessRec [2] employ the BERT [7] to model user behavior sequences and capture the bidirectional information. Based on the mutual information maximization principle, S^3-Rec [45] designs four self-supervised learning objectives to capture the correlations between sequence data and contextual information. UPRec [35] incorporates user information, such as user attributes and social graphs, into pre-trained models for recommendation. PeterRec [40] further improves the fine-tuning efficiency for the pre-trained parameters. With the development of contrastive learning (CL) [15, 29], somes works [1, 3, 22, 36] utilize CL in multi-task learning framework to improve sequential recommendation. For instance, CL4SRec [36] takes advantage of data augmentation strategies to construct self-supervised tasks and extract meaningful interaction patterns with CL. More recently, ICLRec [3] further propose to leverage prototypical contrastive learning to improve sequencial recommendation. These methods seldom consider temporal information in constructing the self-supervised signals, which is less capable of modeling sequential characteristics for the recommendation task.

6 CONCLUSIONS

In this paper, we propose a novel **T**emporal **C**ontrastive **P**re-training method for **S**equential **Rec**ommendation, named **TCPSRec**. Based on the temporal intervals of interactions, we divided the interaction sequence into coherent subsequences, and designed temporal pre-training objectives accordingly. Specifically, TCPSRec modeled both the invariance and periodicity of user behavior. We model the invariance at item level, and consider both global invariance within a sequence and local invariance within a subsequence. Moreover, we modeled the periodicity at subsequence level, considering coarse-grained periodicity and fine-grained periodicity. To integrate the above motivations, we developed a unified contrastive learning appraoch with four pre-training objectives tailored for sequential recommendation. The experimental results on six real-world datasets demonstrate the effectiveness of the proposed TCPSRec.

As future work, we will further consider extending our pre-training method to incorporate more kinds of side information, such as social graphs or auxiliary attributes.

ACKNOWLEDGMENTS

This work was partially supported by Beijing Natural Science Foundation under Grant No. 4222027, National Natural Science Foundation of China under Grant No. 61872369, and Beijing Outstanding Young Scientist Program under Grant No. BJJWZYJH012019100020098. This work was also partially supported by Beijing Academy of Artificial Intelligence (BAAI) and Meituan. Xin Zhao is the corresponding author.

REFERENCES

[1] Shuqing Bian, Wayne Xin Zhao, Kun Zhou, Jing Cai, Yancheng He, Cunxiang Yin, and Ji-Rong Wen. 2021. Contrastive Curriculum Learning for Sequential User Behavior Modeling via Data Augmentation. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 3737–3746.

[2] Xusong Chen, Dong Liu, Chenyi Lei, Rui Li, Zheng-Jun Zha, and Zhiwei Xiong. 2019. Bert4sessrec: Content-based video relevance prediction with bidirectional encoder representations from transformer. In *Proceedings of the 27th ACM International Conference on Multimedia*. 2597–2601.

[3] Yongjun Chen, Zhiwei Liu, Jia Li, Julian McAuley, and Caiming Xiong. 2022. Intent Contrastive Learning for Sequential Recommendation. In *Proceedings of the ACM Web Conference 2022*. 2172–2182.

[4] Mingyue Cheng, Fajie Yuan, Qi Liu, Xin Xin, and Enhong Chen. 2021. Learning Transferable User Representations with Sequential Behaviors via Contrastive Pre-training. In *ICDM*. IEEE, 51–60.

[5] Junsu Cho, Dongmin Hyun, SeongKu Kang, and Hwanjo Yu. 2021. Learning heterogeneous temporal patterns of user preference for timely recommendation. In *Proceedings of the Web Conference 2021*. 1274–1283.

[6] Qiang Cui, Chenrui Zhang, Yafeng Zhang, Jinpeng Wang, and Mingchen Cai. 2021. ST-PIL: Spatial-Temporal Periodic Interest Learning for Next Point-of-Interest Recommendation. In *CIKM*. 2960–2964.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. https://doi.org/10.18653/v1/N19-1423

[8] Ziwei Fan, Zhiwei Liu, Jiawei Zhang, Yun Xiong, Lei Zheng, and Philip S Yu. 2021. Continuous-time sequential recommendation with temporal graph collaborative transformer. In *CIKM*. 433–442.

[9] Bowen Hao, Jing Zhang, Hongzhi Yin, Cuiping Li, and Hong Chen. 2021. Pre-training graph neural networks for cold-start users and items representation. In *WSDM*. 265–273.

[10] F Maxwell Harper and Joseph A Konstan. 2015. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)* 5, 4 (2015).

[11] Ruining He and Julian McAuley. 2016. Fusing similarity models with markov chains for sparse sequential recommendation. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*. IEEE, 191–200.

[12] Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *proceedings of the 25th international conference on world wide web*. 507–517.

[13] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based recommendations with recurrent neural networks. *ICLR* (2016).

[14] Jin Huang, Wayne Xin Zhao, Hongjian Dou, Ji-Rong Wen, and Edward Y Chang. 2018. Improving sequential recommendation with knowledge-enhanced memory networks. In *SIGIR*. 505–514.

[15] Ashish Jaiswal, Ashwin Ramesh Babu, Mohammad Zaki Zadeh, Debapriya Banerjee, and Fillia Makedon. 2020. A survey on contrastive self-supervised learning. *Technologies* 9, 1 (2020), 2.

[16] Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *ICDM*. IEEE, 197–206.

[17] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *nature* 521, 7553 (2015), 436–444.

[18] Jiacheng Li, Yujie Wang, and Julian McAuley. 2020. Time interval aware self-attention for sequential recommendation. In *Proceedings of the 13th international conference on web search and data mining*. 322–330.

[19] Yicong Li, Hongxu Chen, Xiangguo Sun, Zhenchao Sun, Lin Li, Lizhen Cui, Philip S Yu, and Guandong Xu. 2021. Hyperbolic hypergraphs for sequential recommendation. In *CIKM*. 988–997.

[20] Yang Li, Tong Chen, Peng-Fei Zhang, and Hongzhi Yin. 2021. Lightweight self-attentive sequential recommendation. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 967–977.

[21] Yong Liu, Susen Yang, Chenyi Lei, Guoxin Wang, Haihong Tang, Juyong Zhang, Aixin Sun, and Chunyan Miao. 2021. Pre-training graph transformer with multimodal side information for recommendation. In *Proceedings of the 29th ACM International Conference on Multimedia*. 2853–2861.

[22] Ruihong Qiu, Zi Huang, Hongzhi Yin, and Zijian Wang. 2022. Contrastive learning for representation degeneration problem in sequential recommendation. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*. 813–823.

[23] Massimo Quadrana, Alexandros Karatzoglou, Balázs Hidasi, and Paolo Cremonesi. 2017. Personalizing session-based recommendations with hierarchical recurrent neural networks. In *proceedings of the Eleventh ACM Conference on Recommender Systems*. 130–137.

[24] Steffen Rendle. 2010. Factorization machines. In *2010 IEEE International conference on data mining*. IEEE, 995–1000.

[25] Paul Resnick and Hal R Varian. 1997. Recommender systems. *Commun. ACM* 40, 3 (1997), 56–58.

[26] Francesco Ricci, Lior Rokach, and Bracha Shapira. 2011. Introduction to recommender systems handbook. In *Recommender systems handbook*. Springer, 1–35.

[27] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management*. 1441–1450.

[28] Jiaxi Tang and Ke Wang. 2018. Personalized top-n sequential recommendation via convolutional sequence embedding. In *Proceedings of the eleventh ACM international conference on web search and data mining*. 565–573.

[29] Aaron Van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv e-prints* (2018), arXiv–1807.

[30] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[31] Hui Wang, Kun Zhou, Wayne Xin Zhao, Jingyuan Wang, and Ji-Rong Wen. 2022. Curriculum Pre-Training Heterogeneous Subgraph Transformer for Top-N Recommendation. *ACM Trans. Inf. Syst.* (mar 2022). https://doi.org/10.1145/3528667

[32] Jianling Wang, Raphael Louca, Diane Hu, Caitlin Cellier, James Caverlee, and Liangjie Hong. 2020. Time to Shop for Valentine's Day: Shopping Occasions and Sequential Recommendation in E-commerce. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. 645–653.

[33] Shoujin Wang, Liang Hu, Yan Wang, Longbing Cao, Quan Z. Sheng, and Mehmet Orgun. 2019. Sequential Recommender Systems: Challenges, Progress and Prospects. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*. International Joint Conferences on Artificial Intelligence Organization, 6332–6338. https://doi.org/10.24963/ijcai.2019/883

[34] Chao-Yuan Wu, Amr Ahmed, Alex Beutel, Alexander J Smola, and How Jing. 2017. Recurrent recommender networks. In *Proceedings of the tenth ACM international conference on web search and data mining*. 495–503.

[35] Chaojun Xiao, Ruobing Xie, Yuan Yao, Zhiyuan Liu, Maosong Sun, Xu Zhang, and Leyu Lin. 2021. UPRec: User-Aware Pre-training for Recommender Systems. *arXiv preprint arXiv:2102.10989* (2021).

[36] Xu Xie, Fei Sun, Zhaoyang Liu, Shiwen Wu, Jinyang Gao, Jiandong Zhang, Bolin Ding, and Bin Cui. 2022. Contrastive learning for sequential recommendation. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*. IEEE, 1259–1273.

[37] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems* 32 (2019).

[38] Wenwen Ye, Shuaiqiang Wang, Xu Chen, Xuepeng Wang, Zheng Qin, and Dawei Yin. 2020. Time matters: Sequential recommendation with complex temporal information. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1459–1468.

[39] Junliang Yu, Hongzhi Yin, Xin Xia, Tong Chen, Jundong Li, and Zi Huang. 2022. Self-Supervised Learning for Recommender Systems: A Survey. *arXiv preprint arXiv:2203.15876* (2022).

[40] Fajie Yuan, Xiangnan He, Alexandros Karatzoglou, and Liguang Zhang. 2020. Parameter-efficient transfer from sequential behaviors for user modeling and recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1469–1478.

[41] Xu Yuan, Hongshen Chen, Yonghao Song, Xiaofang Zhao, and Zhuoye Ding. 2021. Improving Sequential Recommendation Consistency with Self-Supervised Imitation. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, Zhi-Hua Zhou (Ed.). International Joint Conferences on Artificial Intelligence Organization, 3321–3327. Main Track.

[42] Wayne Xin Zhao, Junhua Chen, Pengfei Wang, Qi Gu, and Ji-Rong Wen. 2020. Revisiting Alternative Experimental Settings for Evaluating Top-N Item Recommendation Algorithms. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 2329–2332.

[43] Wayne Xin Zhao, Yupeng Hou, Xingyu Pan, Chen Yang, Zeyu Zhang, Zihan Lin, Jingsen Zhang, Shuqing Bian, Jiakai Tang, Wenqi Sun, Yushuo Chen, Lanling Xu, Gaowei Zhang, Zhen Tian, Changxin Tian, Shanlei Mu, Xinyan Fan, Xu Chen, and Ji-Rong Wen. 2022. RecBole 2.0: Towards a More Up-to-Date Recommendation Library. In *Proceedings of the 31th ACM International Conference on Information & Knowledge Management*.

[44] Wayne Xin Zhao, Shanlei Mu, Yupeng Hou, Zihan Lin, Yushuo Chen, Xingyu Pan, Kaiyuan Li, Yujie Lu, Hui Wang, Changxin Tian, et al. 2021. Recbole: Towards a unified, comprehensive and efficient framework for recommendation algorithms. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 4653–4664.

[45] Kun Zhou, Hui Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. 2020. S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 1893–1902.