Self-supervised graph learning with target-adaptive masking for session-based recommendation

Yitong WANG[§], Fei CAI[‡], Zhiqiang PAN, Chengyu SONG[§]

Science and Technology on Information Systems Engineering Laboratory,

National University of Defense Technology, Changsha 410073, China

E-mail: wangyitong20@nudt.edu.cn; caifei08@nudt.edu.cn; panzhiqiang@nudt.edu.cn; songchengyu@nudt.edu.cn

Abstract: Session-based recommendation aims to predict the next item based on a user's limited interactions within a short period. Existing approaches use mainly recurrent neural networks (RNNs) or graph neural networks (GNNs) to model the sequential patterns or the transition relationships between items. However, such models either ignore the over-smoothing issue of GNNs, or directly use cross-entropy loss with a softmax layer for model optimization, which easily results in the over-fitting problem. To tackle the above issues, we propose a self-supervised graph learning with target-adaptive masking (SGL-TM) method. Specifically, we first construct a global graph based on all involved sessions and subsequently capture the self-supervised signals from the global connections between items, which helps supervise the model in generating accurate representations of items in the ongoing session. After that, we calculate the main supervised loss by comparing the ground truth with the predicted scores of items adjusted by our designed target-adaptive masking module. Finally, we combine the main supervised component with the auxiliary self-supervision module to obtain the final loss for optimizing the model parameters. Extensive experimental results from two benchmark datasets, Gowalla and Diginetica, indicate that SGL-TM can outperform state-of-the-art baselines in terms of Recall@20 and MRR@20, especially in short sessions.

Key words: Session-based recommendation; Self-supervised learning; Graph neural networks; Target-adaptive masking

1 Introduction

As an effective way to tackle the information overload issue, recommender systems (RSs) can help people make accurate choices when facing abundant products or services (Singhal et al., 2017; Wang XN and Tan, 2020). Traditional approaches like collaborative filtering mainly pay attention to the long-term preference of users while neglecting their current interaction patterns (Wang X et al., 2019). This may lead to inaccurate recommendations in some real-world situations, where a user's long-term interactions are not available (Wang SJ et al., 2022). Thus, session-based recommendation has been proposed to generate recommendations by modeling the user's recent limited interactions (Hidasi et al., 2016; Wu et al., 2019).

Existing methods apply mainly Markov chains (MCs) to extract the sequential patterns between adjacent interactions for session-based recommendation (Rendle et al., 2010). Moreover, considering the advantage of recurrent neural networks (RNNs) in sequential data modeling, Hidasi et al. (2016) first used the gate recurrent unit (GRU) to model the session sequences and proposed session-parallel mini-batches to modify the basic GRU. In addition,

---

[§] These two authors contributed equally to this work

[‡] Corresponding author

ⓘ ORCID: Yitong WANG, https://orcid.org/0000-0003-4020-5767; Fei CAI, https://orcid.org/0000-0002-5709-1682; Chengyu SONG, https://orcid.org/0000-0001-7020-7603

Li J et al. (2017) applied the attention mechanism to determine the importance of items, which was then used to generate the user's main intent. Recently, benefiting from the capability of graph neural networks (GNNs) in modeling complex transition relations between items, GNN-based methods have been widely investigated and have achieved remarkable achievements on session-based recommendation (Chen TW and Wong, 2020; Qiu et al., 2020). For instance, Wang ZY et al. (2020) introduced the global context enhanced GNN (GCE-GNN) to aggregate the item representations from both the session graph and the global graph through a soft attention mechanism.

However, MC- and RNN-based methods fail to take the complicated transition patterns between items into account (Xia et al., 2021b). Furthermore, despite the fact that GNN-based methods have achieved outstanding success, the following limitations still remain. On one hand, deep GNNs may lead to an over-smoothing issue, in which the learned representations of adjacent items in the session are highly similar, making the item features indistinguishable (Chen M et al., 2020). On the other hand, the training objective of existing models is to minimize the cross-entropy loss. In the training process, the target item score will continually increase while other items, including the neighbors of the target item, are treated as negative samples with decreasing scores. However, items adjacent to each other in the graph should be similar; i.e., it is unreasonable that the scores of the neighbors of the target item are continuously decreased. Thus, it leads to an over-fitting problem and model performance degradation.

To tackle the aforementioned issues, in this study, we propose the self-supervised graph learning with target-adaptive masking (SGL-TM) method for session-based recommendation. Specifically, we first construct the global connections between the items of all involved sessions, and then obtain the neighbor and non-neighbor information of each item as the self-supervised signals for enhancing item representation. Subsequently, given a current session, we apply the graph attention network (GAT) to generate the representations of items in the session. After that, we combine the long- and short-term interests of the current session to obtain the user preference. Then, for each session, we employ the designed target-adaptive masking module to remove

the neighbors of the target item for calculating the main supervised loss, which is then combined with the auxiliary self-supervised loss for model optimization. We implement the comprehensive experiments on two real-world datasets, Gowalla and Diginetica. The results demonstrate that our SGL-TM outperforms state-of-the-art baselines in terms of both Recall@20 and MRR@20.

Generally, the main contributions of this paper can be summarized as follows:

1. We propose a neoteric self-supervised framework for session-based recommendation, which can solve the over-smoothing problem using the global connections between items in graph learning.

2. We design a target-adaptive masking module to tackle the over-fitting issue resulting from the cross-entropy loss with a softmax layer in the model optimization.

3. The results of the experiments implemented on two public, real-world datasets, Gowalla and Diginetica, indicate that our proposed SGL-TM outperforms the competitive baselines in terms of Recall@20 and MRR@20.

2 Related works

In this section, we introduce the related works from two aspects, session-based recommendation and self-supervised recommendation.

2.1 Session-based recommendation

Conventional approaches for session-based recommendation typically use Markov chains to capture the temporal patterns between the user's adjacent interactions. For instance, Rendle et al. (2010) combined matrix factorization (MF) with Markov chains to simultaneously capture the sequential patterns and the user's long-term preference in the session. Moreover, Shani et al. (2005) adopted Markov decision processes to use the transition dependencies between items for generating recommendations.

Recently, neural models such as convolutional neural networks (CNNs) and RNNs have been widely used in session-based recommendation. For instance, Yuan et al. (2019) adopted CNNs to learn item representations from both long-term and current item dependencies, where the interactions in a session were regarded as an image and the convolution operation was applied to generate the item

embeddings. Moreover, Hidasi et al. (2016) introduced GRU4REC, which first employs RNNs to model the whole session sequence and introduces a session-parallel training mechanism for model optimization. Furthermore, Tan et al. (2016) proposed multiple optimization techniques for enhancing GRU4REC, such as data augmentation and generalized distillation, to make item prediction accurate. In addition, to emphasize the user's main intent in the current session, Li J et al. (2017) introduced the neural attentive recommendation model based on a hybrid encoder to concurrently take the sequential behavior and the user's main intent into consideration. Besides, Liu et al. (2018) proposed a short-term memory priority model for capturing both the long- and short-term interests of the user.

Considering the powerful ability of GNNs in modeling connections between items, GNNs have been introduced into session-based recommendation and achieved considerable performance. Specifically, the current session is first transformed to a session graph, where GNNs are applied to propagate information between items for learning accurate item representations. For example, Wu et al. (2019) introduced session-based recommendation with GNNs (SR-GNN), which took the lead in adopting GNNs to learn item embeddings, where incoming and outgoing matrices were adopted to reveal the transition pattern between items. On the basis of SR-GNN, Xu et al. (2019) further used a self-attention mechanism to extract the long-term patterns between items in the session. Furthermore, Qiu et al. (2020) employed FGNN+ to connect different sessions using a global graph for capturing cross-session information to generate accurate recommendations. Moreover, Wang ZY et al. (2020) proposed to compute the information flow from both the global and session levels by relying on the constructed global graph and session graph, respectively. In addition, Chen TW and Wong (2020) proposed a shortcut graph attention layer with a lossless encoding scheme to tackle the information loss issue of GNN applications in session-based recommendation.

However, the CNN- and RNN-based approaches fail to take the complicated transition patterns between items into account, limiting the recommendation performance. Moreover, although the GNN-based methods have achieved outstanding achievements in session-based recommendation, they generally face a serious over-smoothing issue. That is, GNNs lead to significant similarities among the item representations learned in a session, which results in inaccurate recommendations.

2.2 Self-supervised recommendation

Self-supervised learning (SSL) (Hjelm et al., 2019) is a neoteric paradigm of machine learning that mines the pseudo labels from data to supervise models and uses the learned representations for various downstream tasks. SSL has achieved great accomplishments in natural language processing (Kong et al., 2020; Zhang et al., 2020), computer vision (Afouras et al., 2020; Chen T et al., 2020), etc.

Recently, SSL has also been used in recommender systems to generate accurate recommendations. For instance, Xie et al. (2022) designed contrastive learning for sequential recommendation (CL4SRec), in which three data enhancement methods were proposed to extract self-supervised signals from the user's behavior to generate the user representation. Xin et al. (2020) used reinforcement learning to extend existing sequential recommendation models, aiming to distinguish different types of user–item interactions. Moreover, Yao et al. (2021) solved the issue of data sparsity in large-scale item recommendations by applying a new data augmentation method and contrastive learning to enhance item representations. Zhou et al. (2020) introduced S^3-Rec (self-supervised learning for sequential recommendation), which learns the correlations among sequences, subsequences, items, and attributes based on mutual information maximization (MIM) to obtain self-supervised signals and to strengthen the item representations through pre-training for recommendation. Furthermore, Xia et al. (2021b) designed a novel dual channel hypergraph convolutional network (DHCN) to obtain the complex high-order information between items, in which SSL was used to maximize the mutual information between the session representations generated through two channels. In addition, Xia et al. (2021a) built graph encoders to simultaneously leverage the internal and external connectivity information of sessions, which was then used to tackle the problem of data sparsity by contrastive learning.

Although the aforementioned self-supervised approaches have made significant improvements in recommender systems, they consider merely the

supervised signals from a session–item perspective or a session–session perspective, while generally ignoring the connections between items in different sessions, which can provide additional supervisory information. In this study, we use the global-level connections between items in the global graph to enhance the item representation learning on the basis of the main supervised learning.

3 Approach

In this section, we first formalize the task of session-based recommendation. Subsequently, we introduce the proposed SGL-TM in detail, which consists mainly of three components (user preference generation, self-supervised learning, and model optimization).

Session-based recommendation aims to recommend the next item with which a user will interact based on the limited interactions in an ongoing session. Denote $V = \{v_1, v_2, \ldots, v_{|V|}\}$ the item set, which contains all items ($|V|$ represents the number of items in V). Assume that $U = \{S_1, S_2, \ldots, S_{|U|}\}$ indicates all sessions, where $|U|$ represents the number of sessions. Denote $S_i = \{v_1, v_2, \ldots, v_n\}$ the ith session in U, which includes n items in chronological order, and v_t ($t = 1, 2, \ldots, n$) represents the item with which interaction occurs at timestamp t in S_i. Given a current session S_i, we first generate the user preference, which is applied to obtain the prediction scores of all items. Eventually, items ranked in the top K positions will form the recommended list for the user.

Fig. 1 presents the framework of our proposed SGL-TM model for session-based recommendation. Given a current session, we first learn the item embeddings by the information propagation of GAT. Then we obtain the user's long-term and current interests in the session, and concatenate them to generate the final session representation. Specifically, we establish a global graph to reveal the transition relations between items among different sessions according to all sessions in the training set. Then, on one hand, we obtain the self-supervised signals from the global-level connections between items to calculate the self-supervised loss. On the other hand, we calculate the prediction scores of all items in the candidate set, which are then adjusted by the designed target-adaptive masking module to obtain

the cross-entropy loss as the main supervised loss. Finally, the main supervised loss and self-supervised loss are combined to generate the final loss for model optimization.

3.1 User preference generation

3.1.1 Item representation learning

For a session S_i, we first establish a directed local graph to reflect the transition relationships between items in S_i. Specifically, S_i can be denoted as $\tilde{\mathcal{G}}_i = \{\tilde{\mathcal{V}}_i, \tilde{\mathcal{E}}_i\}$, where $\tilde{\mathcal{V}}_i$ and $\tilde{\mathcal{E}}_i$ denote the node set and edge set in the local graph $\tilde{\mathcal{G}}_i$, respectively. Furthermore, $\tilde{\mathcal{V}}_i = \{x_1, x_2, \ldots, x_m\}$ comprises all the unique items in S_i, where $m \leq n$ because items may be repeatedly interacted in the session. Each edge $e_{ij} \in \tilde{\mathcal{E}}_i$ indicates that the user interacts with x_i before x_j in the session.

After constructing the local graph $\tilde{\mathcal{G}}_i$, we conduct information propagation on $\tilde{\mathcal{G}}_i$ to learn the item representations. Here we adopt the GAT (Veličković et al., 2018), considering that compared to gated graph neural networks (GG-NNs) (Li YJ et al., 2016) and graph convolutional networks (GCNs) (Kipf and Welling, 2017), GAT can dynamically determine the different importance of its neighbors to each node in the local graph. Specifically, we first initialize the embedding vector $\boldsymbol{x}_i$ of each node x_i in $\tilde{\mathcal{G}}_i$ as follows:

$$\boldsymbol{x}_i^0 = \text{Embedding}(x_i), \tag{1}$$

in which "Embedding" indicates the embedding layer, $\boldsymbol{x}_i^0 \in \mathbb{R}^d$ denotes the initialized item embeddings of x_i, and d represents the dimension of embeddings.

To obtain the pairwise transition relationships between items in the ongoing session, we adopt GAT to propagate neighbor information for each node. For each node x_i at layer l, the importance degree of different neighbor nodes is determined by employing a self-attention mechanism, where the importance between nodes x_i and x_j is denoted by an attention coefficient calculated as follows:

$$e_{ij} = \sigma_1 \left(\boldsymbol{W}_0^{\text{T}}[\boldsymbol{W}_1 \boldsymbol{x}_i^{l-1}; \boldsymbol{W}_2 \boldsymbol{x}_j^{l-1}] \right), \tag{2}$$

where $\boldsymbol{W}_0 \in \mathbb{R}^{2d}$ and $\boldsymbol{W}_1, \boldsymbol{W}_2 \in \mathbb{R}^{d \times d}$ represent the trainable parameters, $[\,;\,]$ denotes the concatenation operation, and σ_1 indicates the Leaky ReLU function.
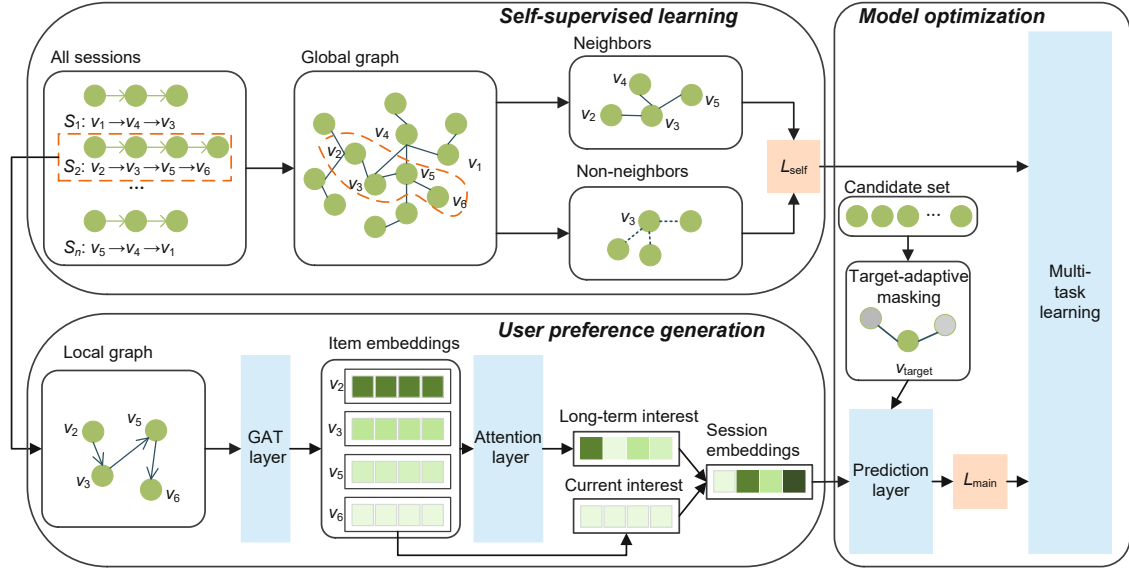
Fig. 1 Framework of self-supervised graph learning with target-adaptive masking (SGL-TM) (GAT: graph attention network)

Next, the attention coefficients are normalized via using a softmax layer:

$$\alpha_{ij} = \text{Softmax}\,(e_{ij}) = \frac{\exp\,(e_{ij})}{\sum_{k \in \mathcal{N}_{v_i}} \exp\,(e_{ik})}, \qquad (3)$$

in which $\mathcal{N}_{v_i}$ indicates the neighbor set of v_i in the local graph $\tilde{\mathcal{G}}_i$.

Finally, a linear combination of the corresponding neighbors is used to update the node vector of v_i using the attention coefficients:

$$\boldsymbol{x}_i^l = \sigma \left(\sum_{j \in \mathcal{N}(v_i)} \alpha_{ij} \boldsymbol{W}_3 \boldsymbol{x}_j^{l-1} \right), \qquad (4)$$

where σ and $\boldsymbol{W}_3 \in \mathbb{R}^{d \times d}$ denote the sigmoid function and learnable parameters, respectively.

After the information propagation on multilayer GATs, we obtain the item representations $\boldsymbol{x}_i$ in the local graph $\tilde{\mathcal{G}}_i$. Finally, the item sequence is recovered from the local graph to gain the representations of the chronological items within the current session as $\{\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_n\}$.

3.1.2 Session representation generation

After learning the item representations in the current session, following previous works (Xu et al., 2019; Qiu et al., 2020), we take the user's long-term and current interests into account to generate the final session representation as the user preference.

Due to that the last interacted item in the session can reflect the current interest of the user (Qiu et al., 2019; Pan et al., 2020), we directly use the embeddings of the last item as the current preference $\boldsymbol{u}_\text{c} \in \mathbb{R}^d$, i.e., $\boldsymbol{u}_\text{c} = \boldsymbol{z}_n$. In addition, the long-term preference of the user is obtained by aggregating the item representations using an attention mechanism:

$$\begin{cases} \boldsymbol{u}_\text{l} = \sum_{i=1}^n \gamma_i \boldsymbol{z}_i, \\ \gamma_i = \text{Softmax}(\beta_i), \\ \beta_i = \boldsymbol{W}_4 \sigma(\boldsymbol{W}_5 \boldsymbol{z}_i + \boldsymbol{W}_6 \boldsymbol{u}_\text{c}), \end{cases} \qquad (5)$$

where $\boldsymbol{u}_\text{l}$ indicates the long-term preference of the user, and $\boldsymbol{W}_4 \in \mathbb{R}^d$ and $\boldsymbol{W}_5, \boldsymbol{W}_6 \in \mathbb{R}^{d \times d}$ denote the trainable parameters.

Eventually, we attain the session representation by combining the long-term and current interests of the user as follows:

$$\boldsymbol{u} = \boldsymbol{W}_7[\boldsymbol{u}_\text{l}; \boldsymbol{u}_\text{c}], \qquad (6)$$

where $\boldsymbol{W}_7 \in \mathbb{R}^{d \times d}$ denotes the trainable parameters.

3.2 Self-supervised learning

Existing session-based recommendation methods generally ignore the over-smoothing issue of

GNNs, resulting in the indistinguishability of the learned item representations. Thus, in this subsection, we employ self-supervised learning to mine the informative signals about item transitions among different sessions from an item–item perspective. This enhances item representation learning to prevent the over-smoothing issue, which includes global graph construction and self-supervision.

3.2.1 Global graph construction

For the sake of obtaining the pairwise transition relationship between items across different sessions, we construct a global session graph, where the self-supervised signals generated from the global connections are taken into consideration. Assume that $\hat{\mathcal{G}} = \{\hat{\mathcal{V}}, \hat{\mathcal{E}}\}$ represents the global graph involving the global connections, in which $\hat{\mathcal{V}}$ reflects the unique items in all sessions, namely V, and $\hat{\mathcal{E}}$ indicates all the item transitions in the training set. Each edge $e_{ij} \in \hat{\mathcal{E}}$ denotes that items v_i and v_j are adjacently interacted within a session.

Moreover, for popular items, there may exist a large number of neighbor items. Here, to avoid the bias caused by accidental clicks, we employ max sampling to choose M most associated items as the ultimate neighbors for each node in the global graph, where the relevance is determined by the edge weights, which indicates the level of similarity between two nodes. By max sampling, loose connections between nodes are filtered out to make sure that the final neighbors are highly related to the target node.

3.2.2 Self-supervision

After constructing the global graph, we can obtain the self-supervised signals through the connections between items among different sessions. Specifically, given an item v_i in a session, the set of its neighbors in the global graph is denoted as $\mathcal{N}_{v_i}$. For v_i, the embeddings learned by GNNs should be more similar to its neighbor node v_j ($v_j \in \mathcal{N}_{v_i}$) than a non-neighbor node $\bar{v}_j$ ($\bar{v}_j \in V \backslash \mathcal{N}_{v_i}$), where "\" represents the set subtraction operation. Based on the intuition, we employ the Jensen–Shannon divergence (JSD) (Hjelm et al., 2019) as the contrastive loss, which aims to minimize the distance between v_i and its neighbors $\mathcal{N}_{v_i}$, and maximize the distance between v_i and its unconnected items $V \backslash \mathcal{N}_{v_i}$. The

self-supervised loss can be calculated as

$$L_{\text{self}} = \frac{1}{T} \sum_{j=1}^{T} \big(-\log \sigma(f(\boldsymbol{z}_i, \boldsymbol{v}_j)) \\ -\log(1 - \sigma(f(\boldsymbol{z}_i, \bar{\boldsymbol{v}}_j))) \big), \tag{7}$$

where $\boldsymbol{z}_i$ is the item representation of v_i output by GNNs, and $\boldsymbol{v}_j$ and $\bar{\boldsymbol{v}}_j$ are the item vectors of nodes v_j ($v_j \in \mathcal{N}_{v_i}$) and $\bar{v}_j$ ($\bar{v}_j \in V \backslash \mathcal{N}_{v_i}$) generated by the embedding layer respectively. Moreover, T indicates the number of neighbors or non-neighbors of v_i, which is sampled from the neighbors $\mathcal{N}_{v_i}$ in the global graph using max sampling, or from the unconnected items $V \backslash \mathcal{N}_{v_i}$ by random sampling, respectively. In addition, f is a linear mapping function calculated as

$$f(\boldsymbol{z}_i, \boldsymbol{v}_j) = \boldsymbol{z}_i^{\mathrm{T}} \boldsymbol{v}_j; \tag{8}$$

$f : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is the similarity calculation function, which uses the embeddings of two items as inputs and subsequently scores the similarity between them.

This means that each item in the current session can obtain information from its neighbor items in the global graph by contrastive learning, which can be used as additional information to enhance item representation learning to solve the over-smoothing issue caused by using GAT in the current session.

3.3 Model optimization

3.3.1 Target-adaptive masking

After obtaining the session representation $\boldsymbol{u}$, the final recommendation probability of each candidate item can be generated by multiplying $\boldsymbol{u}$ with the embeddings of the candidates as follows:

$$\hat{\boldsymbol{y}}_i = \text{Softmax}(\boldsymbol{u}^{\mathrm{T}} \boldsymbol{v}_i), \quad \forall i = 1, 2, \ldots, |V|, \tag{9}$$

in which $\hat{\boldsymbol{y}}_i$ represents the probability of item v_i being the next item recommended to the user, and $\boldsymbol{v}_i$ indicates the initial embeddings of candidate item v_i.

In addition, because directly adopting the cross-entropy loss with a softmax layer can cause overfitting (that is, similar to the unconnected items, the scores of the neighbors of the target item are continuously decreased), we design a target-adaptive masking component. Specifically, given a global graph $\mathcal{G}_{\text{g}} = \{\mathcal{V}_{\text{g}}, \mathcal{E}_{\text{g}}\}$, for the target item v_{target}, we first sample its neighbors with N largest edge weights

in the global graph as the final neighbors of v_{target}, denoted as $\mathcal{N}_{v_{\text{target}}}$. Then we mask the sampled N items from the original candidate set V as the updated candidate set:

$$V_{\text{Update}} = V \backslash \mathcal{N}_{v_{\text{target}}}. \tag{10}$$

Eventually, the cross-entropy is applied as the main supervised loss as follows:

$$
\begin{aligned}
L_{\text{main}} = - \sum_{i=1}^{|V_{\text{Update}}|} \Big[&\boldsymbol{y}_i \log(\hat{\boldsymbol{y}}_i) \\
&+ (1 - \boldsymbol{y}_i) \log(1 - \hat{\boldsymbol{y}}_i) \Big],
\end{aligned} \tag{11}
$$

in which $\boldsymbol{y}_i$ denotes the one-hot vector of the ground truth.

In this way, the scores of the target item's neighbors will not be decreased in the training procedure, thus alleviating the over-fitting issue to a certain extent.

3.3.2 Multi-task learning

After generating the main supervised loss in Eq. (11) and the auxiliary self-supervised loss in Eq. (7), we obtain the total loss by combining the main and self-supervisions losses as follows:

$$L = L_{\text{main}} + \lambda L_{\text{self}}, \tag{12}$$

in which λ is the trade-off parameter controlling the magnitude of self-supervised learning. Finally, we update the item embeddings and learn the trainable parameters using the back propagation algorithm (Rumelhart et al., 1986).

We detail the main process of SGL-TM training in Algorithm 1. For all sessions in the training set, we establish a global graph to show the connections of items among different sessions (line 1). Then, in the current session, item representations are generated (lines 4 and 5), that is, local graph construction and information propagation by GAT. We generate the current interest of the ongoing session (line 6) and long-term interest (line 7), and combine them to obtain the session representation (line 8). We perform max sampling on the neighbors and random sampling on the non-neighbors of v_i (line 9), and calculate the similarity according to their item embeddings to generate the self-supervised loss (line 10). The prediction score is calculated (line 11) and the

---

Algorithm 1 Training procedure of SGL-TM

Input: set $U = \{S_1, S_2, ..., S_{|U|}\}$;
    λ: the trade-off parameter
Output: $\boldsymbol{I}$: the item representations in V;
    ψ: the trainable parameters in SGL-TM
1: $\hat{\mathcal{G}} = \{\hat{\mathcal{V}}, \hat{\mathcal{E}}\} \leftarrow$ global graph construct (U)
2: **for** each training iteration **do**
3:   **for** S_i in U **do**
4:     $\tilde{\mathcal{G}}_i = \{\tilde{\mathcal{V}}_i, \tilde{\mathcal{E}}_i\} \leftarrow$ local graph construct (S_i)
5:     $\boldsymbol{z}_i =$ information propagation (v_i) based on Eqs. (1)–(4)
6:     $\boldsymbol{u}_{\text{c}} =$ current interest $(\boldsymbol{z}_n)$
7:     $\boldsymbol{u}_{\text{l}} =$ long-term interest $(\boldsymbol{z}_i)$ based on Eq. (5)
8:     $\boldsymbol{u} =$ session representations $(\boldsymbol{u}_{\text{c}}, \boldsymbol{u}_{\text{l}})$ based on Eq. (6)
9:     $\mathcal{N}_{v_i}, V \backslash \mathcal{N}_{v_i} =$ sample $(\hat{\mathcal{G}})$
10:    $L_{\text{self}} = \text{JSD}(\boldsymbol{z}_i, \boldsymbol{v}_j, \bar{\boldsymbol{v}}_j)$ based on Eqs. (7) and (8)
11:    $\hat{\boldsymbol{y}}_i =$ prediction $(\boldsymbol{u}, \boldsymbol{v}_i)$ based on Eq. (9)
12:    $V_{\text{Update}} =$ global graph-aware masking $(V, \mathcal{N}_{v_{\text{target}}})$ based on Eq. (10)
13:    $L_{\text{main}} =$ cross-entropy loss $(\hat{\boldsymbol{y}}_i)$ based on Eq. (11)
14:    Model optimization loss: $L = L_{\text{main}} + \lambda L_{\text{self}}$
15:   **end for**
16:   Use back propagation for the optimization of trainable parameters
17: **end for**
18: **return** $\boldsymbol{I}$ and ψ

---

target-adaptive masking on the candidate set is performed (line 12). We calculate the main supervised loss by a cross-entropy loss (line 13). Finally, we obtain the total loss and employ back propagation to optimize the model parameters.

4 Experiments

4.1 Research questions

We guide our experiments by addressing the following five research questions:

1. Can our SGL-TM model be superior to state-of-the-art baselines on real-world datasets?

2. How significant is the contribution of self-supervised learning and target-adaptive masking modules to the performance of SGL-TM?

3. How does SGL-TM perform with various magnitudes of self-supervised signals incorporated?

4. How does SGL-TM perform at different session lengths?

5. What are the efforts of the important

hyper-parameters, i.e., the number of neighbor and non-neighbor pairs and the number of neighbors masked, on the performance of SGL-TM?

4.2 Datasets

We employ two real-world datasets that are extensively employed in session-based recommendations, i.e., Gowalla (https://snap.stanford.edu/data/loc-gowalla.html) and Diginetica (http://cikm2016.cs.iupui.edu/cikm-cup), to conduct the experiments. Gowalla is a check-in dataset that contains the location information shared by users when they checked in between February 2009 and October 2010 (Davidson et al., 2010). Following Tang and Wang (2018), we retain 30 000 of the most popular locations for the experiments. Diginetica collects users' behavior information from an e-commerce website, which is released by CIKM Cup 2016 (Cheng et al., 2017). Here we use the transaction data, which are appropriate for session-based recommendation. Moreover, following Pan et al. (2022), items that occur less than 5 times and sessions with length 1 are removed from both datasets. Detailed statistics of the two datasets after prepossessing are presented in Table 1.

4.3 Baselines

We compare the performance of our model against the following representative baselines:

FPMC (Rendle et al., 2010) is a compound method based on personalized Markov chains, which is applied in capturing the sequential patterns between adjacent items.

NARM (Li J et al., 2017) employs an RNN and an attention mechanism to model the sequential signals in the session and captures the main intent of the user.

NextItNet (Yuan et al., 2019) adopts a CNN to model the session sequence by learning high-level representations from item dependencies.

FGNN (Qiu et al., 2019) adopts a Readout function and a weighted attention graph layer to learn the representations of sessions and items.

SR-GNN (Wu et al., 2019) converts each session into a graph and generates item representations using GG-NNs.

GCE-GNN (Wang ZY et al., 2020) employs both global- and local-level pairwise transitions between items to obtain the user preference.

S^2-DHCN (Xia et al., 2021b) introduces two hypergraphs to extract the beyond-pairwise relations of items and adopts self-supervised learning to enhance item representation.

4.4 Evaluation metrics

We employ Recall@K and MRR@K to assess the performance of recommendation models, which are always used in previous works (Qiu et al., 2019; Choi et al., 2021).

Recall@K: It is an indicator for measuring whether the next-click item appears in the top K positions of the recommended list, which is formulated as follows:

$$\text{Recall@}K = \frac{n_{\text{hit}}}{N_1}, \qquad (13)$$

in which N_1 denotes the number of all test sequences and n_{hit} indicates the number of next-click items in the top K positions of the recommended list.

MRR@K: This is a metric that takes the rank of a target item in the recommended list into consideration. It is set to 0 when the correct recommended item is not returned in the top K positions of the recommended list; otherwise, it is generated as follows:

$$\text{MRR@}K = \frac{1}{N_2} \sum_{v_{\text{correct}} \in S_{\text{test}}} \frac{1}{\text{Rank}\,(v_{\text{correct}})}, \quad (14)$$

in which N_2 and v_{correct} indicate the number of all test sessions ($|S_{\text{test}}|$) and the target item of a session, respectively.

4.5 Experimental setup

We divide Gowalla and Diginetica into the training and test sets for training and evaluation,

Table 1 Dataset statistics

Dataset	Number of clicks	Number of sessions	Number of items	Average session length	Average number of clicks per item
Gowalla	1 122 788	830 893	29 510	3.85	38.05
Diginetica	981 620	777 029	42 956	4.80	22.85

respectively. For the Gowalla dataset, we employ the sessions that occurred in the last week for testing. For the Diginetica dataset, the last 20% of the sessions are separated as the test set.

Moreover, following Chen TW and Wong (2020), we use the data augmentation strategy for both datasets. Adam (Kingma and Ba, 2015) is adopted as our optimizer to train the model. We set the dimension of the item embeddings and the mini-batch size to 128 and 512, respectively. Moreover, the learning rate is initialized at 0.001 and attenuated by 0.5 every three epochs. In addition, the recommendation number K is set to 20 for evaluation. Furthermore, we use grid search to find the best performance of SGL-TM on both datasets by ranging the dropout and layer numbers in $\{0, 0.1, 0.25\}$ and $\{1, 2, 3\}$, respectively. Similarly, we employ grid search to determine the number of neighbor and nonneighbor pairs in the self-supervised learning module (i.e., M) and the number of neighbors masked in the target-adaptive masking component (i.e., N) by ranging M and N both in $\{1, 2, 3, 4\}$.

5 Results and discussion

5.1 Overall performance

We compare SGL-TM with state-of-the-art recommendation baselines in terms of Recall@20 and MRR@20 on both datasets. The experimental results are displayed in Table 2, from which we can draw the following observations:

First, compared with the neural models, the traditional recommendation methods like FPMC, based on Markov chains, have completely lost the superior-

Table 2 Model performance

Method	Recall@20 (%)		MRR@20 (%)	
	Gowalla	Diginetica	Gowalla	Diginetica
FPMC	29.91	28.50	11.45	7.67
NextItNet	45.15	45.41	21.26	15.19
NARM	50.07	49.80	23.92	16.57
FGNN	50.06	50.03	24.12	17.01
SR-GNN	50.32	50.81	<u>24.25</u>	17.31
GCE-GNN	51.51	51.66	23.52	17.53
S^2-DHCN	<u>51.96</u>	<u>52.58</u>	23.47	<u>17.91</u>
SGL-TM	**52.26**$^\triangle$	**53.02**$^\triangle$	**25.29**$^\triangle$	**18.38**$^\triangle$

The results of the best performing model and the best baseline in each column are boldfaced and underlined, respectively. $^\triangle$ indicates a statistical significance of SGL-TM against the best baseline applying a paired t-test ($p < 0.01$)

ity. This is due to the fact that MC-based methods pay more attention to the pairwise sequential signals between adjacent items, and ignore other sequential information in the whole session. Moreover, it is obvious that the performance of NARM on the two datasets is superior to that of NextItNet, because the session sequence in the session-based recommendation is generally generated within a short time and thus is likely to be time-dependent. Compared with the CNN used in NextItNet, RNN applied in NARM is better at modeling such time-dependent sequences. In addition, among the baselines, the GNN-based methods achieve the best performance, which suggests that GNNs can precisely model the complicated transition patterns between items in the session. Moreover, GCE-GNN and S^2-DHCN outperform SR-GNN in most cases on both datasets, which proves that capturing both local and global levels of item information can help users make accurate predictions. However, GCE-GNN and S^2-DHCN lose the competition to SR-GNN in terms of MRR@20 on the Gowalla dataset, which may be a result of the interference of irrelevant items in other sessions making them unable to accurately identify the user's intention.

Generally, our method SGL-TM is superior to the competitive baselines in terms of both Recall@20 and MRR@20 on both datasets, revealing its effectiveness for the session-based recommendation task. The reasons can be summarized as follows: The self-supervised learning module can employ abundant information from other sessions, which can help the model generate accurate item representations by avoiding the over-smoothing issue caused by GNNs. In addition, the target-adaptive masking module can effectively avoid the phenomenon that the scores of items adjacent to the target item are decreasing in a manner comparable to those of other unrelated items in the training process. Moreover, SGL-TM gains the improvements over the best baselines, namely, SR-GNN and S^2-DHCN, by 0.58% in terms of Recall@20 and 4.29% in terms of MRR@20 on Gowalla, respectively; while the corresponding improvements are 0.84% and 2.62% on the Diginetica dataset. It can be found that the improvement of our SGL-TM on MRR@20 is more obvious than that on Recall@20 on both datasets, which suggests that compared to hitting the target item in the recommended list, our SGL-TM is more efficient at ranking it in the correct

location.

5.2 Ablation study

In this subsection, we develop three variants of SGL-TM (i.e., Base, Base-TM, and Base-SSL) to explore the contribution of each component in our proposal by comparing the variants to SGL-TM on the Gowalla and Diginetica datasets. Specifically, the variant Base merely adopts GAT to model the session sequence without self-supervised learning or the target-adaptive masking module. Base-TM and Base-SSL remove the self-supervised learning module and the target-adaptive masking module from SGL-TM, respectively. The results of the variants and SGL-TM are presented in Table 3.

According to Table 3, it can be found that both self-supervised learning and target-adaptive masking modules contribute to the improvement of model performance. Moreover, removing the self-supervised learning module results in a larger degradation of the model performance than removing the target-adaptive masking. This may be because, compared to the over-fitting problem, the over-smoothing issue of the model is more prominent and has a greater effect on the recommendation accuracy. In addition, comparing Base-TM and SGL-TM, we find that after getting rid of the self-supervised learning module, the model performance obviously drops, which shows that the global-level connections between items reflected in different sessions can provide additional self-supervisions in addition to the supervised signals in the session–item perspective for item representation learning. Moreover, compared with SGL-TM, the model performance of Base-TM is reduced by 1.79% and 3.75% in terms of Recall@20 and MRR@20 on Diginetica, while the performance dropping rates are 1.24% and 3.91% on the Gowalla dataset. It is evident that the contribution of the self-supervised learning module on ranking the target items at the correct location is larger than that on hitting them in the recommended list.

Table 3 Ablation study

Method	Recall@20 (%)		MRR@20 (%)	
	Gowalla	Diginetica	Gowalla	Diginetica
Base	50.68	51.14	24.17	17.32
Base-TM	51.61	52.07	24.30	17.69
Base-SSL	52.05	52.65	24.87	18.27
SGL-TM	52.26	53.02	25.29	18.38

5.3 Impact of the magnitude of self-supervision

We introduce a trade-off parameter λ in Eq. (12) in our proposal to control the magnitude of incorporating the self-supervised signals. Specifically, we test the performance of SGL-TM by tuning the parameter λ in $\{0, 0.0001, 0.001, 0.01, 0.1, 0.2, 0.5, 1.0, 2.0, 3.0\}$ to investigate the impact of the self-supervised learning module. The results are presented in Fig. 2.

Based on Fig. 2, the introduction of self-supervised signals with an appropriate λ effectively improves the performance of recommender systems. This is due to the fact that the self-supervised learning module alleviates the over-smoothing problem and introduces the global-level information from other sessions to generate accurate item representations. In detail, for the Diginetica dataset, as λ goes up, the performance of SGL-TM in terms of both metrics, first exhibits an increasing trend and then decreasing, where the best performance is achieved when λ is 0.5. This is because when λ is small, the self-supervised signals are insufficient to solve
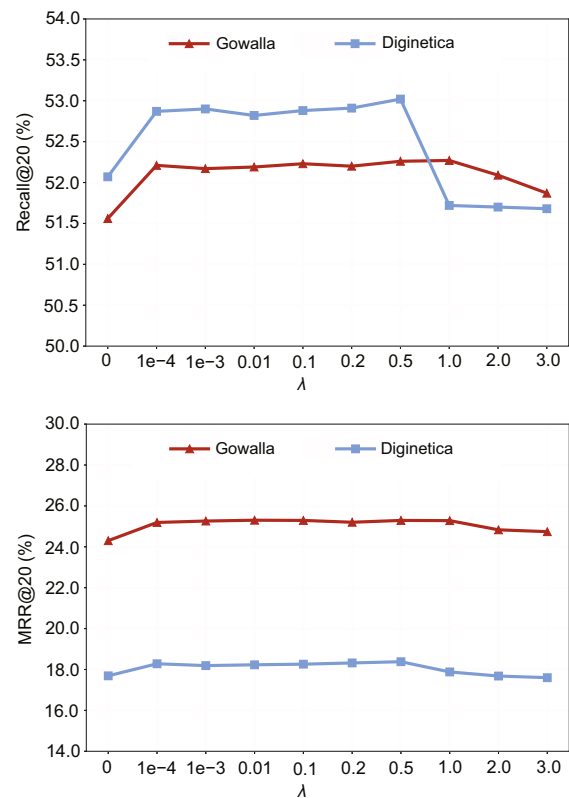

Fig. 2 Impact of the magnitude of self-supervision

the over-smoothing issue, while a large λ may cause the over-fitting problem and thus lead to relatively low model performance. For the Gowalla dataset, when λ is within 1, as λ increases, the performance of SGL-TM first increases and then gently fluctuates in terms of both Recall@20 and MRR@20. When λ is large enough on the Gowalla dataset (e.g., 2 or 3), the performance of SGL-TM begins to decline. When λ is within 1, the different trends of the model performance on Diginetica and Gowalla may be made clear by the fact that the average number of clicks per item on Gowalla is larger than that on Diginetica. This indicates that the degree of items in the Gowalla dataset is correspondingly higher, so the Gowalla dataset is more likely to suffer from a serious over-smoothing problem. Thus, more self-supervised signals are required to tackle the problem on Gowalla (Chen M et al., 2020; Nie et al., 2020, 2021).

5.4 Impact of session length

Considering that sessions have various lengths in real-world situations, it is necessary to assess the performance of SGL-TM as well as the baselines when dealing with sessions with various lengths. Specifically, we separate the sessions in Gowalla and Diginetica into three groups with different lengths: "Short," containing sessions of less than 6 items; "Long," including sessions of more than 10 items; and the others denoted as "Medium." Subsequently, we examine the performance of SGL-TM and the competitive baselines, namely, SR-GNN, GCE-GNN, and S^2-DHCN, on the three groups of sessions on both datasets. The outcomes are exhibited in Fig. 3.

According to Fig. 3, SGL-TM outperforms the baselines in terms of both Recall@20 and MRR@20 at different session lengths, demonstrating the effectiveness of our proposal on dealing with sessions of various lengths. Especially on the Gowalla dataset, the performance of our proposal in terms of both metrics improves when the session length increases.
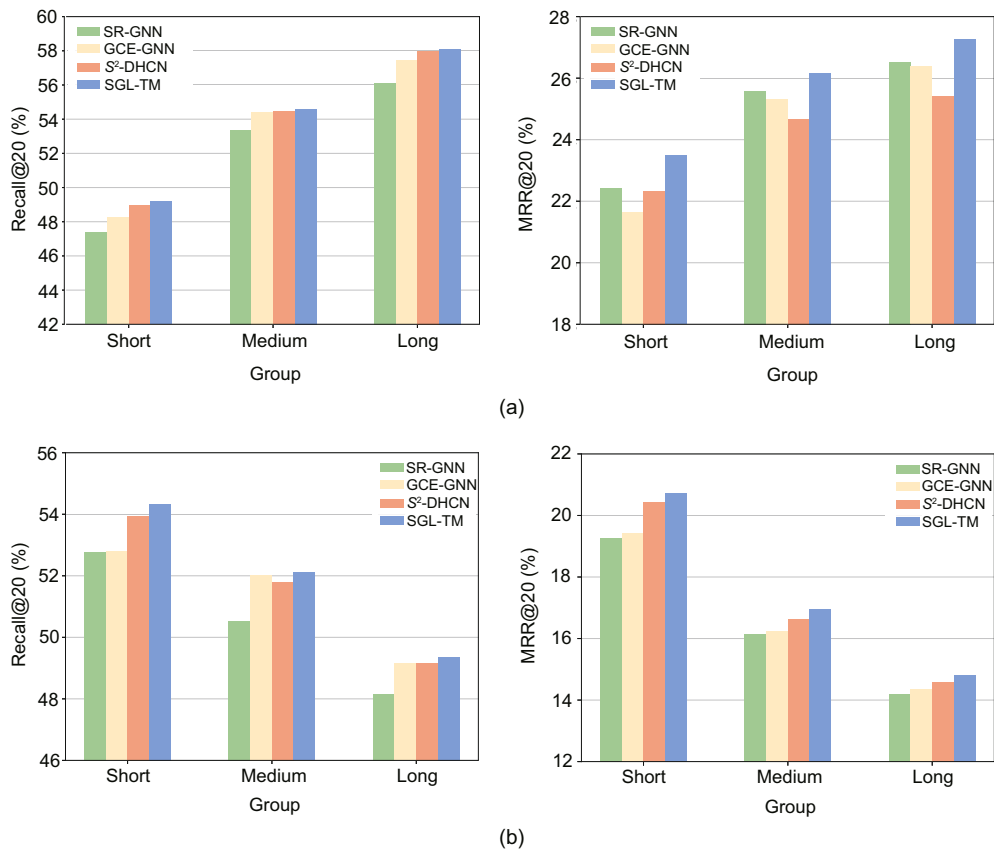
Fig. 3 Impact of the session length on Gowalla (a) and Diginetica (b). References to color refer to the online version of this figure

However, the performance of SGL-TM shows an opposite trend with the session length increasing on Diginetica. We analyze the phenomenon that is caused by the difference between the characteristics of the datasets. Specifically, for Gowalla under the check-in scenario, users generally pay attention to similar places. Thus, with the number of interactive items increasing, more information can be provided to determine the user's purpose. Differently, for the Diginetica dataset, as the number of interactive items goes up, the user preference may become difficult to capture considering that the intent changes rapidly in the e-commerce scenario. In addition, it is observed that SGL-TM gains the most obvious improvements over the competitive baselines in terms of both metrics on the Short sessions among the three groups on both datasets, and the performance gap between SGL-TM and the baselines gradually decreases as the session length increases, which reveals that our proposal can effectively capture the user intent from limited historical interactions.

5.5 Hyper-parameter analysis

We examine the influence of some important hyper-parameters (including M and N, which are introduced in the self-supervised learning and target-adaptive masking components, respectively) on the performance of SGL-TM.

5.5.1 Impact of hyper-parameter M

To illustrate the influence of the number of neighbor and non-neighbor pairs in the self-supervised learning module (i.e., M) on the recommendation accuracy, we search M in $\{1, 2, 3, 4\}$. The results of SGL-TM on Gowalla and Diginetica are presented in Fig. 4.

On the Diginetica dataset, from Fig. 4, we find that the peak performance of SGL-TM is achieved when M is 2. With the number of item pairs increasing, the performance of SGL-TM first increases and then decreases. This could be explained by adding a proper number of item pairs because the self-supervised signals can improve the model performance; however, introducing too many item pairs will result in self-supervision redundancy, which leads the model to over-fitting. Interestingly, on Gowalla, the performance of SGL-TM continually improves as the number of item pairs increases. The phenomenon is consistent with that mentioned in Section 5.3 and can be explained by the fact that the Gowalla dataset needs more self-supervised signals than Diginetica.

5.5.2 Impact of hyper-parameter N

We study the influence of the number of neighbors masked in the target-adaptive masking component, i.e., N. Specifically, we tune N in $\{1, 2, 3, 4\}$, and the results on Gowalla and Diginetica are presented in Fig. 5. We can observe that on both the Gowalla and Diginetica datasets, SGL-TM achieves the peak performance when N is 4. As N increases, the performance of SGL-TM on both Gowalla and Diginetica in terms of both metrics generally increases, which indicates that as the number of masked neighbors of the target item increases, our
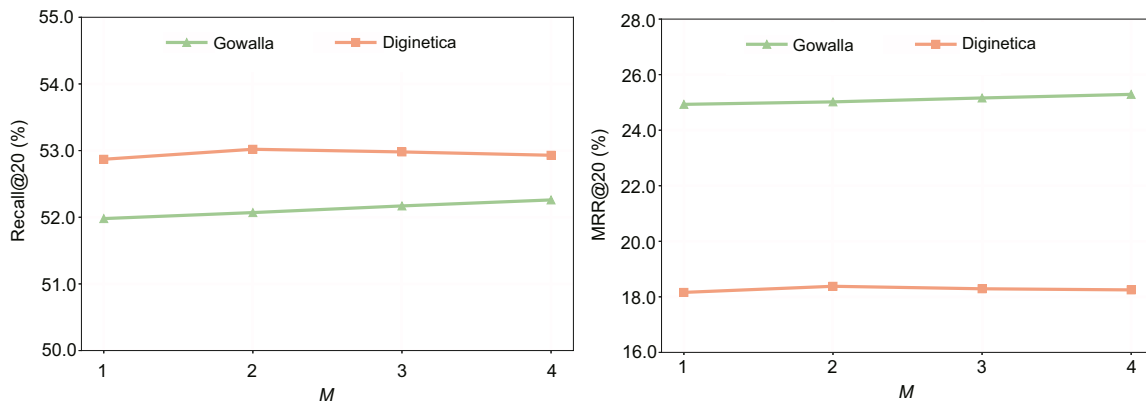

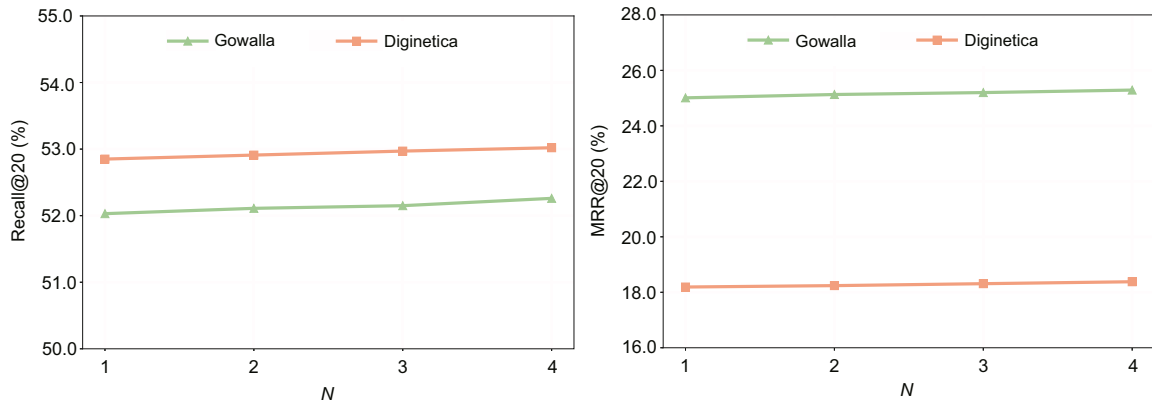**Fig. 4 Impact of M on both datasets**

Fig. 5 Impact of *N* on both datasets

target-adaptive masking module can more effectively tackle the over-fitting problem.

6 Conclusions and future work

In this paper, we have proposed a self-supervised graph learning with target-adaptive masking (SGL-TM) method for session-based recommendation. Specifically, we have employed self-supervised learning to tackle the over-smoothing issue of GNNs by introducing the global-level transition relations between items. Moreover, we have designed a target-adaptive masking module to effectively overcome the over-fitting problem of a cross-entropy loss with a softmax layer. The results of the experiments implemented on two real-world datasets, namely, Gowalla and Diginetica, have verified the effectiveness of our SGL-TM in terms of Recall@20 and MRR@20.

As for future work, we plan to investigate the similarity among sessions to capture more self-supervised signals to generate accurate recommendations. In addition, we would like to tackle the cold-start issue in recommender systems by mining the user–item interaction information in the session graph from multiple perspectives (Zheng et al., 2021).

Contributors

Yitong WANG designed the research and drafted the paper. Chengyu SONG made major revisions to the paper. Fei CAI and Zhiqiang PAN further modified and finalized the paper.

Compliance with ethics guidelines

Yitong WANG, Fei CAI, Zhiqiang PAN, and Chengyu SONG declare that they have no conflict of interest.

Data availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

References

Afouras T, Owens A, Chung JS, et al., 2020. Self-supervised learning of audio-visual objects from video. Proc 16th European Conf, p.208-224.
https://doi.org/10.1007/978-3-030-58523-5_13

Chen M, Wei ZW, Huang ZF, et al., 2020. Simple and deep graph convolutional networks. Proc 37th Int Conf on Machine Learning, p.1725-1735.
https://doi.org/10.5555/3524938.3525099

Chen T, Kornblith S, Norouzi M, et al., 2020. A simple framework for contrastive learning of visual representations. Proc 37th Int Conf on Machine Learning, p.1597-1607. https://doi.org/10.5555/3524938.3525087

Chen TW, Wong RCW, 2020. Handling information loss of graph neural networks for session-based recommendation. Proc 26th ACM SIGKDD Int Conf on Knowledge Discovery & Data Mining, p.1172-1180.
https://doi.org/10.1145/3394486.3403170

Cheng ZY, Shen JL, Zhu L, et al., 2017. Exploiting music play sequence for music recommendation. Proc 26th Int Joint Conf on Artificial Intelligence, p.3654-3660.
https://doi.org/10.24963/ijcai.2017/511

Choi M, Kim J, Lee J, et al., 2021. Session-aware linear item–item models for session-based recommendation. Proc Web Conf, p.2186-2197.
https://doi.org/10.1145/3442381.3450005

Davidson J, Liebald B, Liu JN, et al., 2010. The YouTube video recommendation system. Proc 4th ACM Conf on Recommender Systems, p.293-296.
https://doi.org/10.1145/1864708.1864770

Hidasi B, Karatzoglou A, Baltrunas L, et al., 2016. Session-based recommendations with recurrent neural networks. Proc 4th Int Conf on Learning Representations.

Hjelm RD, Fedorov A, Lavoie-Marchildon S, et al., 2019. Learning deep representations by mutual information estimation and maximization. Proc 7th Int Conf on Learning Representations.

Kingma DP, Ba LJ, 2015. Adam: a method for stochastic optimization. Proc 3rd Int Conf on Learning Representations.

Kipf TN, Welling M, 2017. Semi-supervised classification with graph convolutional networks. Proc 5th Int Conf on Learning Representations.

Kong LP, de Masson d'Autume C, Yu L, et al., 2020. A mutual information maximization perspective of language representation learning. Proc 8th Int Conf on Learning Representations.

Li J, Ren PJ, Chen ZM, et al., 2017. Neural attentive session-based recommendation. Proc ACM Conf on Information and Knowledge Management, p.1419-1428. https://doi.org/10.1145/3132847.3132926

Li YJ, Tarlow D, Brockschmidt M, et al., 2016. Gated graph sequence neural networks. Proc 4th Int Conf on Learning Representations.

Liu Q, Zeng YF, Mokhosi R, et al., 2018. STAMP: short-term attention/memory priority model for session-based recommendation. Proc 24th ACM SIGKDD Int Conf on Knowledge Discovery & Data Mining, p.1831-1839. https://doi.org/10.1145/3219819.3219950

Nie LQ, Li YQ, Feng FL, et al., 2020. Large-scale question tagging via joint question-topic embedding learning. ACM Trans Inform Syst, 38(2):20. https://doi.org/10.1145/3380954

Nie LQ, Jiao FK, Wang WJ, et al., 2021. Conversational image search. IEEE Trans Image Process, 30:7732-7743. https://doi.org/10.1109/TIP.2021.3108724

Pan ZQ, Cai F, Chen WY, et al., 2020. Star graph neural networks for session-based recommendation. Proc 29th ACM Int Conf on Information & Knowledge Management, p.1195-1204. https://doi.org/10.1145/3340531.3412014

Pan ZQ, Cai F, Chen WY, et al., 2022. Graph co-attentive session-based recommendation. ACM Trans Inform Syst, 40(4):67. https://doi.org/10.1145/3486711

Qiu RH, Li JJ, Huang Z, et al., 2019. Rethinking the item order in session-based recommendation with graph neural networks. Proc 28th ACM Int Conf on Information and Knowledge Management, p.579-588. https://doi.org/10.1145/3357384.3358010

Qiu RH, Huang Z, Li JJ, et al., 2020. Exploiting cross-session information for session-based recommendation with graph neural networks. ACM Trans Inform Syst, 38(3):22. https://doi.org/10.1145/3382764

Rendle S, Freudenthaler C, Schmidt-Thieme L, 2010. Factorizing personalized Markov chains for next-basket recommendation. Proc 19th Int Conf on World Wide Web, p.811-820. https://doi.org/10.1145/1772690.1772773

Rumelhart DE, Hinton GE, Williams RJ, 1986. Learning representations by back-propagating errors. Nature, 323(6088):533-536. https://doi.org/10.1038/323533a0

Shani G, Heckerman D, Brafman RI, 2005. An MDP-based recommender system. J Mach Learn Res, 6:1265-1295.

Singhal A, Sinha P, Pant R, 2017. Use of deep learning in modern recommendation system: a summary of recent works. Int J Comput Appl, 180(7):17-22.

Tan YK, Xu XX, Liu Y, 2016. Improved recurrent neural networks for session-based recommendations. Proc 1st Workshop on Deep Learning for Recommender Systems, p.17-22. https://doi.org/10.1145/2988450.2988452

Tang JX, Wang K, 2018. Personalized top-n sequential recommendation via convolutional sequence embedding. Proc 11th ACM Int Conf on Web Search and Data Mining, p.565-573. https://doi.org/10.1145/3159652.3159656

Veličković P, Cucurull G, Casanova A, et al., 2018. Graph attention networks. Proc 6th Int Conf on Learning Representations.

Wang SJ, Cao LB, Wang Y, et al., 2022. A survey on session-based recommender systems. ACM Comput Surv, 54(7):154. https://doi.org/10.1145/3465401

Wang X, He XN, Wang M, et al., 2019. Neural graph collaborative filtering. Proc 42nd Int ACM SIGIR Conf on Research and Development in Information Retrieval, p.165-174. https://doi.org/10.1145/3331184.3331267

Wang XN, Tan QM, 2020. DAN: a deep association neural network approach for personalization recommendation. Front Inform Technol Electron Eng, 21(7):963-980. https://doi.org/10.1631/FITEE.1900236

Wang ZY, Wei W, Cong G, et al., 2020. Global context enhanced graph neural networks for session-based recommendation. Proc 43rd Int ACM SIGIR Conf on Research and Development in Information Retrieval, p.169-178. https://doi.org/10.1145/3397271.3401142

Wu S, Tang YY, Zhu YQ, et al., 2019. Session-based recommendation with graph neural networks. Proc 33rd AAAI Conf on Artificial Intelligence, p.346-353. https://doi.org/10.1609/aaai.v33i01.3301346

Xia X, Yin HZ, Yu JL, et al., 2021a. Self-supervised graph co-training for session-based recommendation. Proc 30th ACM Int Conf on Information & Knowledge Management, p.2180-2190. https://doi.org/10.1145/3459637.3482388

Xia X, Yin HZ, Yu JL, et al., 2021b. Self-supervised hypergraph convolutional networks for session-based recommendation. Proc 35th AAAI Conf on Artificial Intelligence, p.4503-4511.

Xie X, Sun F, Liu ZY, et al., 2022. Contrastive learning for sequential recommendation. Proc IEEE 38th Int Conf on Data Engineering, p.1259-1273. https://doi.org/10.1109/ICDE53745.2022.00099

Xin X, Karatzoglou A, Arapakis I, et al., 2020. Self-supervised reinforcement learning for recommender systems. Proc 43rd Int ACM SIGIR Conf on Research and Development in Information Retrieval, p.931-940. https://doi.org/10.1145/3397271.3401147

Xu CF, Zhao PP, Liu YC, et al., 2019. Graph contextualized self-attention network for session-based recommendation. Proc 28th Int Joint Conf on Artificial Intelligence, p.3940-3946. https://doi.org/10.24963/ijcai.2019/547

Yao TS, Yi XY, Cheng DZ, et al., 2021. Self-supervised learning for large-scale item recommendations. Proc 30th ACM Int Conf on Information & Knowledge Management, p.4321-4330. https://doi.org/10.1145/3459637.3481952

Yuan FJ, Karatzoglou A, Arapakis I, et al., 2019. A simple convolutional generative network for next item recommendation. Proc 12th ACM Int Conf on Web Search and Data Mining, p.582-590.
https://doi.org/10.1145/3289600.3290975

Zhang JQ, Zhao Y, Saleh M, et al., 2020. PEGASUS: pre-training with extracted gap-sentences for abstractive summarization. Proc 37th Int Conf on Machine Learning, p.11328-11339.
https://doi.org/10.5555/3524938.3525989

Zheng JW, Ma QL, Gu H, et al., 2021. Multi-view denoising graph auto-encoders on heterogeneous information networks for cold-start recommendation. Proc 27th ACM SIGKDD Conf on Knowledge Discovery & Data Mining, p.2338-2348.
https://doi.org/10.1145/3447548.3467427

Zhou K, Wang H, Zhao WX, et al., 2020. S^3-Rec: self-supervised learning for sequential recommendation with mutual information maximization. Proc 29th ACM Int Conf on Information & Knowledge Management, p.1893-1902. https://doi.org/10.1145/3340531.3411954