LS-TGNN: Long and Short-Term Temporal Graph Neural Network for Session-Based Recommendation

Zhonghong Ou[1*], Xiao Zhang[2], Yifan Zhu[2], Shuai Lyu[2], Jiahao Liu[3], Tu Ao[2]

[1] State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, China
[2] School of Computer Science, Beijing University of Posts and Telecommunications, China
[3]Meituan Inc.
{zhonghong.ou, xiao20010420, yifan_zhu, Lxb_savior}@bupt.edu.cn

Abstract

Session-Based Recommendation (SBR) based on Graph Neural Networks (GNN) has become a new paradigm for recommender systems, and plays a fundamental role in e-commerce and other relevant domains. Existing graph aggregation methods primarily form node representations by capturing basic relationships between neighboring and central nodes. Despite their encouraging results, the global relationships of items and user intentions within sessions typically change over time, which degrades the effectiveness of existing embedding schemes. To resolve this challenge, we propose a Long and Short-Term Temporal Graph Neural Network (LS-TGNN) for SBR. LS-TGNN employs a novel temporal session graph to aggregate neighborhood information, and models user interests from both long and short-term perspectives. Specifically, we design long-term and short-term encoders to model the long and short-term interests of users, respectively. In order to better model the interests of users in different time dimensions, we introduce an item-granularity method that distinguishes between long and short-term interests. Extensive experiments on three widely used datasets demonstrate that LS-TGNN outperforms existing methods with a large margin.

Introduction

Session-Based Recommendation (SBR) has become a focused recommendation scenario (Jannach, Ludewig, and Lerche 2017), where its essence lies in making recommendations in an anonymous and ad hoc session environment. SBR is more challenging than conventional recommendation tasks. In the rapidly evolving field of recommendation systems, accurately modeling user preferences is crucial for enhancing user experience and engagement. In recent years, methods based on Graph Neural Networks (GNN) have emerged (Liu et al. 2023). By introducing complex user-item interaction learning mechanisms, user preferences are modeled (Mao et al. 2023). Although these methods have made progress in handling complex relationships, they still have to explicitly model the spatiotemporal coupling between user purchases.

In session-based recommendation, the degree of relevance between items depends on their temporal sequence.

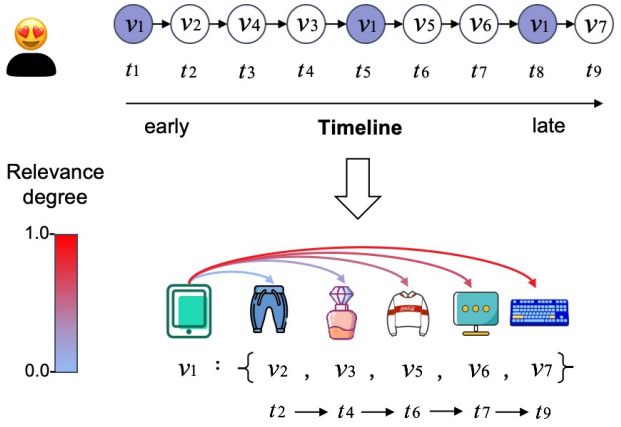

Figure 1: Changes in association pairs between an item and other items in a session.

As shown in Fig. 1, in a user session, taking v_1 as an example, we extract the first-order neighbors of v_1 items from the session data. Its neighbors also form a time series based on the order of their appearances in the original session. In constructing the representation of user interests, v_1 is more relevant to its most recently appearing neighbor v_7, than to earlier ones such as v_2.

To better harness the potential of evolving user interests, we propose to model the evolution of user interests over time from a new perspective. We name the scheme Long and Short-Term Temporal Graph Neural Network (LS-TGNN), which is a novel framework specifically designed to distinguish and model the two critical dimensions of user interests: the temporal scope (long and short-term interests) and the adaptability to evolving patterns. LS-TGNN not only distinguishes between short-term and long-term user interests, but also investigates the evolution and changes in user interests over time, thus resulting in more accurate and personalized recommendations. It proposes a unique architecture that integrates Temporal Graph Neural Network (TGNN) and self-supervised learning mechanisms to dynamically and precisely capture user preferences. The proposed model consists of multiple layers, each designed to address specific challenges in recommendation systems, including modeling long-term and short-term user interests,

*Corresponding author.

session representation learning, and the final prediction layer for next-click recommendations.

The contributions are summarized as follows:

- We propose an approach to achieve Long and Short-term (LS-term) interest disentanglement based on item granularity, which facilitates accurate modelling of users LS interests.
- We introduce a new temporal perspective model that effectively captures the dynamic migration of item neighborhood information over time in a session by constructing a Temporal Graph Neural Network (TGNN).
- We conduct extensive experiments on three widely-used datasets to verify the effectiveness of LS-TGNN. Experimental results demonstrate that LS-TGNN outperforms the state-of-the-art (SOTA) schemes by a large margin.

Preliminaries

Problem Statement

Let $V = \{v_1, v_2, v_3, \ldots, v_M\}$ represent all items, with M being the total number of items. An anonymous session S_q is represented by $S_q = \{v_{t1}, v_{t2}, \ldots, v_{tl}\}$. It refers to a series of interactions by the user in the current session in chronological order. All anonymous sessions are represented by $S' = \{S_1, S_2, \ldots, S_N\}$, with N being the total number of sessions. v_i is the ID of the item that the user clicks during the i-th interaction in the session S_q, and l is the length of S_q. Given a session S_q, session-based recommendation aims to recommend the top-K items ($1 \leq K \leq M$) from V that are most likely to be clicked by the user in the current session.

Temporal Session Graph Construction

In session-based recommendation, the transition between items reflects the evolution of user interests over time. We construct the session as a temporal graph to learn node representations. Given a session $S_q = \{v_{t1}, v_{t2}, \ldots, v_{tl}\}$, denote $\mathcal{G}_s = (V_s, E_s)$ as the corresponding session graph, where $V_s \subseteq V$ is the set of items that the user has clicked in session S_q. $E_s = \{e_{ij} \,|\, (v_i, v_j) \,|\, v_i \in S_q, v_j \in \mathcal{N}_q(v_i)\}$ denote the edge set, wherein $\mathcal{N}_q(v_i)$ represents the neighbor to the item v_i in S_q. To ensure the temporal order of each neighbor, for each item v_i in session S_q, we first add it to the edge set along with its left neighbor edge. An example of a temporal session graph construction is illustrated in Fig. 2. For a central node v_2 in the current session $\{v_1 \rightarrow v_2 \rightarrow v_3 \rightarrow v_2 \rightarrow v_4 \rightarrow v_5 \rightarrow v_2\}$, the set of its neighbors is $\{v_1, v_3, v_4, v_5\}$, which has a temporal order.

Item-Granularity LS-Term Interest Proxy

A user's interaction history S_q reflects both long and short-term interests (Liu et al. 2022). A recommender system first learns these interests from S_q, and then predicts future interactions based on both aspects (Jiang et al. 2024). To better learn long and short-term interests, we define the item-granularity long-short term interest proxy. Specifically, for v_i in a session S_q, its long-term proxy $p^u_{l_i} = \{\mathcal{N}_1(v_i), \mathcal{N}_2(v_i), \ldots, \mathcal{N}_q(v_i), \ldots, \mathcal{N}_N(v_i)\}$ includes all neighbors of v_i in S', and the short-term proxy $p^u_{s_i}$ is the last item in $\mathcal{N}_q(v_i)$.

LS-TGNN Network Structure

In this section, we describe the network architecture of LS-TGNN. We first introduce the LS-TGNN framework, we then present the three components of LS-TGNN: the user interest modeling layer, the disentanglement of long-short term interests layer, and the session representation learning layer. Finally, we describe the final prediction layer.

Framework of LS-TGNN

The overall framework of LS-TGNN is depicted in Fig. 2. LS-TGNN mainly consists of three components, i.e., ①, ②, and ③ in Fig. 2. The first component (①) models users' long-short term interests from two perspectives. The second component (②) disentangles users' long-short term interests by designing a long-short term interest proxy through a comparative learning approach. The third component (③) incorporates the user's long-short term interests into the final recommendation.

User Interest Modelling Layer

Motivated by recent studies (Shen et al. 2023; Li et al. 2023b) that separately learn LS-term interests using two different models, we propose to utilize two separate encoders, i.e., Long-term Encoder and Short-term Encoder, to capture these two aspects, respectively.

Long-term Interest Encoder. Long-term interests reflect an overall view of user preferences, and are relatively stable and less affected by recent interactions (Zheng et al. 2022). Thus, a user's long-term interests can be determined from the entire session of historical interactions. First, each item is embedded into a unified embedding space. Let $H = [h_{v_1}; h_{v_2}; \ldots; h_{v_l}]$, where $h_{v_i} = \text{Emb}(v_i) \in \mathbb{R}^d$ denotes the item embedding for item v_i, and $\text{Emb}(\cdot)$ is the item embedding look-up table and d is the dimension of the vectors. We also use a learnable position embedding matrix $P = [p_{v_1}; p_{v_2}; \ldots; p_{v_l}]$, where $p_{v_i} \in \mathbb{R}^d$ is a position vector for specific position i, and l is the length of the current session. Then, we combine the original embedding with position information as follows:

$$h^{'}_{v_i} = h_{v_i} + p_{v_i}. \tag{1}$$

Inspired by (Hou et al. 2022; Kang and McAuley 2018), we utilize L layers of self-attention modules to learn the latent relationships among session items.

$$F = \text{Transformers}([h'_{v_1}; h'_{v_2}; \ldots; h'_{v_n}]). \tag{2}$$

where $F \in \mathbb{R}^{l \times d}$ and d represents the output dimension of the feedforward network in the final layer of the self-attention module. Subsequently, we obtain the normalized weights $\alpha_1 \in \mathbb{R}^n$:

$$\alpha_1 = \text{softmax}(W_1 \odot F^\top). \tag{3}$$

where $W_1 \in \mathbb{R}^{d \times 1}$ is a learnable parameter, and $\odot$ indicates element-wise product. The final learned long-term interests

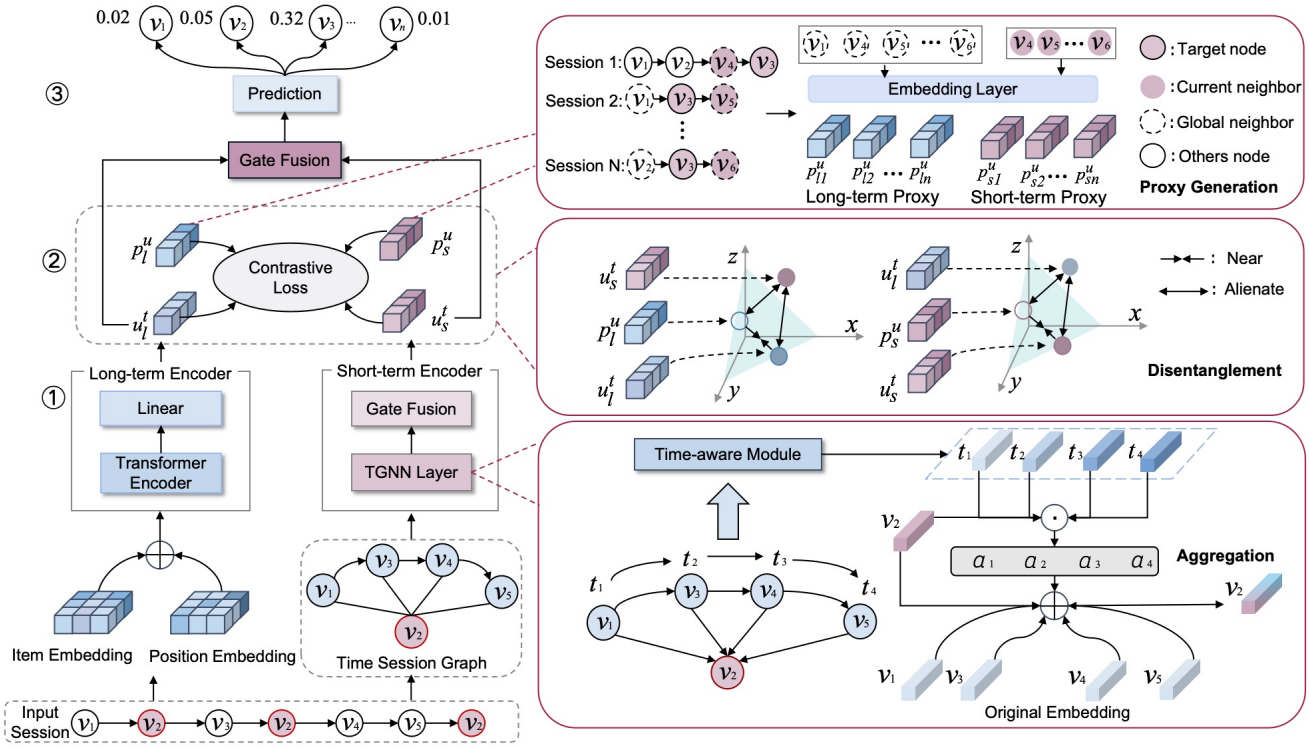

Figure 2: An overview of the proposed LS-TGNN: ① represents the user interest modeling stage, where user interests are modeled from both long-term and short-term perspectives. ② is the disentangling stage of long-term and short-term interests, guided by a self-supervised approach that directs the long-term and short-term encoders. ③ represents the final prediction stage.

representation is a weighted aggregation of the entire interaction history, with weights computed from the attentive network mentioned above and calculated as follows:

$$u_l = \alpha_1 \odot H. \tag{4}$$

Short-term Interest Encoder. Short-term interest crucially reflects immediate user tendencies and its temporal nature is vital. Nevertheless, existing studies (Liu et al. 2022) usually focus on the basic connections between session items and their neighbors, but ignore the temporal nature of the neighbor nodes themselves. This oversight can lead to several issues, particularly in dynamic network settings where user interests evolve. Neglecting the temporal aspects of neighboring nodes can result in models that fail to accurately track these changes. Moreover, the relevance and significance of certain neighbors may shift over time, a factor not considered by traditional methods. To address these challenges, we introduce a novel method that constructs a temporal session graph centered on session items to integrate neighbor nodes' temporal data, thereby enhancing item representation. Recognizing the time-sensitive nature of short-term interests, our approach prioritizes the latest item transitions. We employ a new layer using Temporal Graph Neural Networks (TGNN) to capture these transitions and underscore the importance of recent relational dynamics in accurately depicting short-term user interests.

As shown in Fig. 2, we create a temporal session graph for each session. For v_i $(v_i \in S_q)$ and its neighbor nodes v_j $(v_j \in \mathcal{N}_q(v_i))$, we form a time series $H' = [h'_1; h'_2; \ldots; h'_o]$ of neighbor nodes, where $h'_j = \text{Emb}(v_j) \in \mathbb{R}^d$ denotes the neighbor nodes embedding, and o is the number of neighbor nodes. Moreover, we propose a GRU-like (Chung et al. 2014) time-aware module to capture the variation of the neighbor sequence in the time dimension:

$$z^t_{s,j} = \sigma\left(W_z[H'_t||h_{v_i}] + U_z n^{t-1}_j + b_z\right). \tag{5}$$

$$r^t_{s,j} = \sigma\left(W_r[H'_t||h_{v_i}] + U_r n^{t-1}_j + b_r\right). \tag{6}$$

$$\tilde{n}^t_j = \tanh\left(W_o[H'_t||h_{v_i}] + U_h\left(r^t_{s,j} \odot n^{t-1}_j\right) + b_h\right). \tag{7}$$

$$n^t_j = \left(1 - z^t_{s,j}\right) \odot n^{t-1}_j + z^t_{s,j} \odot \tilde{n}^t_j. \tag{8}$$

where $W_z, W_r, W_o \in \mathbb{R}^{d\times 2d}$, $U_z, U_r, U_h \in \mathbb{R}^{d\times d}$ are learnable parameters, $\odot$ indicates element-wise product, and $||$ represents concatenation operation. To emphasise the importance of the central node, we add the embedding of the central node h_{v_i} at each time step of the input neighbor sequence. Finally, we acquire a representation n^t_j of the sequence of neighbors of each node through the time-aware module.

Instead of directly computing the similarity between the central node representation and the initial neighbor representation (Wang et al. 2020), we take into account the temporal migratory nature of neighbor information at the item granularity level. By calculating the similarity between the output of the time-aware module n^t_j and the central node

representation h_{v_i}, we differentiate the importance of different neighbors based on the temporal dimension. We employ item-aware attention to aggregate the representations of different neighbor nodes. The equations for attention weights between the central node and its neighbors are as follows:

$$\alpha_{i,j} = q_1^T \, LeakyReLU(W_2[(h_{v_i} \odot n_j^t)||W_{ij}]). \quad (9)$$

Here we choose LeakyRelu as the activation function, $W_{ij} \in \mathbb{R}^1$ is the weight of edge (v_i, v_j) in the temporal session graph, and $W_2 \in \mathbb{R}^{d+1 \times d+1}$ and $q_1 \in \mathbb{R}^{d+1}$ are trainable parameters. This approach makes information dissemination dependent on the affinity between h_{v_i} and n_j^t, which means that neighbors that match the preferences of the central node in the time dimension are preferred.

Then we normalize the coefficients across all neighbors connected with v_i by adopting the softmax function:

$$\alpha_{i,j} = \frac{exp(\alpha_{i,j})}{\sum_{v_k \in \mathcal{N}_q(v_i)} exp(\alpha_{i,j}))}. \quad (10)$$

As a result, the final attention score is capable of suggesting which neighbor nodes should be given more attention. We aggregate the neighbor nodes information by calculating the acquired attention score:

$$h_{\mathcal{N}_q(v_i)} = \sum_{v_j \in \mathcal{N}_q(v_i)} \alpha_{i,j} h'_j. \quad (11)$$

The final step involves aggregating the item representation h_{v_i} with its neighbor representation $h_{\mathcal{N}_q(v_i)}$. We design a gated aggregator as follows:

$$\alpha_2 = \sigma\left(W_3 \left[\frac{\sum_i^l h_{v_i}}{l} \middle\| \frac{\sum_i^l h_{\mathcal{N}_q(v_i)}}{l}\right]\right). \quad (12)$$

where $W_3 \in \mathbb{R}^{d \times 2d}$ is a trainable parameter to control information weight of item representation and neighbor representation. We calculate the final item representation as follows:

$$h_{v_i}^T = \alpha_2 * h_{v_i} + (1 - \alpha_2) * h_{\mathcal{N}_q(v_i)}. \quad (13)$$

The representations of items in the temporal session graph are aggregated based on the features of the item itself and its neighbors within the current session. Through the attention mechanism, the influence of noise on item representation learning is mitigated. Item representations, captured through TGNN, are utilized to represent the user's short-term interests u_s.

Disentanglement of LS-Term Interests Layer

As discussed previously, long-term interests offer a comprehensive overview of user preferences by summarizing the entire history of interactions. In contrast, short-term interests, reflecting recent interactions, evolve dynamically over time. Consequently, we derive proxies for long and short-term interests from interaction sequences to guide the two interest encoders.

Item-Granularity LS-Term Interest Proxy Generation. Unlike conventional schemes of generating session-grained long and short-term proxies (Zheng et al. 2022), in order to better disentangle long and short-term interests, we extract the long-term and short-term proxies separately for each item in a session. Specifically, we obtain the long-term proxy for the current item by computing the average of all the neighbor representations of each item over all historical interactions. The short-term interest proxy is obtained by using the last neighbor representation of the item in the current session. Formally, the long-term and short-term interest proxies for each item are computed as follows:

$$p_{li}^u = \frac{1}{C_i} Emb(\{\mathcal{N}_1(v_i), \mathcal{N}_2(v_i), \ldots, \mathcal{N}_N(v_i)\}). \quad (14)$$

$$p_{si}^u = Emb(\mathcal{N}_{qn}(v_i)). \quad (15)$$

where C_i indicates the number of neighbors of item v_i in the full history sessions, and $\mathcal{N}_{qn}(v_i)$ denotes the last neighbor of v_i in the current session.

Self-supervised Disentanglement of LS-Term Interests. With proxies as labels, we utilize them to disentangle long and short-term interests. Specifically, we focus on learning comparisons between encoder outputs and proxies. Our goal is to ensure that long-term and short-term interest representations are similar to their corresponding proxies, while staying away from representations that are opposite to them. We illustrate the contrasting learning tasks in ② of Fig. 2, in the form of two contrasting tasks as follows:

$$sim(u_s, p_s^u) > sim(u_s, p_l^u). \quad (16)$$

$$sim(u_l, p_l^u) > sim(u_l, p_s^u). \quad (17)$$

where Eq. (16) supervises short-term interests, Eq. (17) supervises long-term interests, and $sim(\cdot, \cdot)$ indicates embedding similarity. With these two comparison tasks, we are able to bring the representations of items that constitute short-term interests closer to short-term proxies in high-dimensional space. At the same time, we are able to bring the representations of items that constitute long-term interests closer to long-term proxies.

We leverage the triplet loss function to accomplish contrastive learning on Eq. (16) and Eq. (17). Formally, the loss function which uses the Euclidean distance to capture embedding similarity is calculated as follows:

$$L_{tri}(a, p, q) = max\{d(a, p) - d(a, q) + m, 0\}. \quad (18)$$

where d denotes the the Euclidean distance, and m denotes a positive margin value. L_{tri} is designed to make the anchor a more similar to the positive sample p than the negative sample q. Then the contrastive loss for self-supervised disentanglement of LS-term interests can be calculated as follows:

$$L_c = f(u_s, p_s^u, p_l^u) + f(u_l, p_l^u, p_s^u). \quad (19)$$

where f is L_{tri} or other loss function, e.g., Bayesian Personalized Ranking (BPR) (Rendle et al. 2012).

To summarize, we present two encoders with different temporal dimensions in order to learn representations of long-term and short-term interests. To achieve disentanglement of long-term and short-term interests, we compute

item-level interest proxies from historical interaction sequences. We further propose contrastive learning loss functions to guide the two encoders to learn their respective representations in a self-supervised manner.

Session Representation Learning.

Aggregating LS-term interest representations obtained through disentangling remains a challenge. Since LS-term interest representations are item-based, simple aggregation methods such as summation and concatenation assume that all items in the LS-term representations have the same contribution to next-click prediction, which are inappropriate in many cases. Adaptive fusion of LS-term interest representations is particularly important for next-click prediction. For long-term representation, we use the weights learned through the transformer α_1 to aggregate session items to form session-level representations, which are calculated as follows:

$$U_l = \sum_{i=1}^{l} \alpha_1 h_{v_i}. \tag{20}$$

For short-term representation, the most recently clicked item is more representative of the user's preference (Wu et al. 2019). Thus, we use $concatenation$ and $linear$ layer to combine reversed position information with item representations learnt from the TGNN layer.

$$z_i = \tanh(\, W_4[h^t_{v_i} || p_{v_{l-i+1}}] + b_1). \tag{21}$$

where $W_4 \in \mathbb{R}^{d\times 2d}$ and $b_1 \in \mathbb{R}^d$ are trainable parameters, $h^T_{v_i}$ indicates representations of items after TGNN. We represent session information by averaging the session item representations, i.e., $s' = \frac{1}{l}\sum_{i=1}^{l} h^T_{v_i}$. Similar to GCE-GNN (Wang et al. 2020), we employ a soft attention mechanism to acquire the weight for each item β_i:

$$\beta_i = q_2^T \sigma(W_5 z_i + W_6 s^{'} + b_2). \tag{22}$$

where W_5, $W_6 \in \mathbb{R}^{d\times d}$, and $q_2, b_2 \in \mathbb{R}^d$ are learnable parameters. The short-term session representation can be obtained by linearly combining the item representations:

$$U_s = \sum_{i=1}^{l} \beta_i h^T_{v_i}. \tag{23}$$

Finally, we acquire LS-term session representations that represent the different interests of users at different times through different dimensions. We fuse LS-term session representations through a gated aggregator similar to Eq. (12) and Eq. (13), which is listed as follows:

$$\alpha_3 = \sigma(W_6[U_l || U_s]). \tag{24}$$

where $W_6 \in \mathbb{R}^{d\times 2d}$ is a trainable parameter to control information weight of LS-term session representations. We aggregate the LS-term session representations to form the final session representation by α_3 as follows:

$$S = \alpha_3 * U_s + (1-\alpha_3) * U_l. \tag{25}$$

Prediction Layer

Based on the acquired session representation S, the final recommendation probability of each candidate item is calculated by combining their initial embedding and current session representation. This process is typically achieved through a dot product operation to obtain the click probability for all items:

$$x_i = S^T h_{v_i}. \tag{26}$$

We normalize the scores of all items using a softmax layer. Similar to (Pan et al. 2020), we introduce a scaling coefficient τ to control data scaling and promote model convergence. The final score $\hat{y_i}$ is presented as follows:

$$\hat{y_i} = \frac{exp(\tau x_i)}{\sum_i exp(\tau x_i)}. \tag{27}$$

where $\hat{y_i}$ denotes the probability of item v_i appearing as the next-click in the current session.

To train the main task, we employ cross-entropy as the optimization objective to learn the parameters:

$$L_{main} = -\sum_{i\in V} y_i log(\hat{y}_i) + (1-y_i) log(1-\hat{y}_i). \tag{28}$$

wherein y_i denotes the one-hot encoding vector of the ground truth items. Finally, we acquire the overall loss function L of the model as follows:

$$L = L_{main} + \lambda L_c. \tag{29}$$

where λ is a hyper-parameter to control the scale of the self-supervised disentanglement of LS-term interests.

Experimental Settings

Datasets

To evaluate the effectiveness of LS-TGNN, we conduct experiments on three widely-used datasets: $Tmall$[1], $RetailRocket$[2], and $Diginetica$[3]. The Tmall dataset comes from the IJCAI-15 competition, which contains anonymous user shopping logs on Tmall online shopping platform. RetailRocket is a dataset on a Kaggle contest released by an e-commerce company, which contains user browsing activity within six months. Diginetica is a dataset from CIKM Cup 2016 and contains music listening behavior of users. The statistics of the datasets are presented in Table 1.

Similar to existing studies (Wang et al. 2020; Wu et al. 2019), we augment and annotate the training and test datasets by using a sequence splitting method. It generates multiple labeled sequences with the corresponding labels $([v_{t_1}], v_{t_2}), ([v_{t_1}, v_{t_2}], v_{t_3})..., ([v_{t_1}, v_{t_2}, ..., v_{t_{l-1}}], v_{t_l})$ for every session $S = \{v_{t_1}, v_{t_2}, ..., v_{t_l}\}$. Note that the label of each sequence is the last click item in it. We consider Precision@20 (P@20 for short) and MRR@20 as the metrics, all of which are widely adopted for evaluation (Wang et al. 2020; Wu et al. 2019; Xia et al. 2021a).

[1]https://tianchi.aliyun.com/dataset/42

[2]https://www.kaggle.com/datasets/retailrocket/ecommerce-dataset

[3]https://competitions.codalab.org/competitions/11161

Statistics	Tmall	Retailrocket	Diginetica
#clicks	818,479	1,085,217	982,961
#train sessions	351,268	433,643	719,470
#test sessions	25,898	15,132	60,858
#items	40,728	36,968	43,097
Average session length	6.69	5.43	5.12

Table 1: Statistics of the datasets, including Tmall, Retailrocket, and Diginetica.

Baselines

We compare the proposed LS-TGNN with the state-of-the-art (SOTA) session-based recommendation methods, including RNN-based models, e.g., GRU4REC (Hidasi et al. 2015) and NARM (Li et al. 2017), attention-based models, e.g., STAMP (Liu et al. 2018) and SRGNN (Wu et al. 2019), and graph-based approaches including GCE-GNN (Wang et al. 2020), S^2-DHCN (Xia et al. 2021c), and COTREC (Xia et al. 2021b). Additionally, we consider CORE (Hou et al. 2022), MGIR (Han et al. 2022), and SPARE (Peintner, Mohammadi, and Zangerle 2023) as the SOTA baselines. Detailed description of these baselines can be found in the supplementary materials.

Implementation Details

We fix the dimension of latent vectors to 256, and set the minimum batch size for all models to 256, and fix the λ to 0.1. Following the recommendations from the original studies, we carefully tune the other hyper-parameters of the baseline models and report their performance under the best settings. For LS-TGNN, all parameters are initialized using a Gaussian distribution with a mean of 0 and a standard deviation of 0.1. We employ the Adam optimizer with an initial learning rate of 0.001, decay it by a factor of 0.1 every 3 epochs, and set the L2 penalty term coefficient to 10^{-5}. Additionally, we set the number of neighbors for each item in the temporal session graph to 8, and globally to 100. We conduct all the experiments on Nvidia GeForce RTX 3090 GPU.

Experimental Results

Overall Performance

In this section, we compare LS-TGNN with the SOTA baselines to verify its effectiveness. We boldface the best results and underline the second-best results. Experimental results of all methods are listed in Table 2. Improv. (%) represents the percentage of improvement that LS-TGNN achieves relative to the best result among the baseline models. From the table, we make three important observations.

First, graph-based baseline approaches (e.g., GCE-GNN and SR-GNN) perform well in modelling session data, compared to RNN-based approaches (e.g., NARM and STAMP), which highlights the potential of GNNs in this domain. Particularly noteworthy that GCE-GNN exhibits better performance than SR-GNN, which suggests that fusing different levels of information (both local and global) can effectively improve the accurate prediction of user intent in session-based recommender systems. In addition, the richness of cross-session information is further exploited through unsupervised task-assisted learning (e.g., S^2-DHCN and COTREC), which further improves performance of the model.

Method	Diginetica		Retailrocket		Tmall	
	P@20	MRR@20	P@20	MRR@20	P@20	MRR@20
GRU4REC	29.45	8.33	44.01	23.67	10.93	5.89
NARM	49.70	16.17	50.22	24.59	23.30	10.70
STAMP	45.64	14.32	50.96	25.17	26.47	13.36
SR-GNN	51.26	17.66	50.32	26.57	27.57	13.72
GCE-GNN	<u>54.22</u>	19.04	54.58	28.09	33.42	15.42
S^2-DHCN	53.18	18.44	53.66	27.30	31.42	15.05
COTREC	54.18	<u>19.07</u>	56.17	29.97	36.35	18.04
CORE	52.84	18.47	56.68	30.09	39.16	19.52
MGIR	53.73	18.77	56.62	29.84	36.41	17.42
SPARE	54.08	18.59	<u>56.91</u>	<u>30.22</u>	<u>39.28</u>	<u>20.07</u>
LS-TGNN	**55.10**	**19.47**	**58.01**	**30.74**	**42.61**	**20.33**
Improv.(%)	1.62	2.09	1.93	1.72	8.47	1.30

Table 2: Performance(%) comparison of LS-TGNN against the baselines. The best results are boldfaced and the second-best results are underlined.

Second, SPARE (Peintner, Mohammadi, and Zangerle 2023) demonstrates outstanding performance across three datasets, particularly excelling in the Tmall dataset. It indicates that explicitly modeling multi-hop information aggregation mechanisms via shortest-path edges across multiple layers is effective for session-based recommendation systems. Moreover, CORE (Hou et al. 2022) achieves the best performance on most metrics among all baseline models, highlighting the necessity of matching users and items within a consistent representation space and underscoring the advantages of lightweight models in terms of both performance and efficiency.

Third, as shown in Table 2, our proposed LS-TGNN outperforms all baselines, especially in the Tmall dataset, where it surpasses the best baseline by 8.47% in terms of Precision@20. We attribute these improvements to our approach of modeling user interests from both long-term and short-term perspectives and proposing a temporal session graph construction method. Additionally, unlike the neighbor information aggregation method proposed in GCE-GNN (Wang et al. 2020), we capture the temporal transitions of neighbor information to model the relevance to the central node, thereby effectively aggregating information. We also employ a long-short term disentangling method at the item granularity to form item node representations.

Ablation Study

To demonstrate the effectiveness of the key components in LS-TGNN, we conduct an ablation study on the Diginetica, Tmall, and Retailrocket datasets under the following conditions: (1) -LIE: removing the Long-term Interest Encoder, (2) -SIE: removing the Short-term Interest Encoder, and (3) -

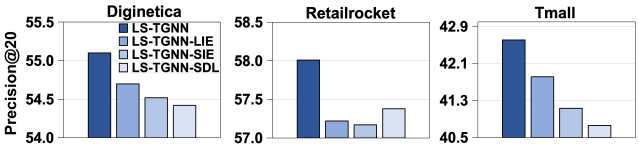

Figure 3: Performances of LS-TGNN with its variants on Diginetica, Retailrocket, and Tmall datasets.

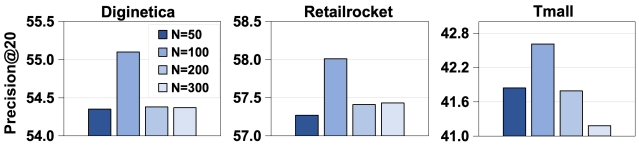

Figure 4: Performances of LS-TGNN with different global number of neighbors for each item on Diginetica, Retailrocket, and Tmall datasets.

SDL: removing the Self-supervised Disentangling Loss. The specific results are illustrated in Fig. 3. It is observed that removing any component results in a performance drop, especially when the Short-term Interest Encoder is removed. This highlights that users' short-term interests are crucial for predicting the next click, and the construction of the temporal session graph plays a vital role in modeling users' short-term interests. We also conduct the same experiments on MRR@20, and the conclusions are consistent. Nevertheless, due to space limit, they are not shown here.

Parameter Sensitivity

We select the global number of neighbors N_1 for each item, and the number of neighbors N_2 in the temporal session graph for each item as two critical hyperparameters to study their impact on the overall performance. We conduct experiments on three datasets, and the specific results are illustrated in Fig. 4 and Fig. 5. From the figures, we can see that too small or too large values of N_1 and N_2 negatively affect the model's performance. We conclude that if the number of neighbors for an item is too large, the model tends to focus more on the information from the neighboring nodes while losing the item's own characteristics. Conversely, if the number of neighbors is too small, it fails to capture sufficient neighborhood information, thus affecting performance.

Related Work

GNN Session-based Recommendation

Graph Neural Networks (GNNs) are a powerful representation learning method with outstanding generalization abilities (Xu et al. 2020). Recently, GNNs have been utilized to capture complex transfer relationships in session-based recommendations. GCE-GNN (Wang et al. 2020) leverages global-level item transitions across all sessions to learn about global-level neighbor information. The aforementioned studies demonstrate good performance in mining item sessions and global transition relationships. MAE (Ye, Xia, and Huang 2023) addresses label scarcity and data

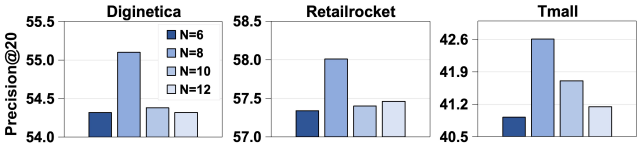

Figure 5: Performances of LS-TGNN with different number of neighbors in the temporal session graph for each item on Diginetica, Retailrocket, and Tmall datasets.

noise in sequential recommendation by dynamically distilling global item transitions for self-supervised augmentation. Disen-GNN (Li et al. 2023a) decomposes item embeddings into multiple factors through disentangled learning and uses Gated Graph Neural Network (GGNN) to learn embeddings for each factor based on the similarity matrix between factors. Nevertheless, most studies employ fixed neighbor information aggregation or leverage inherent session features, often neglecting temporal variations in neighbor information (Jiang et al. 2023). This limitation restricts the representation capabilities of items and impacts the matching process between users and candidate items.

GNN for Temporal Modeling

Integration of temporal information into graph representation learning is increasingly receiving attention, highlighting its critical role in session prediction. A common approach is to use a time-aware encoder to capture the dynamic information of the graph and embed it into the node encodings (Kumar, Zhang, and Leskovec 2019). These time-aware encoders can be designed to synthesize node representations at different times along with the GNN modules (Zhu et al. 2023). For instance, ST-GNN (Lakmal et al. 2024) utilizes Gated Recurrent Units (GRU) and Graph Isomorphism Networks (GIN) to capture temporal and spatial characteristics. DGNN (Manessi, Rozza, and Manzo 2020) introduces an LSTM layer after the graph convolutional layer to sequentially propagate node features. TP-GNN (Liu et al. 2024) uses a message passing technique with information flows between nodes to capture long-term temporal dependencies. But these methods have difficulties in modeling user sessions due to short sequences and time intervals.

Conclusion

In this paper, we proposed a Long and Short-Term Temporal Graph Neural Network (LS-TGNN), a novel framework designed to disentangle and model the long-term and short-term user interests in session-based recommendation systems. LS-TGNN addresses the limitations of existing methods by incorporating temporal graph neural networks and self-supervised learning mechanisms, which effectively capture the dynamic evolution of user interests over time. Experiments on three datasets prove LS-TGNN's effectiveness. Future research may explore multi-modal data integration to improve its prediction accuracy.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant No.62076035 and SMP-Zhipu.AI Large Model Cross-Disciplinary Fund.

References

Chung, J.; Gülçehre, Ç.; Cho, K.; and Bengio, Y. 2014. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *CoRR*, abs/1412.3555.

Han, Q.; Zhang, C.; Chen, R.; Lai, R.; Song, H.; and Li, L. 2022. Multi-Faceted Global Item Relation Learning for Session-Based Recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1705–1715.

Hidasi, B.; Karatzoglou, A.; Baltrunas, L.; and Tikk, D. 2015. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939*.

Hou, Y.; Hu, B.; Zhang, Z.; and Zhao, W. X. 2022. CORE: Simple and Effective Session-based Recommendation within Consistent Representation Space. In Amigó, E.; Castells, P.; Gonzalo, J.; Carterette, B.; Culpepper, J. S.; and Kazai, G., eds., *SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, July 11 - 15, 2022*, 1796–1801. ACM.

Jannach, D.; Ludewig, M.; and Lerche, L. 2017. Session-based item recommendation in e-commerce: on short-term intents, reminders, trends and discounts. *User Model. User Adapt. Interact.*, 27(3-5): 351–392.

Jiang, L.; Wang, Y.; Wang, J.; Wang, P.; and Yin, M. 2023. Multi-View MOOC Quality Evaluation via Information-Aware Graph Representation Learning. In *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI*, 8070–8077. AAAI Press.

Jiang, L.; Xiao, Y.; Zhao, X.; Xu, Y.; Hu, S.; Wang, P.; and Yin, M. 2024. Hierarchical Reinforcement Learning on Multi-Channel Hypergraph Neural Network for Course Recommendation. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI*, 2099–2107.

Kang, W.; and McAuley, J. J. 2018. Self-Attentive Sequential Recommendation. In *IEEE International Conference on Data Mining, ICDM 2018, Singapore, November 17-20, 2018*, 197–206. IEEE Computer Society.

Kumar, S.; Zhang, X.; and Leskovec, J. 2019. Predicting Dynamic Embedding Trajectory in Temporal Interaction Networks. In Teredesai, A.; Kumar, V.; Li, Y.; Rosales, R.; Terzi, E.; and Karypis, G., eds., *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*, 1269–1278. ACM.

Lakmal, D.; Perera, K.; Borovica-Gajic, R.; and Karunasekera, S. 2024. Spatial-Temporal Bipartite Graph Attention Network for Traffic Forecasting. In Yang, D.; Xie, X.; Tseng, V. S.; Pei, J.; Huang, J.; and Lin, J. C., eds., *Advances in Knowledge Discovery and Data Mining - 28th Pacific-Asia Conference on Knowledge Discovery and Data Mining, PAKDD 2024, Taipei, Taiwan, May 7-10, 2024, Proceedings, Part II*, volume 14646 of *Lecture Notes in Computer Science*, 68–80. Springer.

Li, A.; Cheng, Z.; Liu, F.; Gao, Z.; Guan, W.; and Peng, Y. 2023a. Disentangled Graph Neural Networks for Session-Based Recommendation. *IEEE Trans. Knowl. Data Eng.*, 35(8): 7870–7882.

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

Li, M.; Zhang, Z.; Zhao, X.; Wang, W.; Zhao, M.; Wu, R.; and Guo, R. 2023b. AutoMLP: Automated MLP for Sequential Recommendations. 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*, 1190–1198. ACM.

Liu, H.; Xu, Z.; Zhang, Q.; and Tang, Y. 2022. Integrating Users' Long- and Short-Term Preferences for Session-based Recommendation. In *25th IEEE International Conference on Computer Supported Cooperative Work in Design, CSCWD 2022, Hangzhou, China, May 4-6, 2022*, 611–616. IEEE.

Liu, J.; Liu, J.; Zhao, K.; Tang, Y.; and Chen, W. 2024. TP-GNN: Continuous Dynamic Graph Neural Network for Graph Classification. In *40th IEEE International Conference on Data Engineering, ICDE 2024, Utrecht, The Netherlands, May 13-16, 2024*, 2848–2861. IEEE.

Liu, Q.; Zeng, Y.; Mokhosi, R.; and Zhang, H. 2018. STAMP: short-term attention/memory priority model for session-based recommendation. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 1831–1839.

Liu, Z.; Li, Y.; Chen, N.; Wang, Q.; Hooi, B.; and He, B. 2023. A Survey of Imbalanced Learning on Graphs: Problems, Techniques, and Future Directions. *CoRR*, abs/2308.13821.

Manessi, F.; Rozza, A.; and Manzo, M. 2020. Dynamic graph convolutional networks. *Pattern Recognit.*, 97.

Mao, Q.; Liu, Z.; Liu, C.; and Sun, J. 2023. HINormer: Representation Learning On Heterogeneous Information Networks with Graph Transformer. 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*, 599–610. ACM.

Pan, Z.; Cai, F.; Chen, W.; Chen, H.; and de Rijke, M. 2020. Star Graph Neural Networks for Session-based Recommendation. In d'Aquin, M.; Dietze, S.; Hauff, C.; Curry, E.; and Cudré-Mauroux, P., eds., *CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19-23, 2020*, 1195–1204. ACM.

Peintner, A.; Mohammadi, A. R.; and Zangerle, E. 2023. SPARE: Shortest Path Global Item Relations for Efficient Session-based Recommendation. In Zhang, J.; Chen, L.; Berkovsky, S.; Zhang, M.; Noia, T. D.; Basilico, J.; Pizzato,

L.; and Song, Y., eds., *Proceedings of the 17th ACM Conference on Recommender Systems, RecSys 2023, Singapore, Singapore, September 18-22, 2023*, 58–69. ACM.

Rendle, S.; Freudenthaler, C.; Gantner, Z.; and Schmidt-Thieme, L. 2012. BPR: Bayesian Personalized Ranking from Implicit Feedback. *CoRR*, abs/1205.2618.

Shen, Q.; Wen, H.; Zhang, J.; and Rao, Q. 2023. Hierarchically Fusing Long and Short-Term User Interests for Click-Through Rate Prediction in Product Search. *CoRR*, abs/2304.02089.

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, S.; Tang, Y.; Zhu, Y.; Wang, L.; Xie, X.; and Tan, T. 2019. Session-Based Recommendation with Graph Neural Networks. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, 346–353. AAAI Press.

Xia, X.; Yin, H.; Yu, J.; Shao, Y.; and Cui, L. 2021a. Self-Supervised Graph Co-Training for Session-based Recommendation. In Demartini, G.; Zuccon, G.; Culpepper, J. S.; Huang, Z.; and Tong, H., eds., *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021*, 2180–2190. ACM.

Xia, X.; Yin, H.; Yu, J.; Shao, Y.; and Cui, L. 2021b. Self-supervised graph co-training for session-based recommendation. In *Proceedings of the 30th ACM International conference on information & knowledge management*, 2180–2190.

Xia, X.; Yin, H.; Yu, J.; Wang, Q.; Cui, L.; and Zhang, X. 2021c. Self-Supervised Hypergraph Convolutional Networks for Session-based Recommendation. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, 4503–4511. AAAI Press.

Xu, H.; Huang, C.; Xu, Y.; Xia, L.; Xing, H.; and Yin, D. 2020. Global Context Enhanced Social Recommendation with Hierarchical Graph Neural Networks. In *2020 IEEE International Conference on Data Mining (ICDM)*, 701–710.

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.

Zheng, Y.; Gao, C.; Chang, J.; Niu, Y.; Song, Y.; Jin, D.; and Li, Y. 2022. Disentangling Long and Short-Term Interests for Recommendation. In Laforest, F.; Troncy, R.; Simperl, E.; Agarwal, D.; Gionis, A.; Herman, I.; and Médini, L., eds., *WWW '22: The ACM Web Conference 2022, Virtual Event, Lyon, France, April 25 - 29, 2022*, 2256–2267. ACM.

Zhu, Y.; Cong, F.; Zhang, D.; Gong, W.; Lin, Q.; Feng, W.; Dong, Y.; and Tang, J. 2023. WinGNN: Dynamic Graph Neural Networks with Random Gradient Aggregation Window. In Singh, A. K.; Sun, Y.; Akoglu, L.; Gunopulos, D.; Yan, X.; Kumar, R.; Ozcan, F.; and Ye, J., eds., *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2023, Long Beach, CA, USA, August 6-10, 2023*, 3650–3662. ACM.