TEA: A Sequential Recommendation Framework via Temporally Evolving Aggregations

Zijian Li, Ruichu Cai, Senior Member, IEEE, Fengzhu Wu, Sili Zhang,
Hao Gu, Yuexing Hao, and Yuguang Yan

Abstract—Sequential recommendation aims to choose the most suitable items for a user at a specific timestamp given historical behaviors. Existing methods usually model the user behavior sequence based on transition-based methods such as Markov chain. However, these methods also implicitly assume that the users are independent of each other without considering the influence between users. In fact, this influence plays an important role in sequence recommendation since the behavior of a user is easily affected by others. Therefore, it is desirable to aggregate both user behaviors and the influence between users, which are evolved temporally and involved in the heterogeneous graph of users and items. In this article, we incorporate dynamic user–item heterogeneous graphs to propose a novel sequential recommendation framework. As a result, the historical behaviors as well as the influence between users can be taken into consideration. To achieve this, we first formalize sequential recommendation as a problem to estimate conditional probability given temporal dynamic heterogeneous graphs and user behavior sequences. After that, we exploit the conditional random field to aggregate the heterogeneous graphs and user behaviors for probability estimation and employ the pseudo-likelihood approach to derive a tractable objective function. Finally, we provide scalable and flexible implementations of the proposed framework. Experimental results on three real-world datasets not only demonstrate the effectiveness of our proposed method but also provide some insightful discoveries on the sequential recommendation.

Index Terms—Conditional random field (CRF), dynamic heterogeneous graph, recommendation system, sequential recommendation.

I. INTRODUCTION

The sequential recommendation system is achieving more and more attention because of its practicality and effectiveness [1]–[4]. In a sequential recommendation system, the users access different items at different time stamps and frequently interact with each other. The difficulties of sequential recommendation mainly come from two aspects: the temporal dependency of historical behaviors and the nonstationarity of users. The temporal dependency of historical behaviors means that the decision of a user is influenced by the historical behaviors. Also, the nonstationarity of users means that the decision of a user is influenced by the social relationship with the neighbors (i.e., adjacent nodes of a user in the social networks) and the user–item interactions of the neighbors. Therefore, one important challenge is how to effectively leverage the historical behaviors and the social relationship between users.

NOMENCLATURE

\[ U, V \] User and item set.
\[ m, n \] Size of user set and item set.
\[ G_t^b, G_t^b \] Bipartite graph only includes the user–item interaction and that at the rth timestamp.
\[ E_t^b, E_t^b \] Edges set of bipartite graph and that at the rth time step.
\[ G_t \] Social networks.
\[ \mathcal{E}_t \] Edges among users in social network.
\[ \mathcal{H}_t \] Heterogeneous graph that includes the social network \( G_t \) and the bipartite graph \( G_t^b \) at the rth time step.
\[ p_t \] Embedding of user \( u_t \).
\[ q_t \] Embedding of item \( v_j \).
\[ k_t \] Embedding of the \( j \)th position in item sequences.
\[ W, b \] Weights and biases in neural networks.
\[ d \] Dimension number of representation.
\[ \Theta_f \] Parameters of unary scores function.
\[ \Theta_t \] Parameters of transition scores function.
\[ \oplus \] Concatenation operator of any two vectors.
\[ \mathcal{X} \] Observed item sequence.
\[ \mathcal{Y} \] Label item sequence of \( \mathcal{X} \).
\[ \mathcal{N}(u_t) \] First-order neighborhood of user \( u_t \).
\[ \mathcal{I}_t(u_t) \] Accessed items of user \( u_t \) at the rth time step.
\[ \tau \] Time window for selecting the walks in the duration of [\( t - \tau, t + \tau \)].
\[ d \] Dimension of user and item embedding.
Focusing on the above challenge, numerous sequential recommendation algorithms have been proposed in recent years, which leverages the user behavior sequence and employs the Markov chains (MCs) to model the transition of items. Eskandanian and Mobasher [1] mined the user preference and identified change points in the sequence of user interactions by using the hidden Markov model [as shown in Fig. 1(a)]. He et al. [2] modeled the personalized sequential behavior by using the personalized translation vectors and the previous item embedding to predict the next item. These transition-based methods assume that the users are independent of each other, which ignores the influence between the users. Considering that the behavior of a user is easily affected by the neighbors, ignoring the dependence between users will suffer from limited performance in the sequential recommendation.

Another kind of recommendation algorithm [4]–[7] focuses on analyzing the social relationships between users and user–item interactions in a static user–item graph. The typical methods include the traditional collaborative filtering (CF) methods [8]–[10], the deep learning enhanced approaches [11]–[14], the recently developed graph neural network (GNN)-based methods [15], as well as the social-network-based methods [5], [16], [17]. These methods reveal that both the interactions among users in social networks and the user–item bipartite graphs are beneficial to the performance of the recommendation system. However, almost all the aforementioned methods assume that the heterogeneous graphs of the users and the items are static, which ignores the dynamical influence of the temporal interaction between items and social networks and further results in the suboptimal performance of recommendation systems. Take Fig. 1(b) for a toy example. The existing methods without considering the dynamic user–item heterogeneous graph might recommend \( v_1 \) in preference to \( v_2 \) since more friends of user \( U_1 \) choose \( v_1 \).

Thus, it is essential to devise a unified framework to take advantage of both the historical behaviors of a user and dynamic interactions between the neighbors and items. Fig. 1(c) shows our main idea that models the temporally user–item heterogeneous graphs and generates a more accurate prediction. In the figure, the decision of whether a user \( u_t \) will choose a given item \( v_{t+1} \) is controlled by two important factors: 1) the historical interactions between him or her and items and 2) the temporal dynamic heterogeneous graph, including the interactions between the neighbors and items. Hence, the goal of the proposed method is to estimate the conditional probability of \( v_{t+1} \) given a user \( u_t \), the historical accessed item sequence \( v_{1:t} \), as well as the heterogeneous graph sequence \( H_{1:t+1} \), which can be formulated as \( P(v_{t+1}|u_t, H_{1:t+1}; v_{1:t}) \).

Based on the above idea, we propose the Temporally evolving aggregation (TEA) framework for sequential recommendation by aggregating the user behavior sequence as well as the dynamic user–item heterogeneous graph. Inspired by the sequence labeling in natural language processing [18], [19] to model the joint probability distribution, we adopt conditional random field (CRF) to model the item decision sequence and estimate \( P(v_{t+1}|u_t, H_{1:t+1}; v_{1:t}) \). In order to alleviate the issue of the large item space, we use the pseudo-likelihood method to approximate the aforementioned conditional probability. By doing this, the training procedure can be performed by estimating the unary score and transition score in CRF, which are implemented by our designed modules. Technically, we design a time-restricted user behavior sequence aggregation module to estimate the transition score of CRF and a temporal dynamic heterogeneous graphs aggregation module to estimate the unary scores of CRF. We further provide two different practical implementations based on the proposed framework. Extensive experimental studies demonstrate that our TEA framework outperforms the state-of-the-art recommendation methods on two published datasets and one real-world WeChat official accounts dataset.

The contributions of this article can be summarized as follows.

1) We formalize the sequential recommendation problem as a sequential decision problem that coincides with the CRF. To the best of our knowledge, this is the first
attempt to study sequential recommendation based on CRF.

2) We improve the conventional CRF and propose a unified framework that simultaneously aggregates the historical user behaviors and the dynamic interactions between users. Moreover, we also provide two practice implementations.

3) We compare the proposed TEA framework with the 14 compared method on three real-world datasets and achieve the state-of-the-art performance.

The remainder of this article is organized as follows. In Section II, we review related research into recommendation systems, including social recommendation and sequential recommendation. In Section III, we define the problem of sequential recommendation under the dynamical heterogeneous graph and further derive the objective function based on the CRF. In Section IV, we provide the implementation details of the proposed TEA model. We further analyze the connection to existing methods in Section V. Then, we present our experimental results based on two standard benchmarks and one real-world dataset in Section VI. Finally, we give our conclusion of the proposed method.

II. RELATED WORKS

In this section, we discuss the existing techniques on social recommendation and sequential recommendation and the works about heterogeneous graph learning.

In order to effectively mine the deep demands of users, researchers set their sights on social relations, and hence, social recommendation has received more and more attention. One of the most important methods is matrix factorization (MF) [20]–[22]. Based on the traditional MF methods, Hao et al. [23] proposed a co-factorization method, which shares a common latent user-feature matrix factorized by both ratings and social relations. With the development of deep learning methods, He et al. [24] proposed NeuMF by replacing the inner product with a neural architecture that can learn an arbitrary function from data. Fan et al. [25] proposed a deep neural network-based model to learn the nonlinear features of each user from social relations and to integrate them into probabilistic MF for social recommendation. Deng et al. [26] proposed a two-phase recommendation process to utilize deep learning to calculate the impact of community effect from the interests of users’ trusted friends for recommendations.

Recently, GNNs [15], [27] are widely used to aggregate node information and topological structure from social networks, and hence, GNNs are employed to address the social recommendation problem. In order to well aggregate the heterogeneous information, Fan et al. [5] proposed the GraphRec for the social recommendation. Fu et al. [28] leveraged the metapaths [29] to obtain the heterogeneous graph embedding. Considering that the influences in the social network may be context-dependent, Song et al. [16] addressed the session-based social recommendation by using a dynamic-graph-attention neural network architecture. However, the aforementioned methods rarely consider the fact that different friends in social networks choose different items. In this work, considering the fact that social influence and user behaviors are time-dependent, the proposed TEA method focuses on aggregating the temporally evolving social influence and the user behavior sequence.

Since users usually access the items in chronological order, users are likely to choose the items that are closely relevant to those they just accessed. Many works on sequential recommendation follow this assumption. Aiming to model the item–item transition probabilities, some traditional works borrow the idea of the MC. Rendle et al. [30] bridged the MF and MCs. He et al. [2] proposed TransRec to model such third-order relationships (e.g., the relationships among a user, the previously accessed item, and the next item) for large-scale sequential prediction. Motivated by the advantages of sequence learning in natural language processing, many neural network-based methods are proposed to learn the sequential dynamics. Tang and Wang [31] leveraged convolutional neural networks to encode the sequences into the embeddings. Other works [32], [33] leverage recurrent neural networks (RNNs) and their variants to model the sequences of items. Kang and McAuley [34] further leveraged the attention mechanism and proposed the SASRec to balance the goal of MC-based methods and RNN-based methods. Moreover, Sun et al. [35] argued that such left-to-right unidirectional models are suboptimal. Thus, they propose BERT4Rec, which employs deep bidirectional self-attention to model user behavior sequences. In this article, the proposed TEA leverages the CRF to model the translation of items, which calculates the transition score and the unary score by, respectively, aggregating the user behavior sequence information as well as the dynamic user–item heterogeneous graph.

Our works are also related to the link prediction task of the heterogeneous graph since recommending an item to a user can be considered as prediction a link between a user and an item in the bipartite graph. Schlichtkrull et al. [36] first introduced the relational graph convolutional networks (R-GCNs) to model the relation data and address the link prediction task in the heterogeneous graph. Recently, other researchers [37], [38] leverage the powerful attention mechanisms to select the key nodes for message passing and further model the heterogeneous graph data. Considering that the heterogeneous graph usually contains many schema, Manchanda et al. [39] performed message passing to incorporate information of neighbors multiple hops away by taking advantage of the schemas of the heterogeneous graphs.

III. MODEL

In this section, we begin with the problem definition of sequential recommendation. Then, we derive the unified objective function under conditional probability $P(v_t+1|u_t, H_{1:t+1}; v_{1:t})$.

A. Problem Definition

Let $U = \{u_1, u_2, \ldots, u_n\}$ and $V = \{v_1, v_2, \ldots, v_m\}$ denote the sets of users and items, respectively, in which $n$ is the number of users and $m$ is the number of items. For user–item
interactions, we let \( G = \{ U \cup V, E_h \} \) be the user–item bipartite graph with edges \((u_i, v_j) \in E_h\). As for user–user relations, we let \( G' = \{ U, E' \} \) be the social graph with edges \((u_i, u_j) \in E'\). If we combine the bipartite graph and the social graph, we obtain the following heterogeneous graph \( H = \{ U \cup V, E_h \cup E' \} \). Let \( v_{t+1} \) be the user behaviors sequence for \( u_t \). Since we consider the temporal evolving social influence, we let \( H_t+1 = \{ H_t, H_{t+1} \} \) be the heterogeneous graph sequence, where \( H_t = \{ U \cup V, E_h \cup E' \} \) and \( E_h^t \) is the user–item interactions in the \( t \)th time step. For user \( u_t \), given the behavior sequence \( v_{t+1} \) and the heterogeneous graph sequence \( H_{t+1} \) as well as the item \( v_{t+1} \), our goal is to estimate the conditional probability of \( P(v_{t+1} | v_{t+1}, u_t, H_{t+1}) \). The mathematical notation and the corresponding descriptions are summarized in the Nomenclature.

**B. Methodology**

We begin with the traditional CRF, which is a probabilistic graphical model widely used in sequence labeling [18]. CRF has shown to be very effective since it can jointly model the label decision by capturing the dependencies across adjacent labels. Considering the general definition of CRF, let \( x = \{ x_1, \ldots, x_T \} \) and \( y = \{ y_1, \ldots, y_T \} \) denote the observed sequence and its corresponding labels, respectively. Formally, the conditional distribution \( p(y|x) \) of linear chain CRF [40] is given by

\[
p(y|x) = \frac{1}{Z(x)} \exp \left( \sum_{t=1}^{T} f(x_t, y_t; \Theta_f) + \sum_{t=1}^{T} g(y_t, y_{t-1}; \Theta_g) \right)
\]

\[
Z(x) = \sum_{y} \exp \left( \sum_{t=1}^{T} f(x_t, y_t; \Theta_f) + \sum_{t=1}^{T} g(y_t, y_{t-1}; \Theta_g) \right)
\]

(1)

in which \( \Theta_f \) and \( \Theta_g \) are the trainable parameters.

There are three important components in the above CRF model: the partition function \( Z(x) \), the unary scores function \( f(x_t, y_t) \), and the transition scores function \( g(y_t, y_{t-1}) \). The partition function \( Z(x) \) is a normalization factor in order to obtain a probability. The unary scores function \( f(x_t, y_t) \) is used to estimate the probability of \( y_t \) given the observed \( x_t \). Also, the transition scores function \( g(y_t, y_{t-1}) \) is used to estimate the probability of \( y_t \) given \( y_{t-1} \).

The three components framework provides us a unified solution to aggregate both the historical behaviors of users and the dynamic social influence from the social networks. Following the formulation of CRF, the purpose of our model is to estimate the conditional distribution as follows:

\[
P(v_{t+1} | u_t, H_{t+1}) = \frac{1}{Z(H_{t+1}, u_t)} \exp \left( \sum_{t=1}^{T} f(H_{t+1}, u_t, v_{t+1}; \Theta_f) \right)
\]

\[
Z(H_{t+1}, u_t)
\]

(2)

in which \( f(H_{t+1}, u_t, v_{t+1}; \Theta_f) \) denotes the aggregation of temporal evolving social influence and \( g(v_{t+1}, v_t; \Theta_g) \) denotes the aggregation of user behaviors. In specific, \( f(H_{t+1}, u_t, v_{t+1}; \Theta_f) \) describes the relationship between the dynamic heterogeneous graph \( H_{t+1} \) and the available item \( v_{t+1} \) and \( g(v_{t+1}, v_t; \Theta_g) \) models the dependency between the available item \( v_{t+1} \) and the user behavior sequence.

However, it is almost impossible to calculate \( Z(H, u_t) \) since the sequence length is too large. In order to address this issue, we employ the pseudo-likelihood method as an effective approximation method [41, 42] and further derive the following estimation of the conditional probability:

\[
P(v_{t+1} | u_t, H_{t+1}) \approx \prod_{t} P(v_{t+1} | v_{t+1}, u_t, H_{t+1}).
\]

(3)

Combining (2) and (3), we further derive the following estimation of the conditional probability \( P(v_{t+1} | v_{t+1}, u_t, H_{t+1}) \):

\[
P(v_{t+1} | v_{t+1}, u_t, H_{t+1}) = \frac{\exp(f(H_{t+1}, u_t, v_{t+1}; \Theta_f) + g(v_{t+1}, v_{t+1}; \Theta_g))}{\sum_{v_{t+1} \in V} \exp(f(H_{t+1}, u_t, v_{t+1}; \Theta_f) + g(v_{t+1}, v_{t+1}; \Theta_g))}.
\]

(4)

Finally, we can obtain the objective function of our proposed model as follows:

\[
L_{\text{crf}} = \frac{1}{n} \sum_{i=1}^{n} \sum_{t=1}^{T} \log P(v_{t+1} | v_{t+1}, u_t, H_{t+1}).
\]

(5)

The aforementioned objective function is usually impractical because the size of the item set is very large and the computation cost is unaffordable. Inspired by [43], we employ the negative sampling strategy to obtain the tractable unified objective function of sequential recommendation as follows:

\[
L_{\text{crf}} = \frac{1}{n} \sum_{i=1}^{n} \sum_{t=1}^{T-1} \log \sigma(f(H_{t+1}, u_t, v_{t+1}; \Theta_f) + g(v_{t+1}, v_{t+1}; \Theta_g)) \]

\[
+ \sum_{k=1}^{n} \log \sigma(-f(H_{t+1}, u_t, v_k; \Theta_f) - g(v_k, v_{t+1}; \Theta_g))
\]

(6)

where \( \sigma \) is the sigmoid activation function and \( v_k \) is the negative item uniformly sampled from the whole item set \( V \).

The objective function enjoys an appealing physical meaning. It provides the insights into how to design the model for sequential recommendation: \( f(H_{t+1}, u_t, v_{t+1}; \Theta_f) \) models the information of temporal evolving heterogeneous graph in the forms of the unary energy function, meanwhile, \( g(v_{t+1}, v_{t+1}; \Theta_g) \) not only models the alternative item \( v_t \) but also the user behavior sequence in the form of the pairwise energy function. Based on the aforementioned objective function, we further illustrate the training process of the model shown in Algorithm 1.
Fig. 2. Framework of the TEA model for the sequential recommendation. (a) Overview of the proposed model, the temporally dependent heterogeneous graphs aggregated representation \( h_{t+1} \), the user behavior aggregated representation \( h_{v_{t+1}} \), and the item embedding \( q_{t+1} \) are fed into the CRF layer and \( p(v_{t+1}|u_t, v_{1:t}, H_{t+1}) \) is estimated. (b) Time-restricted user behavior sequence aggregation block is based on the user behavior sequence aggregation and the time-restricted aggregation. Note that the GRU used in this module is different from that in (a). (c) Dynamic temporally heterogeneous graph aggregation block, which is based on the bipartite graph aggregation and social network aggregation, takes \( H_t \) as input, and the arrows denote the message passing direction.

Algorithm 1 TEA Training Algorithm

1: \( \text{repeat} \)
2: Randomly select a batch of the users, then select the accessed items and the corresponding heterogeneous graph sequence;
3: Forward propagation;
4: Update \( \Theta_f \) and \( \Theta_g \) based on Equation (6);
5: \( \text{until} \) Max Iteration or Early Stopping.

IV. IMPLEMENTATION OF TEA FRAMEWORK

In this section, we provide the implementation details of the proposed TEA model. As shown in Fig. 2(a), our implementation takes both the aggregation of user behavior sequences and the aggregation of temporally dependent heterogeneous graphs into consideration and employs the gated recurrent unit (GRU) cells [44] and CRF layers to predict the final results. The details of the two aggregation modules are presented in Fig. 2(b) and (c), respectively. We will give detailed descriptions of these two aggregation modules in the following.

A. Time-Restricted User Behavior Sequence Aggregation for the Transition Scores

In this section, we will introduce the technical details of \( g(u_{t+1}, v_{1:t}; \Theta_g) \). Given user \( u_t \) and the corresponding behavior sequence \( v_{1:t} \), we aim to calculate the user-specific item transition score.

1) User Behavior Sequence Aggregation: Considering that the future behavior of a user is not only influenced by the latest accessed items but also the items that the user has accessed before, the user behavior sequence aggregation block should consider both the transition between items and the long-term dependency of items. Inspired by the great success of the self-attention mechanism [45] in various tasks such as machine translation, we propose an extension of the self-attention mechanism to model the personalized item transition and long-term dependency by simultaneously leveraging the item information and the position information. Formally, given the \( j \)th candidate item, we calculate the weights of each historical item as follows:

\[
\alpha_{\tau j} = \text{softmax} \left( \frac{W_Q (q_j + k_j)(W_K (q_{\tau} + k_{\tau}))^\top}{\sqrt{d}} \right), \quad \tau < j
\]  

where \( q_j \) is the embedding of \( v_j \), \( k_j \) is the position embedding at the \( j \)th position of the input sequence, \( W_Q \) and \( W_K \) are trainable projection parameters, \( \sqrt{d} \) is the scaling factor, and \( d \) is the dimension of the embedding. As a result, we can calculate the historical item aggregated representation as follows:

\[
z_j = \sum_{\tau=1}^{\tau=j-1} \alpha_{\tau j} W_V (q_\tau + k_\tau)
\]

in which \( W_V \) are trainable projection parameters.

2) Time-Restricted Aggregation: Since the temporal interactions between users and items are very sparse, for the users that contain limited social relationships and items interactions, it is hard to obtain an ideal user embedding for the sparse social substructure, and it is also difficult to obtain a debiased item embedding. Therefore, it is a challenging task to well aggregate the information from the users to the items and
vice versa. Fortunately, we find that the users that select the same items usually share the same interests and intent. Inspired by this intuition, we further proposed the time-restricted aggregation module.

First, we selected the walk with three nodes (e.g., USER–ITEM–USER) with the restriction of time window \( \tau \). In detail, given the interaction \((u_i, v_t)\), we find the other users that select the same item in the time window of \([t - \tau, t + \tau]\), where \( \tau \) is the window size. In our experimental implementation, we choose \( \tau = 60 \) days. Therefore, we can collect the \( \tau \)-restricted walks, for example, \( u_i - v_t - u' \). Sequentially, we employ another GRU to aggregate the information from the dense substructures to the sparse substructures, which can be formalized as follows:

\[
\mathbf{h}_{u_i, h_{v_t}, h_{u'}} = \text{GRU}(p_i, q_i, p'; \mathbf{W}_R).
\]

Note that we take the whole sequence \( u_i - v_t - u' \) as input of a GRU layer instead of a GRU cell and \( h_{u_i}, h_{v_t}, h_{u'} \) are the output corresponding to \( p_i, q_i, \) and \( p' \), respectively. \( \mathbf{W}_R \) are the trainable parameters.

3) Calculate the Transition Scores: In order to well perform the personalized user behavior sequence aggregation, we further add the user embedding \( p_i \) into the transformed item representation. Formally, we can calculate the transition score \( s_t \) for:

\[
s_t = (\mathbf{W}_g^{(3)}[\mathbf{h}_{u_i}^{(t)} \oplus h_{v_t}^{(t)} \oplus h_{u'}^{(t)}])^T q_j
\]

\[
h_{u_i}^{(t)} = p_i + \mathbf{W}_g^{(1)}(\text{ReLU}([\mathbf{W}_g^{(1)}v_t + b_0])) + b_2^{(2)}
\]

where \( \mathbf{W}_g^{(1)}, \mathbf{W}_g^{(2)}, \mathbf{b}_0 \), and \( b_2^{(2)} \) are the trainable parameters. For convenience, we let \( \Theta_g = \{\mathbf{W}_g, \mathbf{W}_K, \mathbf{W}_V, \mathbf{W}_f^{(1)}, \mathbf{W}_f^{(2)}, \mathbf{b}_0, \mathbf{b}_2^{(2)}, \mathbf{p}, \mathbf{q}, \mathbf{k}, \omega_R\} \) be the trainable parameters of \( g(v_t+1, v_t; \Theta_g) \).

B. Dynamic Temporal Heterogeneous Graphs Aggregation for the Unary Scores

In this section, we will introduce the details of the dynamic temporally heterogeneous graphs aggregation \( f(H_{t+1}, u_i, v_t, u_{t+1}; \Theta_f) \), which is used to calculate the unary scores. The dynamic temporally heterogeneous graphs aggregation contains the bipartite graph aggregation and the social network aggregation.

1) Bipartite Graph Aggregation: In this section, we aim to obtain the aggregated representation of the bipartite graphs at the \( t \)th time step. Given user \( u_i \) and the heterogeneous graph sequence \( H_{t+1} \), we first obtain the user-specific representation \( \mathbf{h}_i \) of \( H_t \). Specifically, we employ two different aggregated strategies and raise two variants of the proposed method: the GraphSAGE-based \([46]\) method (named TEA-S) and the graph attention network-based \([47]\) method (named TEA-A). More experimental details will be introduced in Section VI.

As for the TEA-S variation, we can obtain the user-specific representation \( \mathbf{h}_i \) as follows:

\[
\mathbf{h}_i = \text{ReLU}(\mathbf{W}_A \text{MEAN}(\mathbf{q}_k, \forall k \in I_i(N(u_i))))
\]

where \( \mathbf{W}_A \) are the trainable parameters, \( I_i(N(u_i)) \) denotes the items interacted by \( u_i \)'s neighbors at between the \( t \)th and \( (t + 1) \)th time step, and \( \text{MEAN} \) denotes the average-pooling operation.

As for the TEA-A variation, we aggregate the item information to the user with the help of the graph attention mechanism, which can be formulated as

\[
\mathbf{h}_i = \text{ReLU}\left( \sum_{j \in E_i(N(u_i))} \alpha_{ij} \mathbf{q}_j \right)
\]

where \( \alpha_{ij} \) is the weight of user \( u_i \) and item \( v_j \) and is defined as

\[
\alpha_{ij} = \frac{\exp(\text{LeakyReLU}(\mathbf{W}_A^T[\mathbf{W}_A \mathbf{q}_i \oplus \mathbf{W}_A \mathbf{q}_j]))}{\sum_{k \in E_i(N(u_i))} \exp(\text{LeakyReLU}(\mathbf{W}_A^T[\mathbf{W}_A \mathbf{q}_i \oplus \mathbf{W}_A \mathbf{q}_k]))}
\]

in which \( \mathbf{q}_i \) is the embedding of the item interacted by \( u_i \) at the \( t \)th time step, \( \oplus \) is the concatenation operation, \( \mathbf{W}_A \) and \( \mathbf{W}_A \) are trainable parameters, and LeakyReLU is the leaky version of a rectified linear unit.

In order to model temporally dependent heterogeneous graphs propagation, we set \( \mathbf{h}_i \) into the GRU \([44]\). The GRU cell operation at the \( t \)th time step can be formulated as

\[
\mathbf{h}_i = \text{GRUCell}(\mathbf{h}_i, \mathbf{h}_{i-1}; \mathbf{W}_G)
\]

where \( \mathbf{W}_G \) denotes all trainable parameters of the GRU cell.

2) Social Network Aggregation: To propagate the information of neighbors’ interests, we further aggregate the information from the social network. For simplicity, we only formulate the GraphSAGE aggregation as follows:

\[
\mathbf{h}_i = \text{ReLU}(\mathbf{W}_S \text{MEAN}(\mathbf{p}_k, \forall k \in N(u_i)))
\]

where \( \mathbf{W}_S \) is the trainable parameters.

3) Calculate the Unary Scores: Based on the aforementioned aggregation, we fuse the time-dependent representation \( \mathbf{h}_i \) and time-independent representation \( \mathbf{h}_i \) into one vector and calculate the social influence score \( s_f \), i.e., the output of unary scores function \( f(\cdot) \). It is formulated as

\[
s_f = \mathbf{h}_i^T \mathbf{q}_j
\]

\[
\mathbf{h}_i^T = \mathbf{W}_f^2 \text{ReLU}\left( \mathbf{W}_f^1[\mathbf{h}_i \oplus \mathbf{b}_f^1] + \mathbf{b}_f^1 \right) + \mathbf{b}_f^1
\]

where \( \mathbf{W}_f^1, \mathbf{W}_f^2, \mathbf{b}_f^1 \), and \( \mathbf{b}_f^1 \) are trainable parameters. In summary, we let \( \Theta_f = \{\mathbf{W}_A, \mathbf{W}_S, \mathbf{W}_G, \mathbf{W}_f^1, \mathbf{W}_f^2, \mathbf{b}_f^1, \mathbf{b}_f^1, \mathbf{p}, \mathbf{q}\} \) be the trainable parameters of \( f(H_{t+1}, u_i, v_t, u_{t+1}; \Theta_f) \).

C. Model Summarization

The total loss of our proposed model is summarized as follows:

\[
L = L_{crf} + \gamma L_{reg}
\]

where \( L_{reg} \) is the L2 normalization on all parameters and \( \gamma \) is a tradeoff hyperparameter.

Based on this objective function, our model is trained by the following procedure:

\[
(\Theta_g, \Theta_f) = \arg \min_{\Theta_g, \Theta_f} L
\]
TABLE I
STATISTICS OF THE DATASETS

| Dataset         | Epinions | Yelp     | WeChat   |
|-----------------|----------|----------|----------|
| # users         | 22,167   | 270,770  | 568,100  |
| # items         | 296,278  | 184,134  | 242,702  |
| # interactions  | 798,620  | 3,602,495| 9,422,722|
| # social links  | 335,813  | 5,974,526| 5,667,864|
| density         | 0.0121%  | 0.0072%  | 0.0068%  |
| social density  | 0.0724%  | 0.0081%  | 0.0018%  |

All parameters are jointly optimized using the Adam [48] algorithm.

In the testing, we estimate the probability of $P(v_{t+1}|v_{1:t}, u_i, H_{t+1})$ as follows:

$$P(v_{t+1}|v_{1:t}, u_i, H_{t+1}) = \sigma(f(H_{t+1}, u_i, v_{t+1}; \Theta_f) + g(v_{t+1}, v_{1:t}; \Theta_g)).$$

(19)

V. CONNECTIONS TO EXISTING MODELS

We will discuss the connections to the existing transition-based sequential recommendation methods. Most of the existing works of transition-based sequential recommendation methods [2], [30] are based on MCs. These methods mainly consider two important factors: 1) the interactions between users and items to capture the inherent intent of users and 2) the sequential dynamics between items to capture the relationships between items. Thus, we find that our method is more general and some of the existing works can be taken as special cases of ours. The detailed discussions for each work are as follows.

Regarding the work FPMC [30], it simplifies the huge state space problem by introducing the basket of items and consequently ignores the sequence information of historical items in each basket. In the contrast, our method utilizes the historical item sequence by using the self-attention mechanism with position embedding and is more general than FPMC.

Regarding the work TransRec [2], it models the personalized sequential behavior by using the personalized translation vectors and the previous item embedding to predict the next item but ignores the long-term dependencies since it only considers the relationships between any two items. Moreover, TransRec addresses the problem of the huge state space of items by introducing the subspace, while our method utilizes the negative sampling strategy. Thus, our method is more feasible and efficient to capture the dynamic social influence of the target users.

A. Datasets

We evaluate our proposed TEA framework on two public datasets (Epinions and Yelp) and a large-scale private dataset (WeChat Official Accounts Dataset). The statistics of datasets are summarized in Table I. The brief information of the datasets is given as follows.

1) Epinions: A benchmark dataset for the recommendation. In Epinions, a user can rate and give comments on items. Besides, a user can also select other users as their trusters, and we use the trust graphs (which are composed of the trust relationships) as the network information.

2) Yelp: An online review platform where users review local businesses (e.g., restaurants and shops). The user–item interactions and the social networks are extracted in the same way as Epinions.

3) WeChat Official Accounts Dataset: WeChat is a Chinese multipurpose messaging, social media, and mobile payment application developed by Tencent. Also, WeChat official accounts dataset is one of the functions. On the WeChat Official Platform, users can read and share articles. This dataset is constructed by user-article hitting such as records and user–user networks on this platform, which shares the similar interests. We also anonymize the data and conduct strict desensitization processing.

We preprocess the datasets following the approach in [2]. Specifically, for all these datasets, we follow the previous works [34], [35] and treat a rating or review as implicit feedback. We further use the timestamps to determine the sequence order of actions. We discard users and items with fewer than five associated actions. In cases where star ratings are available, we take the item with a rating higher than 3 as users’ positive feedback.

For data splitting, we employ the widely used leave-one-out evaluation [10], [24]. We hold out the latest interaction of each user as the test set and select the second latest interaction as the validation set. The remaining data are used for training.

B. Implementation Details

We use PyTorch to implement our model and deploy it on RTX 2080 GPU. Hyperparameter settings for all three datasets are given as follows: embedding dimension $d = 64$, batch size $B = 1024$, dropout rate $p_{drop} = 0.5$, $L2$ regularization weight $\gamma = 5e^{-4}$, negative sampling size $n_s = 50$, sequence truncation length $L_s = 50$, neighbor truncation length $L_n = 20$, and learning rate $\eta = 0.01$. We train all the methods with five different random seeds and report the means and standard deviations.

C. Evaluation Metrics

We evaluate all the models with two widely used Top-N metrics: Hit Rate@K (HR@K) and Normalized Discounted
TABLE II
PERFORMANCE EVALUATION OF THE COMPARED METHODS ON EPINIONS DATASET. THE VALUES PRESENTED ARE AVERAGED OVER FIVE REPLICAED WITH DIFFERENT RANDOM SEEDS

| Model Class          | Models          | HR@5      | NDCG@5   | HR@10     | NDCG@10  | HR@20     | NDCG@20   |
|----------------------|-----------------|-----------|----------|-----------|----------|-----------|-----------|
| Matrix Factorization based | BPRMF [10]      | 38.72±0.10 | 29.66±0.12 | 47.53±0.10 | 32.50±0.07 | 57.21±0.22 | 34.95±0.13 |
|                      | NeuMF [24]      | 41.35±0.59 | 31.13±0.69 | 51.15±0.43 | 34.31±0.64 | 60.93±0.34 | 36.78±0.59 |
|                      | SocialMF [7]    | 41.78±0.16 | 32.57±0.29 | 50.01±0.18 | 35.23±0.29 | 58.23±0.14 | 37.31±0.25 |
|                      | SoRec [23]      | 40.81±0.33 | 31.14±0.30 | 49.61±0.16 | 33.99±0.24 | 58.42±0.19 | 36.22±0.25 |
| Graph Neural Network based | R-GCNs [36]    | 32.98±0.44 | 23.95±0.27 | 43.28±0.66 | 27.27±0.34 | 56.06±0.37 | 30.48±0.27 |
|                      | HAN [37]       | 35.52±1.04 | 24.45±0.69 | 46.84±0.96 | 28.12±0.67 | 58.04±0.36 | 30.95±0.52 |
|                      | HGT [38]       | 42.00±0.21 | 31.61±0.14 | 51.72±0.08 | 34.76±0.10 | 60.09±0.14 | 36.88±0.02 |
|                      | GraphRec [5]   | 39.50±0.35 | 30.16±0.27 | 48.94±0.42 | 33.21±0.21 | 58.87±0.29 | 35.72±0.20 |
|                      | LightGCN [49]  | 42.59±0.07 | 32.20±0.09 | 51.92±0.08 | 35.22±0.07 | 60.54±0.09 | 37.41±0.08 |
|                      | DGRec [16]     | 40.36±0.25 | 30.52±0.16 | 49.67±0.14 | 33.53±0.15 | 59.26±0.19 | 35.95±0.15 |
| Sequence based       | DMAN [50]       | 35.15±0.27 | 27.06±0.33 | 45.01±0.06 | 30.23±0.24 | 55.85±0.27 | 32.98±0.30 |
|                      | TransRec [2]   | 44.79±0.12 | 36.09±0.21 | 52.51±0.11 | 38.58±0.17 | 60.98±0.11 | 40.72±0.07 |
|                      | SASRec [34]    | 43.32±0.20 | 33.97±0.20 | 51.88±0.20 | 36.74±0.20 | 60.31±0.20 | 38.87±0.18 |
|                      | ASAS [51]      | 44.97±0.34 | 35.59±0.29 | 53.44±0.29 | 38.33±0.27 | 61.41±0.29 | 40.35±0.28 |
| Ours                 | TEA-A           | 47.84±0.04 | 38.40±0.41 | 55.99±0.04 | 41.04±0.41 | 63.51±0.29 | 42.95±0.38 |
|                      | TEA-S           | 48.13±0.25 | 38.65±0.18 | 56.10±0.17 | 41.24±0.15 | 63.58±0.08 | 43.13±0.11 |

Cumulative Gain@$K$ (NDCG@$K$). HR measures the percentage that recommended items contain at least one correct item interacted by the user, while NDCG considers the positions of correct recommended items. In the context of sequential recommendation, since we only test on the latest item in a user behavior sequence, HR is identical to recall and proportional to precision [34].

Since it is time-consuming to rank all items for each user during the evaluation, we followed the strategy in [34]. Specifically, for each user, we randomly sample 100 negative items and rank these items with the ground-truth item. HR and NDCG are estimated based on the ranking results. We report the experiment results for $K = 5/10/20$.

D. Compared Methods

We compare our proposed models (TEA-S and TEA-A) based on TEA framework with three kinds of baselines: the MF-based models, the GNN-based models, and the sequence recommendation methods.

1) MF-Based Methods:
   a) BPRMF [10]: A general learning framework for personalized ranking recommendation uses implicit feedback.
   b) NeuMF [24]: It replaces the inner product with a multilayer perception (MLP) to learn the user–item interaction function.
   c) SocialMF [7]: It considers the social information and propagation of social information into the MF model.
   d) SoRec [23]: It performs co-factorization on the user–item rating matrix and user–user social relations matrix.

2) GNN-Based Methods:
   a) GraphRec [5]: It uses the GNN to combine user behavior information and social network information into the recommendation task. For fairness, we discard the opinion/rate embedding in our implementation.
   b) LightGCN [49]: A state-of-the-art graph-based CF method. It explicitly integrates a bipartite graph structure into the embedding learning process to model the high-order connectivity in the user–item interaction graph.
   c) DGRec [16]: A session-based recommendation method that combines the user action-temporal information and the social information via RNNs and dynamic graph attention networks.
   d) R-GCNs [36] use a different projection matrix for each relation in the heterogeneous graph and address the link prediction task.
   e) HAN [37] uses node- and semantic-level hierarchical attention to model heterogeneous graph networks.
   f) HGT [38] designs node- and edge-type-dependent parameters to characterize the heterogeneous attention.

3) Sequential Recommendation Methods:
   a) TransRec [2]: A sequential recommendation method that models each user as a translation vector to capture the transition from the current item to the next.
   b) SASRec [34]: It leverages the transformer [45] to capture users’ sequential behaviors.
   c) ASASRec [51]: An improved version of SASRec with an adversarial training strategy.
   d) DMAN [50]: It effectively utilizes the sequential data by segmenting the overall behavior sequence and maintaining the long-term interests of users.

4) Model Variants:
   a) TEA-S: We use the GraphSAGE-based aggregation method in the bipartite graph aggregation.
b) **TEA-A**: We use the graph attention mechanism-based aggregation method in the bipartite graph aggregation.

c) **TEA-RS**: We remove the time-restricted aggregation and use the GraphSAGE-based aggregation method in the bipartite graph aggregation.

d) **TEA-RA**: We remove the time-restricted aggregation and use the graph attention mechanism-based aggregation method in the bipartite graph aggregation.

### E. Results

Tables II–IV present the recommendation performance of all the methods on the three datasets. We do not report the performance of LightGCN, DMAN, and HAN on the WeChat Official Accounts dataset because of the limitation of GPU memory.

First, by modeling social influence, the performances of social-aware methods (SocialMF, SoRec, GraphRec, and DGRec) are improved compared with that of BPRMF in most cases, which is consistent with previous works. This observation indicates that social information reflects users' interests effectively. Second, the sequence-based methods (DGRec, TransRec, SAS, and ASAS) also perform comparably well. These improvements reflect the importance of temporal information on recommendation tasks. Third, DGRec and our proposed methods (including TEA-S and TEA-A) that combine social information and temporal information achieve much better performance, especially on large datasets. Finally, our proposed TEA-S and TEA-A consistently outperform all the compared methods on both public and real-world datasets with an average improvement of 3.15% on HR@10 and 8.38% on NDCG@10 against the best competitor. The significant improvements validate the effectiveness of aggregating the user behavior sequence and the influence between the users. To some extent, the experiment results reflect that the proposed time-restricted aggregation can mitigate the drawbacks of sparse social networks and user–item interactions.

### F. Ablation Study

In order to evaluate the effectiveness of the time-restricted aggregation, we remove the aforementioned aggregation module and obtain the variants **TEA-RS** and **TEA-RA**. The experimental results on each dataset are shown in Figs. 3–5. From these results, we can find that the models with time-restricted aggregation achieve a better performance, especially the results on the Yelp dataset. We also find that the promotion in the Epinions dataset is not so remarkable, and this is because the social networks in Epinions are much denser than that of Yelp. To some extent, the experiment results reflect that the proposed time-restricted aggregation can mitigate the drawbacks of sparse social networks and user–item interactions.
TABLE III
PERFORMANCE EVALUATION OF THE COMPARED METHODS ON THE YELP DATASET. THE VALUES PRESENTED ARE AVERAGED OVER FIVE REPLICATED WITH DIFFERENT RANDOM SEEDS

| Model Class          | Models   | HR@5   | NDCG@5 | HR@10  | NDCG@10 | HR@20  | NDCG@20 |
|----------------------|----------|--------|--------|--------|---------|--------|---------|
| Matrix Factorization based | BPRMF [10] | 66.33±0.27 | 52.46±0.16 | 76.51±0.26 | 55.77±0.16 | 84.59±0.22 | 57.82±0.15 |
|                      | NeuMF [24] | 70.38±0.26 | 56.14±0.28 | 79.35±0.12 | 59.06±0.24 | 86.14±0.12 | 60.79±0.22 |
|                      | SocialMF [7] | 64.82±0.24 | 49.69±0.24 | 76.27±0.28 | 53.42±0.21 | 84.99±0.28 | 56.63±0.19 |
|                      | SoRec [23] | 70.41±0.10 | 54.55±0.10 | 81.45±0.04 | 58.15±0.07 | 89.03±0.04 | 60.08±0.06 |
| Graph Neural Network based | R-GCNs [36] | 79.90±0.09 | 63.77±0.14 | 89.17±0.06 | 66.80±0.13 | 94.45±0.05 | 68.15±0.12 |
|                      | HAN [37] | 74.00±0.62 | 56.46±0.30 | 83.55±0.01 | 60.10±0.26 | 92.35±0.37 | 61.95±0.23 |
|                      | HGT [38] | 76.60±0.11 | 61.19±0.01 | 85.79±0.12 | 64.18±0.01 | 91.25±0.12 | 65.80±0.01 |
|                      | GraphRec [5] | 68.37±0.23 | 51.44±0.27 | 81.55±0.17 | 57.54±0.18 | 90.61±0.17 | 58.05±0.16 |
|                      | LightGCN [49] | 73.04±0.21 | 57.10±0.21 | 84.39±0.07 | 60.80±0.19 | 92.08±0.07 | 62.76±0.17 |
|                      | DGRRec [16] | 76.22±0.24 | 60.18±0.28 | 86.57±0.18 | 63.55±0.26 | 92.93±0.08 | 65.18±0.16 |
| Sequence based | DMAN [50] | 72.93±0.33 | 57.43±0.16 | 83.64±0.34 | 60.94±0.29 | 91.03±0.75 | 62.82±0.26 |
|                      | TransRec [2] | 75.81±0.15 | 60.63±0.16 | 80.19±0.20 | 64.00±0.15 | 93.13±0.12 | 65.78±0.15 |
|                      | SASRec [34] | 69.28±0.39 | 53.18±0.43 | 81.66±0.08 | 57.21±0.37 | 90.36±0.08 | 59.43±0.34 |
|                      | AASRec [51] | 72.97±0.13 | 56.76±0.16 | 84.53±0.04 | 60.53±0.09 | 92.18±0.04 | 62.48±0.07 |
| Ours                 | TEA-A | 81.13±0.25 | 66.79±0.36 | 88.65±0.14 | 69.24±0.33 | 93.50±0.10 | 70.43±0.21 |
|                      | TEA-S | 84.08±0.18 | 70.16±0.26 | 90.57±0.08 | 72.29±0.23 | 94.68±0.07 | 73.33±0.21 |

TABLE IV
PERFORMANCE EVALUATION OF THE COMPARED METHODS ON THE WECHAT DATASET. THE VALUES PRESENTED ARE AVERAGED OVER FIVE REPLICATED WITH DIFFERENT RANDOM SEEDS

| Model Class          | Models   | HR@5   | NDCG@5 | HR@10  | NDCG@10 | HR@20  | NDCG@20 |
|----------------------|----------|--------|--------|--------|---------|--------|---------|
| Matrix Factorization based | BPRMF [10] | 62.33±0.12 | 56.38±0.12 | 68.55±0.18 | 58.38±0.14 | 75.60±0.10 | 60.16±0.14 |
|                      | NeuMF [24] | 68.58±0.16 | 61.66±0.16 | 73.26±0.18 | 63.82±0.16 | 82.53±0.14 | 65.65±0.15 |
|                      | SocialMF [7] | 68.23±0.13 | 61.30±0.42 | 74.62±0.07 | 63.36±0.39 | 81.44±0.04 | 65.09±0.37 |
|                      | SoRec [23] | 73.66±0.04 | 66.43±0.11 | 79.60±0.02 | 68.36±0.10 | 85.56±0.03 | 68.66±0.09 |
| Graph Neural Network based | R-GCNs [36] | 77.57±0.18 | 66.43±0.28 | 83.04±0.13 | 68.83±0.24 | 91.04±0.07 | 70.38±0.22 |
|                      | HAN [37] | 74.53±0.11 | 65.70±0.10 | 81.45±0.06 | 67.94±0.08 | 88.02±0.03 | 69.61±0.08 |
|                      | HGT [38] | 66.04±0.03 | 52.17±0.29 | 76.80±0.25 | 55.66±0.26 | 85.72±0.18 | 57.93±0.24 |
|                      | GraphRec [5] | - | - | - | - | - | - |
|                      | LightGCN [49] | - | - | - | - | - | - |
|                      | DGRRec [16] | - | - | - | - | - | - |
| Sequence based | DMAN [50] | - | - | - | - | - | - |
|                      | TransRec [2] | - | - | - | - | - | - |
|                      | SASRec [34] | - | - | - | - | - | - |
|                      | AASRec [51] | - | - | - | - | - | - |
| Ours                 | TEA-A | - | - | - | - | - | - |
|                      | TEA-S | - | - | - | - | - | - |

G. The Sensitivity of Hyperparameters

1) Embedding Dimension: According to the experiment results shown in Fig. 6, we analyze the sensitivity of the embedding dimension $d$ by showing HR@10 and NDCG@10 of our proposed TEA-S with $d$ varying from 8 to 64 on the Epinions, Yelp, and Wechat datasets. According to the experiment results, we find that the experiment results ($d = 32$ and 64) of HR@10 on Epinions are slightly lower than the result of $d = 16$, but the experiment results of NDCG@10 on the other datasets reflect that larger dimension benefits the model performance and a small dimension ($d = 16$) is enough for TEA-S to achieve the best performance.

2) Sensitivity of the Number of Negative Samples: Figs. 7 and 8 show the sensitivity of the number of negative samples $n_{s}$ in (6) by showing HR@10 and NDCG@10 of our proposed TEA-S with $n_{s}$ varying from 1 to 100 on the Epinions, Yelp, and Wechat datasets. The variant with $n_{s} = 5$ performs comparably well, though using $n_{s} \geq 10$ still boosts performance especially on the large-scale dataset, which means that using more negative samples is helpful to estimate the item transition probability. The variant with $n_{s} = 50$ indicates that our model is stable with $n_{s}$.

VII. CONCLUSION

This article presents a TEA framework for the sequential recommendation. Beginning from the original CRF, we derive the unified objective function for the sequential recommendation, which leverages the social influence between users and the dynamic user–item heterogeneous graph. The proposed framework provides the insights and principles for designing the sequential recommendation model. We further provide two different implementations of the proposed framework. Experimental results on three real-world datasets show that the TEA framework outperforms state-of-the-art methods.

ACKNOWLEDGMENT

The authors would like to thank Lingling Yi and Li Li from WeChat for their help and support on this work.
Zijian Li received the B.S. degree in computer science from the Guangdong University of Technology, Guangzhou, China, in 2017, where he is currently pursuing the Ph.D. degree with the School of Computer. He is currently a Visiting Student with the Mohamed Bin Zayed University of Artificial Intelligence, Abu Dhabi, United Arab Emirates. His research interests cover a variety of different topics including causality, transfer learning, and their applications.

Ruichu Cai (Senior Member, IEEE) received the B.S. degree in applied mathematics and the Ph.D. degree in computer science from the South China University of Technology, Guangzhou, China, in 2005 and 2010, respectively. He is currently a Professor with the School of Computer Science and the Director of the Data Mining and Information Retrieval Laboratory, Guangdong University of Technology, Guangzhou. His research interests cover various topics, including causality, deep learning, and their applications.

Dr. Cai is a Senior Member of China Computer Federation (CCF). He was a recipient of the National Science Fund for Excellent Young Scholars, the Natural Science Award of Guangdong, and so on. He has served as the Area Chair for ICML 2022, NeurIPS 2022, and UAI 2022, and a Senior PC Member for AAAI for the term 2019-2022 and IJCAI for the term 2019–2022.

Fengzhu Wu received the bachelor’s degree in computer science and technology from the School of Computer, Guangdong University of Technology, Guangzhou, China, in 2019, where he is currently pursuing the master’s degree with the School of Computer. He was a Research Intern at WeChat, Shenzhen, China, in 2020. His current research interests include graph representation learning and transfer learning.

Yuguang Yan received the B.S. and Ph.D. degrees from the School of Software Engineering, South China University of Technology, Guangzhou, China, in 2013 and 2019, respectively. He was a Post-Doctoral Fellow with The University of Hong Kong, Hong Kong, from 2019 to 2021. He is currently with the School of Computer, Guangdong University of Technology, Guangzhou. His current research interests include transfer learning, optimal transport, and medical data analysis.

Yuexing Hao received the B.A. degree in computer science from Rutgers University–New Brunswick, New Brunswick, NJ, USA, in 2020, and the M.S. degree in computer science from Tufts University, Medford, MA, USA, in 2022. She is currently pursuing the Ph.D. degree with Cornell University, Ithaca NY, USA.

Hao Gu received the master’s degree in computer science and technology from the Harbin Institute of Technology, Harbin, China, in 2017. After joining WeChat, Shenzhen, China, he has been engaged in machine learning relevant work. Now, he works on the recommender system for WeChat Finder with Tencent Technology (Shenzhen) Company Ltd., Shenzhen.

Sili Zhang received the bachelor’s degree in network engineering from the Guangdong University of Technology, Guangzhou, China, in 2020, where he is currently pursuing the master’s degree with the School of Computer Science. He had an internship at ByteDance, Shenzhen, China, in 2019. His current research directions are graph representation learning and reinforcement learning.

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.