Temporal Graph Contrastive Learning for Sequential Recommendation

Shengzhe Zhang[1], **Liyi Chen**[1], **Chao Wang**[2, 3], **Shuangli Li**[1], **Hui Xiong**[3, 4*]
[1]University of Science and Technology of China
[2]Guangzhou HKUST Fok Ying Tung Research Institute
[3]The Thrust of Artificial Intelligence, The Hong Kong University of Science and Technology (Guangzhou)
[4]The Department of Computer Science and Engineering, The Hong Kong University of Science and Technology
{szowen.contact, liyichencly, chadwang2012}@gmail.com, lsl1997@mail.ustc.edu.cn, xionghui@ust.hk

Abstract

Sequential recommendation is a crucial task in understanding users' evolving interests and predicting their future behaviors. While existing approaches on sequence or graph modeling to learn interaction sequences of users have shown promising performance, how to effectively exploit temporal information and deal with the uncertainty noise in evolving user behaviors is still quite challenging. To this end, in this paper, we propose a Temporal Graph Contrastive Learning method for Sequential Recommendation (TGCL4SR) which leverages not only local interaction sequences but also global temporal graphs to comprehend item correlations and analyze user behaviors from a temporal perspective. Specifically, we first devise a Temporal Item Transition Graph (TITG) to fully leverage global interactions to understand item correlations, and augment this graph by dual transformations based on neighbor sampling and time disturbance. Accordingly, we design a Temporal item Transition graph Convolutional network (TiTConv) to capture temporal item transition patterns in TITG. Then, a novel Temporal Graph Contrastive Learning (TGCL) mechanism is designed to enhance the uniformity of representations between augmented graphs from identical sequences. For local interaction sequences, we design a temporal sequence encoder to incorporate time interval embeddings into the architecture of Transformer. At the training stage, we take maximum mean discrepancy and TGCL losses as auxiliary objectives. Extensive experiments on several real-world datasets show the effectiveness of TGCL4SR against state-of-the-art baselines of sequential recommendation.

1 Introduction

Sequential recommendation (SR) aims to forecast subsequent interactions of users by analyzing their historical behaviors in a chronological sequence. Employed extensively across online platforms, SR models capitalize on sequential dependencies of interactions, thus offering potential advantages in grasping item correlations and tracking user interest evolution over non-temporal recommendation systems (Tang and Wang 2018; Wu et al. 2019; Wang et al. 2023). However, the inherent unpredictability and intricacy of sequential behaviors amplify the challenges in SR. Temporal data, often laden with noise, further obfuscates the precise mapping of user-item interactions (Wang et al. 2019).

Predominantly, SR models in existing literature either utilize sequential modeling for encoding user interaction sequences (Hidasi and Karatzoglou 2018; Kang and McAuley 2018; Sun et al. 2019) or transform these sequences into graph structures for enhanced item correlation mining (Wu et al. 2019; Xu et al. 2019; Wang et al. 2020). Recent advancements underscore the significance of employing specific temporal features, revealing that such features can substantially augment model performance. For instance, TiSASRec (Li, Wang, and McAuley 2020) introduces a time interval-aware self-attention mechanism, enabling superior modeling of item correlations. Likewise, TGSR (Fan et al. 2021) designs a temporal interaction bipartite graph paired with a collaborative transformer network, adeptly mapping user interest trajectories. MOJITO (Tran et al. 2023) deploys attention blocks to encode varied temporal dimensions, enriching the understanding of user interest evolution.

Despite these advancements, challenges persist in leveraging specific temporal information in SR. First, the multifaceted nature of temporal data, whether expressed in absolute or relative terms, poses integration complexities (Tran et al. 2023). Hence, it's difficult to appropriately and effectively exploit multi-type temporal information in sequential recommendations. Second, diverse changes at which user interests evolve complicate the deciphering of global interest transition patterns (Wang et al. 2020, 2021b). Learning global transition patterns of user interests is important and difficult. Third, unpredictable and diverse user behaviors tend to introduce noisy information (Wu et al. 2019), requiring robust and accurate temporal integration in SR.

To address the above challenges, in this paper, we propose a Temporal Graph Contrastive Learning method for Sequential Recommendation (TGCL4SR) which leverages not only local interaction sequential information but also global temporal graphs to comprehend item correlations and analyze user behaviors. Specifically, to fully leverage global interactions to explore item correlations, we first design a Temporal Item Transition Graph (TITG), which builds edges between items from adjacent interactions within a sequence and takes corresponding timestamps and users as edge attributes. Besides, due to the noise and the large scale of the proposed temporal graph, we augment the TITG by dual

*The corresponding author.

transformations based on neighbor sampling and time disturbance. For further analyzing item transition patterns in augmented TITGs, we devise a Temporal item Transition graph Convolutional network (TiTConv) to aggregate the features of neighboring items with absolute time and users together. Then, to soften data sparsity and uncertainty of user behaviors, temporal graph contrastive learning including subgraph and disturbed contrastive learning is proposed to improve the uniformity of representations among the augmented graphs stemming from identical sequences. For local interaction sequences, a temporal sequence encoder is designed to incorporate time interval embeddings into the architecture of Transformer (Vaswani et al. 2017), which accurately captures the user interest evolution. Finally, at the training stage, apart from the cross-entropy and contrastive losses, we use Maximum Mean Discrepancy (MMD) (Li, Swersky, and Zemel 2015) loss to align item representations from global graphs and local sequences respectively, which maintains semantic consistency. Extensive experiments on four public datasets show our proposed TGCL4SR outperforms competitive baselines with significant improvements. We summarize the contributions of this paper as follows:

- To fully comprehend item correlations and analyze user behaviors from a temporal perspective, we propose a novel method, namely TGCL4SR, which learns both global temporal graphs and local interaction sequences.
- We specially design a TITG to integrate global interactions for effectively understanding item correlations. Moreover, to analyze item transition patterns in the TITG, we devise a TiTConv to aggregate the features of neighboring items with absolute time and users.
- For handling the large-scale graph and noisy temporal information, we augment the TITG by dual transformations based on neighbor sampling and time disturbance, and propose a temporal graph contrastive learning strategy to enhance the robustness and quality of representations.

2 Related Work

In this section, we provide a review of related work, which can be grouped into sequential recommendation and contrastive learning for recommendation.

2.1 Sequential Recommendation

SR models aim to predict the next item a user will consume based on the user's behavior sequence. Many works on SR focus on how to capture users' long-term and short-term interest patterns. For instance, GRU4Rec (Hidasi and Karatzoglou 2018), SASRec (Kang and McAuley 2018) and BERT4Rec (Sun et al. 2019) utilized RNN, Transformer and BERT respectively. In the meantime, many studies employ GNNs on various graphs for SR to discover underlying item transition patterns, such as SR-GNN (Wu et al. 2019) and GCE-GNN (Wang et al. 2020). Recently, MAERec (Ye, Xia, and Huang 2023) suggested a light graph-masked encoder to avoid noise in data and enhance pattern representation learning. Since time significantly influences user interaction context, some SR works start to focus on temporal information utilization. For example, TiSASRec (Li, Wang, and McAuley 2020) added time interval encoding into SASRec, which considered absolute position and related time simultaneously. TGSR (Fan et al. 2021) used temporal bipartite interaction graphs and designed a temporal collaborative transformer to learn user interest tendencies. Recently, TiCoSeRec (Dang et al. 2023) found out that uniformly distributed time intervals improve SR models better than varying time intervals, so it suggested specific operators to convert origin time intervals into uniform ones. Differently, we integrate time information into transition graphs and use time disturbance to tackle noise issues.

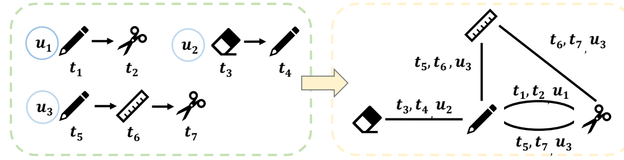


Figure 1: A toy example of the construction of TITG.

2.2 Contrastive Learning for Recommendation

Contrastive learning is an effective unsupervised learning paradigm in various domains (Chen et al. 2022b; Li et al. 2022, 2023) and has been widely used in SR. For instance, CL4SRec (Xie et al. 2022) introduced contrastive learning to SR by augmenting sequences with several operations. DuoRec (Qiu et al. 2022) considered sequences with the same target to have similar semantics and made them pairs of positive samples to conduct contrastive learning. Discovering that employing contrastive learning on graph representations (Hassani and Khasahmadi 2020) can obtain global item correlation signals, some works also designed various graph contrastive learning. For example, SGL (Wu et al. 2021) improved recommendation models by random nodes and edges dropout to create contrastive views. GCL4SR (Zhang et al. 2022) first performed graph contrastive learning on SR by random neighbor sampling on item transition graphs to get stable item representations in different subgraphs. LightGCL (Cai et al. 2023) applied singular value decomposition as graph augmentation to enhance global collaborative relation learning process. Different from previous works, we focus on conducting graph contrastive learning on temporal graphs.

3 Problem Definition

Traditional sequential recommendation systems focus on predicting the subsequent item a user might interact with by analyzing their prior interaction sequences. However, such a method, which primarily leverages sequential information, may not sufficiently capture intricate relationships embedded in item transitions and their chronological intricacies.

Temporal Item Transition Graph. To delve deeper into item correlations from a temporal viewpoint, we propose a Temporal Item Transition Graph (TITG), denoted as $\mathcal{G}$. This graph effectively merges temporal aspects and user representations, offering a comprehensive perspective on temporal item relations across behavioral sequences of all users.

Specifically, considering a user u from the user set $\mathcal{U}$. For each item v_i^u from the user sequence S^u, we construct an edge between v_i^u and $v_{i+\epsilon}^u$ in $\mathcal{G}$, where $\epsilon \in \{1,2,3,4\}$. The attributes of this edge include the corresponding timestamps t_i^u and $t_{i+\epsilon}^u$ and the user u, which can be represented as a triplet $(t_i^u, t_{i+\epsilon}^u, u)$. Figure 1 shows a toy example of this construction process. Note that the same pair of items may have multiple edges, since they can co-occur in different interaction sequences. Therefore, we use a quintuple $e = (v_i, v_j, u, t_1^u, t_2^u)$ to indicate that edge e is built based on the fact that user u interacted with item v_i, v_j on t_1^u, t_2^u respectively. Given the backdrop above, the formal problem definition of our task is presented as follows:

Definition 1 *Assume a user set $\mathcal{U}$ and item set $\mathcal{V}$. For each user $u \in \mathcal{U}$, the chronologically ordered interaction sequence is $S^u = \{v_1^u, v_2^u, .., v_{|S^u|}^u\}$, and its associated timestamp sequence is $T^u = \{t_1^u, t_2^u, .., t_{|S^u|}^u\}$, where t_i^u signifies the timestamp of the interaction (u, v_i^u). TITG $\mathcal{G}$ is constructed based on all historical sequences. Our objective is to forecast the next item that user u will probably interact with at the $(|S_u|+1)$-th step based on S^u, T^u, and $\mathcal{G}$.*

4 Methodology

In this section, we introduce the technical details of our proposed TGCL4SR. As illustrated in Figure 2, TGCL4SR includes five main components, i.e., Dual Graph Augmentation, TiTConv Layer, Temporal Graph Contrastive Learning, Temporal Sequence Encoder and Prediction Layer. Through the above parts, our method learns behavior sequence representations from both global and local perspectives. Then representations of each sequence are concatenated at the Prediction Layer. Finally, we use the multi-task manner for model learning at the training stage.

4.1 Dual Graph Augmentation

To relieve the problems of data sparsity and noise, we augment TITG through a dual transformation. We first sample node neighbors on the graph to obtain subgraphs to directly lower the size of TITG to compute and alleviate data sparsity. Then we add random time disturbance to the subgraphs so that the model can be less influenced by time noises.

Neighbor Sampling As mentioned above, TITG is a multi-edge graph. Therefore, directly computing representations on the whole graph is costly and ineffective. Also, the sparsity of the graph data may lead to overfitting. Hence, the neighbor sampling method (Hamilton, Ying, and Leskovec 2017) is applied to derive smaller and augmented graph views. In specific, given TITG $\mathcal{G}$ and an interaction sequence $\mathcal{S}$, we sample the neighbor nodes of each item $v \in \mathcal{S}$ in $\mathcal{G}$ randomly and uniformly. We repeat this process with the sample depth M and the sample size N at each depth. In this way, we can generate two augmented subgraph views $\mathcal{G}_\mathcal{S}^1 = (\mathcal{V}_\mathcal{S}^1, \mathcal{E}_\mathcal{S}^1)$ and $\mathcal{G}_\mathcal{S}^2 = (\mathcal{V}_\mathcal{S}^2, \mathcal{E}_\mathcal{S}^2)$, where $\mathcal{V}_\mathcal{S}^1, \mathcal{E}_\mathcal{S}^1, \mathcal{V}_\mathcal{S}^2, \mathcal{E}_\mathcal{S}^2$ are the sets of nodes and edges for $\mathcal{G}_\mathcal{S}^1$ and $\mathcal{G}_\mathcal{S}^2$, respectively.

Time Disturbance Temporal information sometimes contains noises and cannot precisely indicate the change of user interest, this requires models to avoid being overly influenced by the temporal noise in the data. Meanwhile, the sparsity and noises of the data may result in overfitting. Therefore, we devise the method of time disturbance to augment the graph data. Here, we randomly add noise to the timestamp attribute of edges in TITGs to generate a temporal-augmented graph view from the given graph. Specifically, taking $\mathcal{G}_S^1$ as an example, for each edge $e \in \mathcal{E}_S^1$, we add Gaussian noise $o_1 \sim N(0, \sigma)$ and $o_2 \sim N(0, \sigma)$ with probability p to its timestamp attributes t_1^u and t_2^u respectively. Here, σ is the standard deviation of the Gaussian distribution. Then we can obtain the temporal-augmented graph $\mathcal{G}_S^{1'}$, and through the same way we generate $\mathcal{G}_S^{2'}$ from $\mathcal{G}_S^2$.

Through the two graph-augment methods above, we finally obtain four enhanced graphs based on sequence S: $\mathcal{G}_S^1, \mathcal{G}_S^2, \mathcal{G}_S^{1'}, \mathcal{G}_S^{2'}$, effectually reducing issues of graph scale, data sparsity and noise impact.

4.2 TiTConv Layer

To learn fine-grained item correlations on TITG, we propose a Temporal item Transition graph Convolutional layer (TiTConv). Below we present the details of TiTConv in our model at its $(l+1)$-th layer.

For timestamp attributes of edges, we use the harmonic encoder (Xia et al. 2021) to encode them:

$$\boldsymbol{\phi}(t) = [\cos(w_1 t + b_1), \ldots, \cos(w_n t + b_n)], \quad (1)$$

where $\cos(\cdot)$ is the cosine function, n is the hidden dimension, $w_1, ..., w_n$ and $b_1, ..., b_n$ are learnable weights and bias parameters respectively, $\phi(t)$ denotes the embedding of the timestamp t. Then for each edge $e = (v_i, v_j, u, t_1^u, t_2^u)$ in the input graph, the message $\boldsymbol{m}_e^{(l)}$ that v_i conveys to v_j through e is calculated as follows:

$$\boldsymbol{m}_i^{(l)} = \mathrm{Concat}\left(\boldsymbol{h}_i^{(l)}, \boldsymbol{\phi}(0), \boldsymbol{u}\right), \quad (2)$$

$$\boldsymbol{m}_j^{(l)} = \mathrm{Concat}\left(\boldsymbol{h}_j^{(l)}, \mathrm{MLP}\left(\boldsymbol{\phi}(t_1) \| \boldsymbol{\phi}(t_2)\right), \boldsymbol{u}\right), \quad (3)$$

$$\boldsymbol{m}_e^{(l)} = \mathrm{Concat}(\mathrm{head}_1, \ldots, \mathrm{head}_\eta) \boldsymbol{W}_h, \quad (4)$$

$$\mathrm{head}_k = \mathrm{Attention}\left(\boldsymbol{m}_i^{(l)} \boldsymbol{W}_k^Q, \boldsymbol{m}_j^{(l)} \boldsymbol{W}_k^K, \boldsymbol{m}_j^{(l)} \boldsymbol{W}_k^V\right), \quad (5)$$

where $\boldsymbol{h}_i^{(l)}, \boldsymbol{u} \in \mathbb{R}^{1\times n}$ represents the embedding of their corresponding item v_i and user u respectively, $\|$ is the concatenation operation, MLP is the multi-layer perceptron, η is the number of heads, $\boldsymbol{W}_k^Q, \boldsymbol{W}_k^K, \boldsymbol{W}_k^V \in \mathbb{R}^{3n\times 3n/\eta}$ are projection matrices for each attention head, and $\boldsymbol{W}_h \in \mathbb{R}^{3n\times 3n}$ is the projection matrix. The attention function is defined as:

$$\mathrm{Attention}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \mathrm{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{d/\eta}}\right)\boldsymbol{V}, \quad (6)$$

where $\boldsymbol{Q}, \boldsymbol{K}$ and $\boldsymbol{V}$ denote queries, keys and values respectively, softmax is the activation function, $\sqrt{d/\eta}$ is the scale factor. Note that there may exist multiple edges between v_i and v_j in many practical datasets, we flatten them and compute their messages respectively. After message propagation, we can obtain the representation of v_k at the next

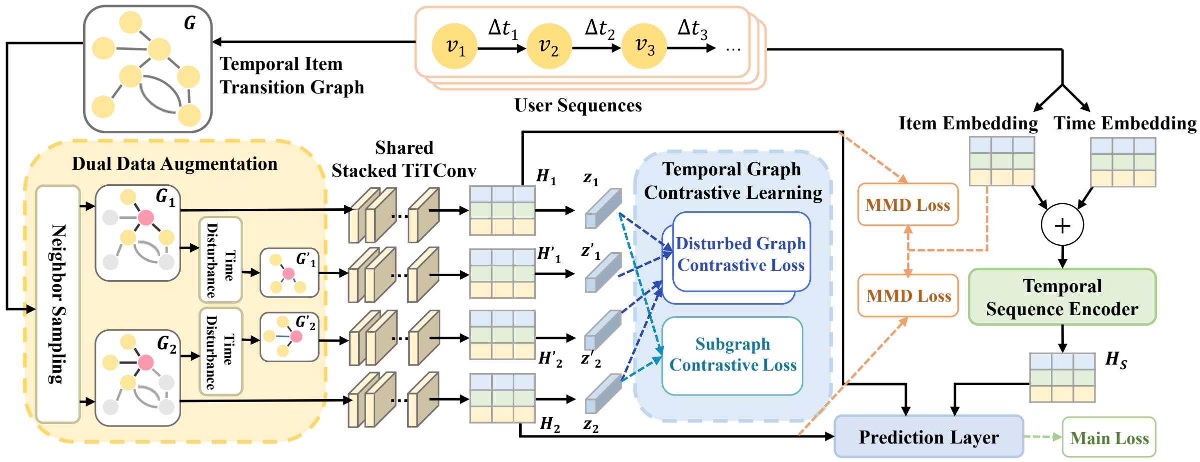


Figure 2: The overall architecture of the proposed TGCL4SR.

layer using a sum aggregation and MLP:

$$\boldsymbol{h}_k^{(l+1)} = \mathrm{MLP}\left(\boldsymbol{h}_k^{(l)} \| \sum_{e \in \mathcal{E}_{v_k}} \boldsymbol{m}_e^{(l)}\right), \tag{7}$$

where $\mathcal{E}_{v_k}$ is the set of edges incident to v_k. After that, GraphSAGE (Hamilton, Ying, and Leskovec 2017) with mean aggregation is stacked for deeper neighbor features.

Employing shared TiTConv on four augmented graphs above, at the last layer we can get the embeddings of the items in S denoted as $\boldsymbol{H}_S^1, \boldsymbol{H}_S^2, \boldsymbol{H}_S^{1'}, \boldsymbol{H}_S^{2'} \in \mathbb{R}^{|S| \times n}$ respectively. We conduct mean pooling on these representations respectively and obtain $\boldsymbol{z}_S^1, \boldsymbol{z}_S^2, \boldsymbol{z}_S^{1'}, \boldsymbol{z}_S^{2'} \in \mathbb{R}^{1 \times n}$.

4.3 Temporal Graph Contrastive Learning

To mitigate data sparsity and bolster stable item representation, we devise the unique graph contrastive learning from a temporal perspective. In the scenario of SR, positive sample pairs consist of different views of identical sequences, while views of different sequences in the same mini-batch are negative ones. The model should minimize the distance between positive sample pairs and make negative ones less similar to improve the uniformity of representations between augmented graphs derived from identical sequences.

Specifically, we first utilize the subgraph contrastive learning (SGCL) objective to contrast the representations of different subgraphs. Negative sampling is adopted (Chen et al. 2020). For each sequence S in a mini-batch B, $(\mathcal{G}_S^1, \mathcal{G}_S^2)$ is a positive sample pair, while for other sequence $\mathcal{P} \in B$, $(\mathcal{G}_S^1, \mathcal{G}_\mathcal{P}^2)$ is a negative pair. Then InfoNCE (He et al. 2020) is applied to calculate the loss of SGCL on S:

$$\mathcal{L}_{SGCL}(S) = -\log \frac{\exp\left(sim\left(\boldsymbol{z}_S^1, \boldsymbol{z}_S^2\right)/\tau\right)}{\sum_{\mathcal{P} \in B} \exp\left(sim\left(\boldsymbol{z}_S^1, \boldsymbol{z}_\mathcal{P}^2\right)/\tau\right)}, \tag{8}$$

where $sim(\cdot, \cdot)$ is the cosine similarity function, τ is the temperature parameter for SGCL.

Besides SGCL, a disturbed graph contrastive learning (DGCL) objective is designed to contrast the hidden representations of original subgraphs and their temporal-augmented graphs. For each sequence $S \in B$, $(\mathcal{G}_S^1, \mathcal{G}_S^{1'})$ and $(\mathcal{G}_S^2, \mathcal{G}_S^{2'})$ are positive sample pairs, while for other sequence $\mathcal{P} \in B$, $(\mathcal{G}_S^1, \mathcal{G}_\mathcal{P}^{1'})$ and $(\mathcal{G}_S^2, \mathcal{G}_\mathcal{P}^{2'})$ are negative ones. Then the loss function of DGCL on S is formulated as follows:

$$\mathcal{L}_{DGCL}(S) = -\frac{1}{2}\left(\log \frac{\exp\left(sim\left(\boldsymbol{z}_S^1, \boldsymbol{z}_S^{1\prime}\right)/\tau'\right)}{\sum_{\mathcal{P} \in B} \exp\left(sim\left(\boldsymbol{z}_S^1, \boldsymbol{z}_\mathcal{P}^{1\prime}\right)/\tau'\right)} + \log \frac{\exp\left(sim\left(\boldsymbol{z}_S^2, \boldsymbol{z}_S^{2\prime}\right)/\tau'\right)}{\sum_{\mathcal{P} \in B} \exp\left(sim\left(\boldsymbol{z}_S^2, \boldsymbol{z}_\mathcal{P}^{2\prime}\right)/\tau'\right)}\right), \tag{9}$$

where τ' is the temperature parameter for DGCL.

Then, we sum up them as the total loss for TGCL on S:

$$\mathcal{L}_{TGCL}(S) = \mathcal{L}_{SGCL}(S) + \mathcal{L}_{DGCL}(S). \tag{10}$$

4.4 Temporal Sequence Encoder

To study the representations of user local interaction sequences with temporal information, we design a Temporal Sequence Encoder. To effectively learn the complex sequential interest patterns, we utilize the stacked Transformer layers as the basic sequence encoder, which can capture the evolving user interest. Considering that the length of time intervals may affect the influence of the current interaction toward the next interaction in a sequence (Chen et al. 2022a; Wang et al. 2021a), we combine the representations of items and internal time intervals. Given the sequence S and its corresponding timestamp sequence T, we first convert $\{t_1, .., t_{|S|}\}$ into a time-interval sequence:

$$\left\{\delta_1, \delta_2, \ldots, \delta_{|S|}\right\} = \left\{0, t_2 - t_1, \ldots, t_{|S|} - t_{|S|-1}\right\}. \tag{11}$$

After that, in light of the logarithmic decay for user interest (Wu, Cai, and Wang 2020), we apply the following function to calculate the corresponding position embedding of the time interval δ_i:

$$\mathrm{pos}_i = \lfloor a \log\left(\delta_i / c + 1\right)\rfloor, \tag{12}$$

where a and c are scaling constants. After that we get the embedding of δ_i denoted as $\boldsymbol{\delta_i} \in \mathbb{R}^{1\times n}$, and add it to the embedding of the i-th item in S as the input representation $\boldsymbol{R}_S^{(0)} \in \mathbb{R}^{|S|\times n}$. Next, given the sequential representation $\boldsymbol{R}_S^{(l)}$ at the l-th layer of the multi-head self-attention Transformer encoder, the output of the $(l+1)$-th layer is as below:

$$\boldsymbol{R}_S^{(l+1)} = \text{Transformer}(\boldsymbol{R}_S^{(l)}). \tag{13}$$

4.5 Prediction Layer

After performing the designed TiTConv and temporal sequence encoder for S^u, we can obtain the representations $\boldsymbol{H}^1_{S^u}$ and $\boldsymbol{H}^2_{S^u}$ from augmented temporal graphs, as well as $\boldsymbol{R}_{S^u}$ from the user sequence. The next step involves concatenating and integrating these multi-faceted sequential features in the following manner:

$$\boldsymbol{P}^u = \text{AttNet}\left(\left(\boldsymbol{H}^1_{S^u} \| \boldsymbol{H}^2_{S^u} \| \boldsymbol{R}_{S^u}\right) \boldsymbol{W}_T\right), \tag{14}$$

where $\boldsymbol{P}^u \in \mathbb{R}^{1\times n}$ is the interest matrix of user u, $\boldsymbol{W}_T \in \mathbb{R}^{3n\times n}$ is a trainable weight matrix. $\text{AttNet}(\cdot)$ denotes the attention network employed in (Zhang et al. 2022). Then the following formulation calculates the possibility that the user u would interact with the expected item v at the $(|S^u|+1)$-th step according to S^u:

$$\hat{\boldsymbol{y}}_v^{S^u} = \text{Sigmoid}\left(\boldsymbol{P}^u \boldsymbol{v}^T\right), \tag{15}$$

where Sigmoid is the activation function, $\boldsymbol{v} \in \mathbb{R}^{1\times n}$ denotes the corresponding embedding of v.

4.6 Model Optimization

In order to make model training more stable and achieve better recommendation performance, we combine the above TGCL to enhance the sequential recommendation with an additional MMD loss in a multi-task learning manner.

Firstly, the main target of the sequential recommendation is to use the behavior sequence S^u to predict the next item that user u is most interested in. For the interaction sequence $S^u = \{v_1^u, v_2^u, ..., v_{|S^u|}^u\}$ of each user u, we take the subsequence $S_{k-1}^u = \{v_1^u, v_2^u, ..., v_{k-1}^u\}$ and its corresponding target item v_k^u as training data at each time step k from 2 to $|S^u|$. Then for all users, we adopt the cross-entropy loss function to optimize the model:

$$\mathcal{L}_{\text{SR}} = -\sum_{u\in\mathcal{U}} \sum_{k=2}^{|S^u|} \log \frac{\exp\left(\hat{\boldsymbol{y}}_{v_k^u}^{S_{k-1}^u}\right)}{\sum_{v\in\mathcal{V}} \exp\left(\hat{\boldsymbol{y}}_{v}^{S_{k-1}^u}\right)}, \tag{16}$$

We also attach MMD (Li, Swersky, and Zemel 2015) as an auxiliary objective to our training task to advance the semantic agreement between item embeddings and representations learned from TITG. Given two sample sets of two representation distributions $\boldsymbol{X} = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_m\}$ and $\boldsymbol{Y} = \{\boldsymbol{y}_1, ..., \boldsymbol{y}_{m'}\}$, MMD between them can be estimated as:

$$\begin{aligned} MMD(\boldsymbol{X}, \boldsymbol{Y}) &= \frac{1}{m^2} \sum_{i=1}^{m} \sum_{j=1}^{m} \mathcal{K}\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) + \\ \frac{1}{m'^2} \sum_{i=1}^{m'} \sum_{j=1}^{m'} \mathcal{K}\left(\boldsymbol{y}_i, \boldsymbol{y}_j\right) &- \frac{2}{mm'} \sum_{i=1}^{m} \sum_{j=1}^{m'} \mathcal{K}\left(\boldsymbol{x}_i, \boldsymbol{y}_j\right), \end{aligned} \tag{17}$$

Datasets	Beauty	Games	CDs	Comics
# User	22,363	24,303	75,258	14,109
# Item	12,101	10,672	64,443	17,035
# Interaction	198,502	231,780	1,097,592	353,878
Avg. actions/user	8.88	9.53	14.58	25.08
Avg. actions/item	16.40	21.72	17.03	20.77
# Edges in TITG	398,636	497,113	3,053,824	1,163,516

Table 1: Statistics of processed datasets.

$$\mathcal{K}(x, y) = \exp\left(-\frac{\|x-y\|^2}{2\rho^2}\right) \tag{18}$$

where $\mathcal{K}(\cdot,\cdot)$ denotes the Gaussian kernel with the bandwidth ρ. Following (Zhang et al. 2022), we intend to minimize MMD between the local sequence embedding $\boldsymbol{S}^u$ and the temporal global context $\boldsymbol{H}^1_{S^u}$ and $\boldsymbol{H}^2_{S^u}$. Therefore, the MMD loss for S^u is as follows.

$$\mathcal{L}_{MMD}(S^u) = MMD\left(\boldsymbol{S}^u, \boldsymbol{H}^1_{S^u}\right) + MMD\left(\boldsymbol{S}^u, \boldsymbol{H}^2_{S^u}\right). \tag{19}$$

Finally, we sum up the loss functions of the main prediction task, temporal graph contrastive learning task, and the MMD constraint to reach the following objective:

$$\mathcal{L} = \mathcal{L}_{\text{SR}} + \sum_{u\in\mathcal{U}} \sum_{k=2}^{|S_u|} \left(\lambda_1 \mathcal{L}_{TGCL}(S_{k-1}^u) + \lambda_2 \mathcal{L}_{MMD}(S_{k-1}^u)\right), \tag{20}$$

where λ_1, λ_2 are loss weight hyper-parameters.

5 Experiments

In this section, we present detailed experiments to demonstrate the effectiveness of our proposed TGCL4SR on sequential recommendation tasks.

5.1 Experiment Settings

Datasets. Four public datasets are chosen for the evaluation of SR models from Amazon Review (He and McAuley 2016) and Goodreads (Wan et al. 2019). Amazon Review collects user reviews on products in various categories from Amazon. We choose the datasets of three categories “Beauty”, “Video Games” and “CDs” for experiments. In addition, Goodreads Review contains user reviews on books of various genres from Goodreads. We take the dataset of “Comics Graphic” for evaluation. For each dataset, we remove duplicated interactions and sort each user’s interactions by their timestamps chronologically to build user behavior sequences. We filter out users and items that have less than 5 reviews to get the 5-core subset of each dataset. The statistics of processed datasets are shown in Table 1.

Metrics and Evaluation. The performances are evaluated by top-K Hit Ratio (HR@K) and Normalized Discounted Cumulative Gain (NDCG@K), K $\in$ {10, 20}. We adopt the *leave-one-out* evaluation strategy. The ranking of predictions is computed on the full item set but not sampling.

Datasets	Metrics	GRU4Rec	SASRec	TiSASRec	CL4SRec	DuoRec	TCPSRec	GCL4SR	MStein	TGCL4SR
Beauty	HR@10	0.0305	0.0550	0.0561	0.0579	0.0690	0.0857	<u>0.0910</u>	0.0774	**0.0946**
	HR@20	0.0467	0.0789	0.0837	0.0905	0.1033	0.1202	<u>0.1222</u>	0.1058	**0.1295**
	NDCG@10	0.0164	0.0280	0.0298	0.0278	0.0369	0.0427	<u>0.0548</u>	0.0463	**0.0557**
	NDCG@20	0.0305	0.0339	0.0375	0.0370	0.0491	0.0519	<u>0.0627</u>	0.0545	**0.0648**
Games	HR@10	0.0430	0.0798	0.0779	0.0968	0.1183	0.1192	<u>0.1406</u>	0.1066	**0.1572**
	HR@20	0.0683	0.1178	0.1175	0.1513	0.1809	0.1847	<u>0.1991</u>	0.1550	**0.2239**
	NDCG@10	0.0225	0.0371	0.0381	0.0478	0.0619	0.0563	<u>0.0788</u>	0.0586	**0.0884**
	NDCG@20	0.0288	0.0466	0.0479	0.0616	0.0776	0.0728	<u>0.0935</u>	0.0709	**0.1051**
CDs	HR@10	0.0059	0.0286	0.0351	0.0498	0.0931	0.0914	<u>0.1107</u>	0.0544	**0.1119**
	HR@20	0.0102	0.0387	0.0499	0.0771	0.1366	0.1365	**0.1546**	0.0737	<u>0.1529</u>
	NDCG@10	0.0031	0.0141	0.0159	0.0244	0.0454	0.0413	<u>0.0631</u>	0.0318	**0.0652**
	NDCG@20	0.0041	0.0166	0.0197	0.0313	0.0564	0.0527	<u>0.0741</u>	0.0367	**0.0756**
Comics	HR@10	0.0425	0.0846	0.0857	0.0911	0.1736	0.1787	<u>0.1831</u>	0.1628	**0.1952**
	HR@20	0.0746	0.1045	0.1087	0.1298	0.2248	<u>0.2267</u>	0.2260	0.1934	**0.2463**
	NDCG@10	0.0213	0.0482	0.0498	0.0487	0.1061	0.1000	<u>0.1294</u>	0.1187	**0.1346**
	NDCG@20	0.0293	0.0532	0.0556	0.0585	0.1190	0.1131	<u>0.1402</u>	0.1264	**0.1470**

Table 2: Experimental results of TGCL4SR along with all baselines on four datasets. The best score and the second-best score of each row are bolded and underlined respectively.

Baselines. We compare TGCL4SR with two groups of representative and competitive SR baselines. The first group includes SR models without contrastive learning. GRU4Rec (Hidasi and Karatzoglou 2018) and SASRec (Kang and McAuley 2018) adopt RNN and Transformer respectively as sequence encoders. In addition, TiSASRec (Li, Wang, and McAuley 2020) adds a time-interval aware mechanism to SASRec. The second group contains SR models taking contrastive learning objectives as an auxiliary training task. CL4SRec (Xie et al. 2022) applies contrastive learning on augmented sequences for representation learning. DuoRec (Qiu et al. 2022) proposes a model-level augmentation and conducts contrastive learning between semantically similar sequences. TCPSRec (Tian et al. 2022) designs specific temporal pre-training objectives to learn interest patterns better. GCL4SR (Zhang et al. 2022) utilizes graph contrastive learning to obtain more informative representations. MStein (Fan et al. 2023) computes Wasserstein distance between augmented sequences as mutual information to achieve effective training.

Implementation Details. Our work is implemented by Pytorch. The sample parameters M and N are set as 2 and 20 respectively. We set the training batch size and all the embedding dimension sizes as 1024 and 64 respectively. The max length of user sequences is limited to 50. We set both the number of self-attention blocks and multi-heads for TiTConv and the temporal sequence encoder as 2. The scaling constant a is searched in $\{50, 100, 200, 400\}$ and c is set as 60000. Next, we set p as 0.5, and tune σ within $[0.01, 1]$. For TGCL, we tune τ and τ' within $[0.1, 1]$. Last, we search λ_1 within $[0.25, 1.5]$ stepping by 0.25, while λ_2 is selected from $\{0.05, 0.1, 0.2, 0.3, 0.5\}$. We repeat the experiments three times and report the average results. The experiments are conducted on a server with fifteen vCPU AMD EPYC 7543 32-Core Processors and one NVIDIA A40 GPU.

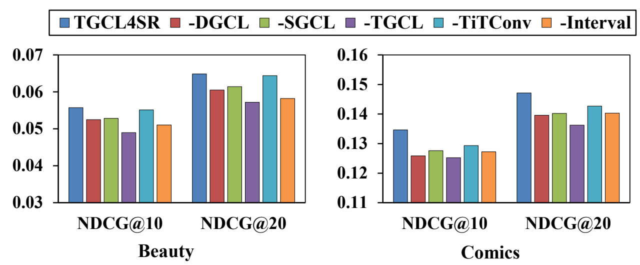
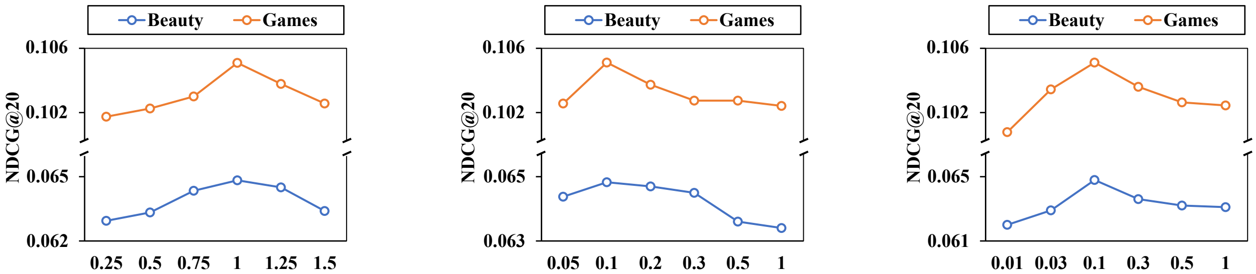


Figure 3: Performances of TGCL4SR with its variants on Beauty and Comics dataset.

5.2 Results and Analysis

Overall Comparison. Table 2 shows the overall evaluation results of TGCL4SR and other baseline models. Based on the results, we get key observations as follows: First, TGCL4SR performs best in most of the metrics over all datasets. It achieves improvements ranging from 1.08% to 12.4% over the best baseline, showing the effectiveness of our method. Second, TiSASRec and TGCL4SR outperform general SR methods in most cases, indicating the utility of temporal information for SR. Whereas TGCL4SR still performs better, since it utilizes absolute and relative time, and adds time disturbance to TITG. By contrast, TiSASRec only use time interval information on sequences. Thus our model exploits richer temporal information and lessens influences from noise simultaneously. Third, by contrastive learning, CL4SRec, GCL4SR and TGCL4SR perform better than general SR methods, yet TGCL4SR achieves further advancement. We attribute this to our designed temporal graph contrastive learning based on dual graph augmentation on the TITG. In comparison, CL4SRec augments each sequence separately, while GCL4SR simply enhances data by neighbor sampling on weighted transition



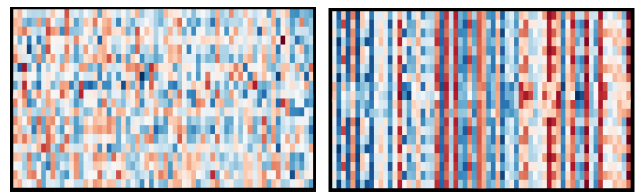
a) Hyper-parameter λ_1 b) Hyper-parameter λ_2 c) Hyper-parameter σ

Figure 4: TGCL4SR's performance comparison w.r.t different hyper-parameters in terms of NDCG@20.

graphs. Therefore, TGCL4SR obtains more refined information from contrastive learning than CL4SRec and GCL4SR. Last, TGCL4SR improves higher on Beauty, Games and Comics than on CDs dataset. A possible reason is that larger scale of sequences in CDs dataset alleviates the difficulty of transition pattern modeling, limiting the improvement from temporal information.

Ablation Study. To verify the effectiveness of key components in TGCL4SR, we conduct an ablation study on Beauty and Comics datasets with the following conditions: 1) -DGCL removing L_{DGCL}, 2) -SGCL removing L_{SGCL}, 3) -TGCL removing L_{TGCL}, 4) -TiTConv replacing TiTConv by GCN with stacked GraphSAGE, 5) -interval using the sequence encoder without adding the embeddings of time intervals. The corresponding evaluation results are illustrated in Figure 3. We can observe removing any key components causes the performance to drop off, indicating that all key components are useful for SR tasks. In specific, TGCL is the most significant since the model without it always performs worst. That's because SGCL and DGCL improve representation learning from different aspects of graphs. What's more, TiTConv achieves higher improvements on Comics than Beauty. A probable cause is that each item in Comics has more neighbors on average in TITG than Beauty, thus TiTConv can learn richer item transition patterns. Besides, time interval embeddings play an important role, because they directly influence the hidden representations of sequences generated by sequence encoders.

Parameter Sensitivity. We choose three important hyper-parameters λ_1, λ_2, and σ to study their influences on the TGCL4SR model. To control variables, when changing one of the parameters, we keep other parameters optimal. Figure 4 shows the consequential evaluation results on Beauty and Games dataset. λ_1 is the weight of TGCL loss, and λ_2 decides the intensity of MMD loss. We can see setting their values either too low or too high reduces the performance. Appropriate values of λ_1 and λ_2 can make two auxiliary objectives improve the model more impressively. Next, σ represents the intensity of added time perturbations. It's obvious that unsuitable σ can cause a decline in performance. The model will get less informative self-supervised signals if the disturbance is too light, while an excessive disturbance can hinder transition pattern learning.



(a) Neighbors of v_{11031} (b) Neighbors of v_{2654}

Figure 5: Timestamp lag influences representation consistency between neighbor items in TITG.

Case Study. We conduct a case study on Comics dataset to investigate whether item representations learned from TITG conform to item correlations. Specifically, We select an item with huge time differences with its neighbors (v_{11031}'s average timestamp lag with neighbors is 30787627.2) and another having slight ones with its neighbors (v_{2654}'s average timestamp lag with neighbors is 51.55). Then we visualize their neighbor item embeddings on the temporal item transition graph in Figure 5. This chart shows that neighbors have similar representations to the center node v_{2654} while v_{11031}'s neighbors are quite inconsistent. It demonstrates that TGCL4SR can effectively utilize temporal information and learn item correlations wisely. Additionally, note that adding time perturbation may be more useful to study the representation of v_{2654} than v_{11031}, due to the enormous average timestamp lag of the latter one.

6 Conclusion

In this paper, we proposed a novel sequential recommendation model named TGCL4SR, which learned user interest evolution on item transition graphs and their behavior sequences. Along this line, we designed the Temporal Item Transition Graph and made dual augmentation on it for better use of time information. Meanwhile, the graph neural network TiTConv was designed to capture item transition patterns effectively. Next, we proposed the temporal graph contrastive learning to enhance representation learning. We also applied time interval embeddings on the sequence encoder for comprehensive temporal information. The results of extensive experiments on four real-world datasets demonstrated the effectiveness of the proposed TGCL4SR.

Acknowledgements

This work was partially supported by National Natural Science Foundation of China (Grant No.92370204), Guangzhou-HKUST(GZ) Joint Funding Program (Grant No.2023A03J0008), Education Bureau of Guangzhou Municipality, and Project funded by China Postdoctoral Science Foundation (Grant No.2023M730785).

References

Cai, X.; Huang, C.; Xia, L.; and Ren, X. 2023. LightGCL: Simple Yet Effective Graph Contrastive Learning for Recommendation. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Chen, L.; Li, Z.; He, W.; Cheng, G.; Xu, T.; Yuan, N. J.; and Chen, E. 2022a. Entity summarization via exploiting description complementarity and salience. *IEEE Transactions on Neural Networks and Learning Systems*.

Chen, L.; Li, Z.; Wang, Y.; Xu, T.; Wang, Z.; and Chen, E. 2020. MMEA: entity alignment for multi-modal knowledge graph. In *Knowledge Science, Engineering and Management: 13th International Conference, KSEM 2020, Hangzhou, China, August 28–30, 2020, Proceedings, Part I 13*, 134–147. Springer.

Chen, L.; Li, Z.; Xu, T.; Wu, H.; Wang, Z.; Yuan, N. J.; and Chen, E. 2022b. Multi-modal siamese network for entity alignment. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, 118–126.

Dang, Y.; Yang, E.; Guo, G.; Jiang, L.; Wang, X.; Xu, X.; Sun, Q.; and Liu, H. 2023. Uniform Sequence Better: Time Interval Aware Data Augmentation for Sequential Recommendation. In Williams, B.; Chen, Y.; and Neville, J., eds., *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, 4225–4232. AAAI Press.

Fan, Z.; Liu, Z.; Peng, H.; and Yu, P. S. 2023. Mutual Wasserstein Discrepancy Minimization for Sequential Recommendation. In Ding, Y.; Tang, J.; Sequeda, J. F.; Aroyo, L.; Castillo, C.; and Houben, G., eds., *Proceedings of the ACM Web Conference 2023, WWW 2023, Austin, TX, USA, 30 April 2023 - 4 May 2023*, 1375–1385. ACM.

Fan, Z.; Liu, Z.; Zhang, J.; Xiong, Y.; Zheng, L.; and Yu, P. S. 2021. Continuous-time sequential recommendation with temporal graph collaborative transformer. In *Proceedings of the 30th ACM international conference on information & knowledge management*, 433–442.

Hamilton, W. L.; Ying, R.; and Leskovec, J. 2017. Inductive Representation Learning on Large Graphs. *CoRR*, abs/1706.02216.

Hassani, K.; and Khasahmadi, A. H. 2020. Contrastive multi-view representation learning on graphs. In *International conference on machine learning*, 4116–4126. PMLR.

He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 9729–9738.

He, R.; and McAuley, J. 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.

Hidasi, B.; and Karatzoglou, A. 2018. Recurrent neural networks with top-k gains for session-based recommendations. In *Proceedings of the 27th ACM international conference on information and knowledge management*, 843–852.

Kang, W.-C.; and McAuley, J. 2018. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*, 197–206. IEEE.

Li, J.; Wang, Y.; and McAuley, J. 2020. Time interval aware self-attention for sequential recommendation. In *Proceedings of the 13th international conference on web search and data mining*, 322–330.

Li, S.; Zhou, J.; Liu, J.; Xu, T.; Chen, E.; and Xiong, H. 2023. Multi-Temporal Relationship Inference in Urban Areas. In *Proceedings of the 29th ACM SIGKDD conference on knowledge discovery and data mining*, 1316–1327.

Li, S.; Zhou, J.; Xu, T.; Dou, D.; and Xiong, H. 2022. Geomgcl: Geometric graph contrastive learning for molecular property prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, 4541–4549.

Li, Y.; Swersky, K.; and Zemel, R. 2015. Generative moment matching networks. In *International conference on machine learning*, 1718–1727. PMLR.

Qiu, R.; Huang, Z.; Yin, H.; and Wang, Z. 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.

Sun, F.; Liu, J.; Wu, J.; Pei, C.; Lin, X.; Ou, W.; and Jiang, P. 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.

Tang, J.; and Wang, K. 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.

Tian, C.; Lin, Z.; Bian, S.; Wang, J.; and Zhao, W. X. 2022. Temporal Contrastive Pre-Training for Sequential Recommendation. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, CIKM '22, 1925–1934. New York, NY, USA: Association for Computing Machinery. ISBN 9781450392365.

Tran, V.; Salha-Galvan, G.; Sguerra, B.; and Hennequin, R. 2023. Attention Mixtures for Time-Aware Sequential Recommendation. In Chen, H.; Duh, W. E.; Huang, H.; Kato, M. P.; Mothe, J.; and Poblete, B., eds., *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2023, Taipei, Taiwan, July 23-27, 2023*, 1821–1826. ACM.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Wan, M.; Misra, R.; Nakashole, N.; and McAuley, J. 2019. Fine-Grained Spoiler Detection from Large-Scale Review Corpora. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2605–2610.

Wang, C.; Zhu, H.; Hao, Q.; Xiao, K.; and Xiong, H. 2021a. Variable interval time sequence modeling for career trajectory prediction: Deep collaborative perspective. In *Proceedings of the Web Conference 2021*, 612–623.

Wang, C.; Zhu, H.; Wang, P.; Zhu, C.; Zhang, X.; Chen, E.; and Xiong, H. 2021b. Personalized and explainable employee training course recommendations: A bayesian variational approach. *ACM Transactions on Information Systems (TOIS)*, 40(4): 1–32.

Wang, C.; Zhu, H.; Zhu, C.; Qin, C.; Chen, E.; and Xiong, H. 2023. SetRank: A Setwise Bayesian Approach for Collaborative Ranking in Recommender System. *ACM Transactions on Information Systems*, 42(2): 1–32.

Wang, S.; Hu, L.; Wang, Y.; Cao, L.; Sheng, Q. Z.; and Orgun, M. A. 2019. Sequential Recommender Systems: Challenges, Progress and Prospects. In Kraus, S., ed., *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, 6332–6338. ijcai.org.

Wang, Z.; Wei, W.; Cong, G.; Li, X.-L.; Mao, X.-L.; and Qiu, M. 2020. Global context enhanced graph neural networks for session-based recommendation. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*, 169–178.

Wu, J.; Cai, R.; and Wang, H. 2020. Déjà vu: A contextualized temporal attention mechanism for sequential recommendation. In *Proceedings of The Web Conference 2020*, 2199–2209.

Wu, J.; Wang, X.; Feng, F.; He, X.; Chen, L.; Lian, J.; and Xie, X. 2021. Self-supervised graph learning for recommendation. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, 726–735.

Wu, S.; Tang, Y.; Zhu, Y.; Wang, L.; Xie, X.; and Tan, T. 2019. Session-based recommendation with graph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, 346–353.

Xia, W.; Li, Y.; Tian, J.; and Li, S. 2021. Forecasting interaction order on temporal graphs. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 1884–1893.

Xie, X.; Sun, F.; Liu, Z.; Wu, S.; Gao, J.; Zhang, J.; Ding, B.; and Cui, B. 2022. Contrastive learning for sequential recommendation. In *2022 IEEE 38th international conference on data engineering (ICDE)*, 1259–1273. IEEE.

Xu, C.; Zhao, P.; Liu, Y.; Sheng, V. S.; Xu, J.; Zhuang, F.; Fang, J.; and Zhou, X. 2019. Graph contextualized self-attention network for session-based recommendation. In *IJCAI*, volume 19, 3940–3946.

Ye, Y.; Xia, L.; and Huang, C. 2023. Graph Masked Autoencoder for Sequential Recommendation. In Chen, H.; Duh, W. E.; Huang, H.; Kato, M. P.; Mothe, J.; and Poblete, B., eds., *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2023, Taipei, Taiwan, July 23-27, 2023*, 321–330. ACM.

Zhang, Y.; Liu, Y.; Xu, Y.; Xiong, H.; Lei, C.; He, W.; Cui, L.; and Miao, C. 2022. Enhancing Sequential Recommendation with Graph Contrastive Learning. In *IJCAI 2022*.