

# Dynamic Group Link Prediction in Continuous-Time Interaction Network

Shijie Luo, He Li\*, Xuejiao Li, and Tian Tian

Xidian University, Xi'an 710000, China  
heli@xidian.edu.cn

**Abstract.** Recently, group link prediction has received increasing attention due to its important role in analyzing relationships between individuals and groups. However, most existing group link prediction methods emphasize static settings or only make cursory exploitation of historical information, so they fail to obtain good performance in dynamic applications. To this end, we attempt to solve the group link prediction problem in continuous-time dynamic scenes with fine-grained temporal information. We propose a novel continuous-time group link prediction method CTGLP to capture the patterns of future link formation between individuals and groups. A new graph neural network CTGNN is presented to learn the latent representations of individuals by biasedly aggregating neighborhood information. Moreover, we design an importance-based group modeling function to model the embedding of a group based on its known members. CTGLP eventually learns a probability distribution and predicts the link target. Experimental results on various datasets with and without unseen nodes show that CTGLP outperforms the state-of-the-art methods by 13.4% and 13.2% on average.

**Keywords:** Group Link Prediction · Continuous-Time Interaction Network · Graph Neural Network (GNN).

## 1 Introduction

Link prediction, aiming to predict relationships between pairs of entities, has received wide attention as the increasing importance of network data [11, 14, 17]. Since it helps us understand the inherent characteristics and evolutionary mechanisms of real-world networks, link prediction has been widely applied in many practical applications, such as knowledge graph completion [22], biochemical reaction reconstruction [18] and content recommendation [13]. Almost all existing link prediction methods focus only on relationships between pairs of entities [27], but the analysis of relationships between individuals and groups (i.e., group link prediction) also deserves attention since the patterns of relationship formation are not exclusively limited to a pair of entities [24, 23].

Nevertheless, in many real scenarios, our focus is on the prediction of future relationships between individuals and groups, such as organizers of hobby clubs expecting to invite target participants to their future events. The task

of predicting future relationships between individuals and groups is known as *continuous-time group link prediction*. Despite recent efforts, three deficiencies remain in addressing continuous-time group link prediction problem with fine-grained temporal information. First, previous methods rarely discuss future links between individuals and groups, but tend to mine missing ones. The assumption that all members are connected to the group at the same time makes the fine-grained raw temporal information missing. Second, individuals are assumed to be isolated from each other, which neglects the neighborhood information that laterally depicts dynamic link preferences. Third, equal treatment of all group members leads to ignoring the diversity of members’ importance in groups.

In this paper, we propose CTGLP, a novel continuous-time group link prediction method, to infer future relationships between individuals and groups in continuous-time dynamic networks with fine-grained temporal information. We first present CTGNN, a new graph neural network (GNN) with a continuous-time neighbor sampling strategy, to learn the embeddings of individuals, where a novel aggregation function is designed to jointly capture neighborhood features and fine-grained temporal information. Second, an importance-based group modeling function is provided to model the latent representation of a group based on the embeddings of existing members. Finally, CTGLP outputs conditional probability distributions by using the embeddings of groups and finds out the link targets.

The contributions of this paper are summarized as follows.

1. We propose a novel continuous-time group link prediction method CTGLP, which learns the patterns of link formation between individuals and groups in continuous-time dynamic networks with fine-grained temporal information and predicts the future links between individuals and groups.

2. We propose a new graph neural network CTGNN to learn the representations of individuals. A continuous-time neighbor sampling strategy is designed to control the computational consumption, and an aggregation function CTA<sub>g</sub> is presented to bias the aggregation weights of the features of sampled neighbors.

3. We propose an importance-based group modeling function that models the groups into the latent space by measuring the importance of each member to the group based on the time of link formation.

4. Extensive experiments on various datasets with and without unseen nodes are conducted to validate CTGLP and the experimental results demonstrate that CTGLP outperforms the baselines by a significant margin, with average gains of 13.4% and 13.2%.

## 2 Related Work

### 2.1 Link prediction

Based on the network structural similarity, heuristic link prediction methods, such as Common Neighbors (CN) [12] and Adamic-Adar (AA) [1], assume that edges are more likely to exist between nodes with higher similarity scores. However, they only exploit shallow topological features of networks and lack general

applicability. Besides, embedding techniques have also shown great potential in link prediction. Some embedding algorithms, such as DeepWalk [19], node2vec [6], HTNE [28], etc., calculate the possibility of generating links between nodes using the embeddings. Nevertheless, some studies [16, 4] have demonstrated that embedding models may be inferior to well-designed mechanistic methods. Recently, some well-designed methods have shown their superiority in link prediction. TDGNN [21] leverages temporal information in dynamic networks to achieve continuous-time link prediction. GNMFCFA [15] predicts future links using global and local information of temporal networks. GC-LSTM [3] applies an embedded Long Short-Term Memory (LSTM) of Graph Convolutional Network to perform dynamic link prediction. Dyngraph2vec [5] integrates longer-term temporal information to learn node embeddings and predict future links. LP-ROBIN [2] utilizes incremental embedding to capture temporal dynamics and predict new connections. Despite the great success of link prediction, existing link prediction methods cannot be directly applied to group link prediction focusing on the relationships between individuals and groups because they only concentrate on the relationships between node pairs.

## 2.2 Group link prediction

Due to the inevitable limitations of link prediction methods applied to group link prediction, recent attempts have been made to solve the group link prediction problem. An LSTM-based model [24] is elaborately designed to address this problem. Feeding the sum of random vectors of members in a series of groups into LSTM, this model learns the embedding vectors of members and trains a classifier to predict the target. Despite the input of a series of group vectors, it ignores the neighborhood information that potentially expresses link preferences and may introduce information noise. Subsequently, a CVAE-based model [23] is proposed to estimate the probability of link existence by tuning the model parameters in a supervised manner. However, the absence of historical interaction information between individuals and groups prevents the model from addressing the problem of continuous-time group link prediction well. To further leverage historical information, CVAEH [23] additionally introduces a vector that encodes the previous groups. Nevertheless, the rough encoding of historical group information is still not a good solution. Summarizing existing group link prediction methods, they do not consider the fine-grained historical group information and fail to generalize to unseen data. Therefore, how to infer the future relationships between individuals and groups more effectively remains an open question.

## 3 Preliminaries

### 3.1 Definitions and Problem

**Definition 1 (Continuous-time interaction network).** *A continuous-time interaction network  $G = (V, E^T, T)$  consists of node set  $V$ , edge set  $E^T$  and*

time set  $T$ , where  $v_i \in V$  denotes the node in the network.  $e_{ij}^t \in E^T$  denotes the edge/interaction between node  $v_i$  and node  $v_j$  at time  $t \in T$ .

**Definition 2 (Group).** Individuals jointly participating in a certain event are denoted as a group, i.e.,  $s_i = \{v_{i,1}^{t_1}, v_{i,2}^{t_2}, \dots, v_{i,k}^{t_k}\} \subseteq V$ , where  $i$  denotes the index of a group.  $v_{i,k}^{t_k}$  denotes the  $k$ -th member node of group  $s_i$ , where  $t_k$  denotes the link time and  $k \geq 2$  indicates that the number of members in a group should not be less than two.  $s_i \subseteq V$  indicates that the members of group  $s_i$  are from node set  $V$ .

*Problem 1 (Continuous-time group link prediction).* Given a continuous-time interaction network  $G = (V, E^T, T)$ , there is a node set  $V = \{v_1, v_2, \dots, v_N\}$  with  $N$  nodes and a group set  $S = \{s_1, s_2, \dots, s_M\}$  with  $M$  groups. For a group  $s_i = \{v_{i,1}^{t_1}, v_{i,2}^{t_2}, \dots, v_{i,k}^{t_k}\} \subseteq V$ ,  $\max(t_1, t_2, \dots, t_k) \leq t$  with  $k$  members observed at current time  $t$ , the purpose of continuous-time group link prediction is to predict the target  $v_{i,k+1}^{t'} \in V \setminus s_i$  that is most likely to be linked to group  $s_i$  at the future time  $t'$  ( $t' > t$ ) based on the  $k$  known members in group  $s_i$ . Formally, it is defined as:

$$v_{i,k+1}^{t'} = \mathcal{F}(v_{i,1}^{t_1}, v_{i,2}^{t_2}, \dots, v_{i,k}^{t_k}), \quad (1)$$

where  $\mathcal{F}$  is the continuous-time group link prediction function.  $\{v_{i,1}^{t_1}, v_{i,2}^{t_2}, \dots, v_{i,k}^{t_k}\}$  denotes the  $k$  known members of group  $s_i$  observed at current time  $t$ .

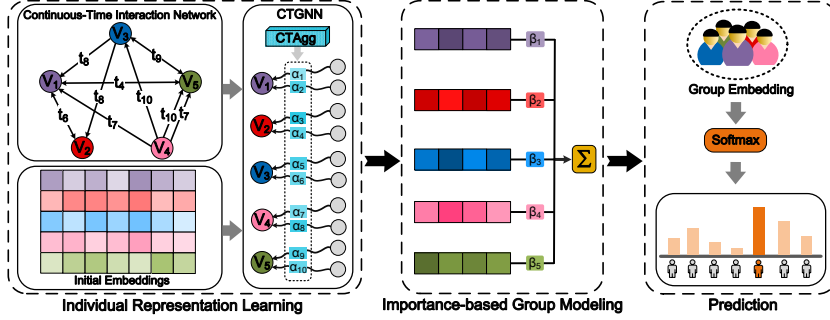
## 4 Methodology

Figure 1 shows the architecture of CTGLP. It consists of three main components: 1) Individual Representation Learning, 2) Importance-based Group Modeling and 3) Prediction. The Individual Representation Learning component aims to learn the latent embeddings of individuals using our proposed CTGNN. The Importance-based Group Modeling component is to model the latent representations of groups. The Prediction component aims to predict the targets.

### 4.1 Individual Representation Learning

Given a continuous-time interaction network  $G = \{V, E^T, T\}$  with the initial random embeddings of nodes  $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$ ,  $\mathbf{x}_i \in \mathbb{R}^D$ , individual representation learning part obtains the final latent embeddings of members  $\mathbf{Z} = \{\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_N\}$ ,  $\mathbf{z}_i \in \mathbb{R}^d$  using our proposed CTGNN, where  $D$  and  $d$  are dimensions.

**Continuous-Time Neighbor Sampling** Whereas previous GNNs simply examine  $k$ -hop neighborhood or the sampling schemes are only applicable to static networks [26], the sampling process in all convolutional layers of our CTGNN are continuous-time respected, i.e., the time of the sampled edge should be less than that of the sampled edge of the previous layer. It can be ensured that all sampled neighbors of the central node exist in the past with respect to the central node, thus ensuring that all sampled neighborhood information exists



**Fig. 1.** The overall framework of CTGLP. CTGLP is composed of three main components: individual representation learning, importance-based group modeling and prediction.

prior to the current time during aggregation. We first define the time-limited neighbor set  $\Gamma_{\mathcal{T}}(u)$  of node  $u$  at time  $\mathcal{T}$ :

$$\Gamma_{\mathcal{T}}(u) = \{(v, t) \mid e = (u, v, t) \in E^T \cap t < \mathcal{T}\}. \quad (2)$$

Notably, node  $v$  may appear multiple times in  $\Gamma_{\mathcal{T}}(u)$  as multiple edges may exist between the same pair of nodes.

Then, in one convolutional iteration, we sample a fixed number of neighbors from  $\Gamma_{\mathcal{T}}(u)$  for node  $u$ :

$$Samp = \begin{cases} \Gamma_{\mathcal{T}}(u), & |\Gamma_{\mathcal{T}}(u)| \leq \theta; \\ r_{\theta}(\Gamma_{\mathcal{T}}(u)), & |\Gamma_{\mathcal{T}}(u)| > \theta, \end{cases} \quad (3)$$

where  $r_{\theta}(\cdot)$  is the random sampling operation.  $\theta$  is the neighbor sampling size and  $\theta$  may be different for each layer.

Performing multiple sampling, CTGNN obtains higher-order neighbors and reduces the number of neighbors involved in the computation. The  $l$ -order continuous-time sampled neighbor set of node  $u$  at time  $\mathcal{T}$  can be obtained by performing neighbor sampling operations  $l$  times:

$$\hat{\mathcal{N}}_{\mathcal{T}}^l(u) = Samp_l(\Gamma_{\mathcal{T}_l}^l(\dots Samp_1(\Gamma_{\mathcal{T}_1}^1(u)))) , \quad (4)$$

where  $Samp_l$  represents the sampling operation in the  $l$ -th convolutional layer.  $\mathcal{T}_{i+1} < \mathcal{T}_i$  for  $1 \leq i < l$ , and  $\mathcal{T}_1 = \mathcal{T}$ .

**Embedding Update** By iteratively aggregating neighborhood features, GNNs learn the embeddings of nodes. A simple but effective aggregation scheme is mean operator [10, 8], which assumes that all neighbors of a central node contribute equally to the update of its new representation. However, mean operator may not be the optimal aggregation scheme for representation learning in continuous-time group link prediction, since the impact of different neighbors on the central node may vary dramatically depending on the time of link formation. Inspired

by a study in event-based social networks [20], we argue that the groups users recently linked to are typically more representative of their preferences than those they linked to earlier. The manifestation of this insight in aggregation is that a newly connected neighbor has a higher contribution to the embedding update of the central node.

We provide an aggregation coefficient  $\alpha$  calculated by our aggregation function CTagg to bias the contribution of each neighbor. Given the edge time  $t_{ij}$  between node  $u_i$  and  $u_j$  as well as the edge time  $t_{ik}$  between node  $u_i$  and  $u_k$ , if  $t_{ij} > t_{ik}$ , the aggregation coefficient  $\alpha_{ij}^t$  of node  $u_j$  should be greater than the aggregation coefficient  $\alpha_{ik}^t$  of node  $u_k$ . At the  $l$ -th layer of CTGNN, the embedding update of node  $u$  at time  $t$  can be denoted as follows:

$$\mathbf{n}_u^{(l)} = \text{AGG}^{(l)}(\{\alpha_{uv}^t \cdot \mathbf{h}_v^{(l-1)}, v \in \hat{\mathcal{N}}_t^l(u)\}), \quad (5)$$

$$\mathbf{h}_u^{(l)} = \sigma(\mathbf{W}^{(l)} \cdot \text{COM}(\mathbf{h}_u^{(l-1)}, \mathbf{n}_u^{(l)}) + \mathbf{w}^{(l)}), \quad (6)$$

where  $\text{AGG}(\cdot)$  is a function that aggregates the information of sampled neighbors and  $\text{COM}(\cdot)$  is a function that combines information about sampled neighborhoods and the pre-update information of the central node in the previous layer.  $\sigma$  is a nonlinear activation.  $\hat{\mathcal{N}}_t^l(u)$  is the  $l$ -th hop sampled neighbor set of node  $u$  at time  $t$ .  $\mathbf{W}$  and  $\mathbf{w}$  are learnable shared parameter matrices.  $\alpha_{uv}^t$  is the aggregation coefficient of neighbor  $v$  at time  $t$ , and it can be interpreted as the contribution of  $v$  to the embedding update of the central node  $u$  at time  $t$ . The calculation of  $\alpha_{uv}^t$  is defined as:

$$\alpha_{uv}^t = \frac{\exp(t_{uv} - t)}{\sum_{v \in \hat{\mathcal{N}}_t^l(u) \cup u} \exp(t_{uv} - t)}, \quad (7)$$

where  $t_{uv}$  is the time of the edge between nodes  $u$  and  $v$ .

After obtaining the embedding  $\mathbf{h}_u$  of node  $u$  output by the last convolution iteration, a Multiple-layer Perceptron (MLP) with activation functions is employed to attain the final representation  $\mathbf{z}_u$  of node  $u$ :

$$\begin{aligned} \mathbf{e}_u^{(1)} &= \sigma(\mathbf{U}^{(1)} \cdot \mathbf{h}_u + \mathbf{u}^{(1)}), \\ \mathbf{e}_u^{(2)} &= \sigma(\mathbf{U}^{(2)} \cdot \mathbf{e}_u^{(1)} + \mathbf{u}^{(2)}), \\ &\dots \\ \mathbf{z}_u &= \sigma(\mathbf{U}^{(j)} \cdot \mathbf{e}_u^{(j-1)} + \mathbf{u}^{(j)}), \end{aligned} \quad (8)$$

where  $j$  is the index of neural layers.  $\mathbf{U}$  and  $\mathbf{u}$  are learnable parameter matrices.

## 4.2 Importance-based Group Modeling

The practice of previous work [24] is to sum up the vectors of all group members as the vector of the group. While it is intuitively sound, it ignores the fact that the influence of different members on the group may differ greatly. To this end,

we present an importance-based group modeling strategy to represent groups into the latent space.

We first define an importance factor  $\beta$  to measure the importance of each group member. The value of the importance factor depends on the time when members are linked to the group, i.e., more recent members have larger importance factor values as they are intuitively more in line with the group formation trend. Formally, the importance factor of member  $k$  on group  $s_i$  is defined as:

$$\beta_{ik} = \frac{\frac{1}{\log(T-t_k)}}{\sum_{j=1}^K \frac{1}{\log(T-t_j)}}, \quad (9)$$

where  $K$  is the number of members in group  $s_i$  and  $T$  is the prediction time for group  $s_i$ .

Given a group  $s_i$  with  $K$  members (i.e.,  $s_i = \{v_{i,1}, v_{i,2}, \dots, v_{i,K}\}$ ) and the embeddings of these  $K$  members (i.e.,  $\{\mathbf{z}_{i,1}, \mathbf{z}_{i,2}, \dots, \mathbf{z}_{i,K}\}$ ), the importance-based group modeling part models the vectors of members as the vector  $\mathbf{m}_i \in \mathbb{R}^s$  of group  $s_i$ , where  $s$  is the size of the group vector. The importance-based group modeling for group  $s_i$  is:

$$\mathbf{p}_i = \sum_{j=1}^K \beta_{ij} \cdot \mathbf{z}_{i,j}, \quad (10)$$

$$\mathbf{m}_i = \mathbf{C}_2 \cdot \sigma(\mathbf{C}_1 \cdot \mathbf{p}_i + \mathbf{c}), \quad (11)$$

where  $\mathbf{C}_1$ ,  $\mathbf{C}_2$  and  $\mathbf{c}$  are learnable parameter matrices.

### 4.3 Prediction

The final outputs of the importance-based group modeling part are fed into an MLP, and a Softmax activation function is employed to generate the link probability distributions between candidate individuals and groups. Formally, given the latent vector  $\mathbf{m}_i$  of group  $s_i$ , the prediction process is:

$$\begin{aligned} \mathbf{q}_i^{(1)} &= \sigma(\mathbf{G}^{(1)} \cdot \mathbf{m}_i + \mathbf{g}^{(1)}), \\ &\dots \\ \mathbf{q}_i^{(k)} &= \sigma(\mathbf{G}^{(k)} \cdot \mathbf{q}_i^{(k-1)} + \mathbf{g}^{(k)}), \end{aligned} \quad (12)$$

$$\mathbf{P}_i = \text{Softmax}(\mathbf{Q} \cdot \mathbf{q}_i^{(k)} + \mathbf{q}) \quad (13)$$

where  $\mathbf{G}$ ,  $\mathbf{Q}$ ,  $\mathbf{g}$  and  $\mathbf{q}$  are learnable parameter matrices.  $k$  is the index of hidden layers.  $\mathbf{P}_i$  is a link probability distribution whose elements represent the connection possibility between individuals and group  $s_i$ . The index corresponding to the element with the largest value in  $\mathbf{P}_i$  is the index of the target predicted by CTGLP.

---

**Algorithm 1** Training of CTGLP

---

**Input:** Continuous-time interaction network  $G = (V, E^T, T)$ ; set of node initial embeddings  $\{\mathbf{x}_v, \forall v \in V\}$ ; set of groups  $S = (s_1, s_2, \dots, s_M), s_i \subset V$ .

**Parameter:** Convolutional layer depth  $K$ ; neighbor sampling size  $\theta_k, \forall k \in \{1, \dots, K\}$ ; learnable parameters.

**Output:** Continuous-time group link prediction function  $\mathcal{F}$ .

- 1: **while** *model not converge* **do**
- 2:   **for**  $i = 1 : M$  **do**
- 3:     **for**  $u \in s_i$  **do**
- 4:        $\hat{\mathcal{N}}_{\mathcal{T}}^K(u) \leftarrow \text{Sample-Neighbors}(u, G, K, \mathcal{T}, \theta)$
- 5:        $\mathbf{z}_u \leftarrow \text{Update-Embedding}(\mathbf{x}_u, \mathbf{x}_{\hat{\mathcal{N}}_{\mathcal{T}}^K(u)}, K)$
- 6:     **end for**
- 7:      $\mathbf{m}_i \leftarrow \text{Group-Modeling}(\{\beta_u \cdot \mathbf{z}_u, u \in s_i\})$
- 8:     Calculate link probabilities:  
       $\mathbf{P}_i \leftarrow \text{Softmax}(\mathbf{m}_i, \mathbf{G}, \mathbf{g}, \mathbf{Q}, \mathbf{q})$
- 9:     Obtain the target:  $u_i \leftarrow \text{argmax}(\mathbf{P}_i)$
- 10:   **end for**
- 11:   Update parameters by stochastic gradient descent
- 12: **end while**
- 13: **return**  $\mathcal{F}$

---

#### 4.4 Training

Algorithm 1 shows the overall process of training. Let  $\mathbf{y}_i$  denote the one-hot encoding of the target in the  $i$ -th training sample and  $\mathbf{P}_i$  be the link probability distribution output by CTGLP. Our objective function is formulated as:

$$\mathcal{L} = \frac{1}{M} \sum_{i=1}^M \sum_{j=1}^N y_{ij} \log(P_{ij}) \quad (14)$$

where  $M$  denotes the number of group samples.  $N$  denotes the number of nodes in the node set  $V$ .  $y_{ij}$  and  $P_{ij}$  are the  $j$ -th element of  $\mathbf{y}_i$  and  $\mathbf{P}_i$ , respectively.

## 5 Experiments

### 5.1 Experimental Setup

**Datasets.** MovieLens-100K (ML100K for short) [9] and MovieLens-25M (ML25M for short) [9] contains rating data from users on movies. CiaoDVD [7] consists of DVD rating data. We select the rating data and regard the set of users who rate the same item as a group and take out the last member of each group as our prediction target. The number of members in a group are set to be 3 to 20. To construct the datasets without unseen nodes, we remove nodes that appear in validation and testing but not in training, and further clean the data. The statistics of datasets are shown in Table 1.



Datasets	Nodes	Edges	Groups	Unseen*
ML100K	755	59 118	590	42
CiaoDVD	8 714	165 598	4 040	1 195
ML25M	16 065	1 048 836	18 882	1 578
ML100K <sub>w/o</sub>	650	22 683	510	0
CiaoDVD <sub>w/o</sub>	5 766	85 562	3 341	0
ML25M <sub>w/o</sub>	9 998	491 704	16 494	0

\* The value indicates the number of nodes that are not presented during training.

**Table 1.** Statistics of two versions of the three datasets. Note that the subscript w/o denotes the dataset without unseen nodes.

**Metrics.** To evaluate the performance of methods, we introduce three widely used evaluation metrics: Hit Ratio@ $K$ , Normalized Discounted Cumulative Gain@ $K$  and Mean Reciprocal Rank@ $K$  (denoted as HR@ $K$ , NDCG@ $K$  and MRR@ $K$  respectively). HR@ $K$  measures the ability to find the target and emphasizes the accuracy of prediction. NDCG@ $K$  and MRR@ $K$  measure the ability to rank targets and emphasize the ranking of targets.

**Baselines.** We compare our CTGLP with the following baselines: (1) Three group link prediction methods. **LSTM-based model** (LSTM for short) [24] employs LSTM to combine the historical information of groups and outputs link probabilities. **CVAE-based model** (CVAE for short) [23] uses CVAE to reconstruct the membership of groups using a vector encoding an entire group and a vector encoding known members. **CVAEH** [23] additionally introduces a vector encoding historical information of previous groups for prediction. (2) Two neural network-based methods. **MLP** [25] utilizes mini-batch gradient descent strategy to update model parameters. **GraphSAGE** (GSAGE for short) [8] uses multi-layer aggregation functions to learn representations of nodes and obtains group vectors to make prediction. (3) Two heuristic link prediction methods. **AA** [1] utilizes correlation coefficients of overlapping neighborhoods between nodes to measure the similarity between nodes. **CN** [12] uses the number of common neighbors between nodes to measure the similarity between nodes. Nodes with higher similarity to members of a group are considered more likely to link to the group. (4) Three network embedding methods. **DeepWalk** (DW for short) [19] uses random walks to generate vectors of nodes. **node2vec** (n2v for short) [6] learns node representations using biased random walks. **HTNE** [28] integrates the Hawkes process and attention mechanism to learn the time-related representations of nodes. The individual-group link scores are obtained by aggregating the similarities between individuals.

**Implementation details.** For each dataset, we split it into 8:1:1 for training, validation and testing. We implement our CTGLP with PyTorch 1.6.0 and adopt the SGD as the optimizer. We apply batch normalization and dropout strategy with  $p = 0.5$  for ML100K and CiaoDVD and  $p = 0.1$  for ML25M. The dimension  $D$  of initial embeddings, the dimension  $d$  of hidden states and the dimension  $s$  of group vectors are all tested in  $\{16, 32, 64, 128, 256, 512\}$ . The batch size and learning rate are searched in  $\{32, 64, 128, 256\}$  and  $\{0.005, 0.01, 0.05, 0.1\}$  respectively. Two convolutional layers are employed in CTGNN, and the neighbor sampling sizes are empirically set to 25 and 10 respectively. For the parameters of our method, we initialize it randomly using a uniform distribution with values from 0 to 1. For baselines, we initialize the parameters according to the corresponding paper.

## 5.2 Performance Comparison

**Performance on datasets *with* unseen nodes.** From the overall results in Table 2, we observe that CTGLP outperforms most of the competing methods by a comfortable margin. On ML100K, our method obtains better performance than all baselines, especially in ranking targets. Specifically, CTGLP achieves average gains of 7.3%, 31.9% and 43.9% in terms of HR, NDCG and MRR scores. On CiaoDVD, CTGLP outperforms other baselines except for CVAE when  $K = 10$ . On ML25M, our method shows its great superiority in finding and ranking targets. In terms of three evaluation metrics, CTGLP obtains average gains of 11.45%, 14.7% and 11.1%, respectively.

**Performance on datasets *without* unseen nodes.** From the overall results in Table 3, we can see that CTGLP always achieves the best performance or the second-best one. On  $ML100K_{w/o}$ , our method is slightly worse than CVAE in HR scores, but it brings average gains of 22.4% and 45.9% in NDCG and MRR scores, which indicates that CTGLP can rank targets better. It is worth noting that most methods, including CTGLP, perform worse on this dataset than on ML100K. We argue that the removal of group members may affect the intrinsic nature of small dataset to a greater extent. On  $CiaoDVD_{w/o}$ , our method always outperforms all competing models. Specifically, in three different metrics, CTGLP outperforms the best comparative method by 10.1%, 26.6% and 36.8% on average. On  $ML25M_{w/o}$ , CTGLP is inferior to heuristic link prediction methods AA and CN in the ability to rank targets, but it is still satisfactory in finding targets (obtains average gain of 13.9%). Besides, task-independent embedding-based methods perform quite poorly, indicating that they are not suitable for continuous-time group link prediction. Compared to the performance on ML25M, the gains obtained by the three well-designed group link prediction methods (LSTM, CVAE and CVAEH) are much smaller than those of our method, suggesting that the lack of fine-grained temporal information and neighborhood features does limit the performance improvement of models.

**Summaries.** From the above analysis, several conclusions are drawn: (1) Our proposed CTGLP outperforms most baselines in datasets with and without unseen nodes, especially in finding targets. (2) Neighborhood information (the

	Method	HR@K(%)		NDCG@K(%)		MRR@K(%)	
		K=10	K=20	K=10	K=20	K=10	K=20
ML100K	LSTM	15.9	22.7	8.7	10.4	6.6	7.0
	CVAE	28.8±1.5	39.0±1.7	16.1±1.4	18.6±1.4	12.3±2.2	13.0±2.3
	CVAEH	23.7±1.9	32.2±1.0	12.3±1.1	14.5±0.8	8.9±1.3	9.5±1.3
	MLP	14.4±1.1	26.1±1.2	6.3±1.4	9.3±1.2	4.1±1.2	4.7±1.4
	GSAGE	27.1±0.7	35.6±0.4	16.2±0.4	18.3±0.4	12.9±0.6	13.5±0.5
	CTGLP	<b>30.5±0.9</b>	<b>42.4±0.5</b>	<b>21.5±0.9</b>	<b>24.4±0.8</b>	<b>18.6±0.7</b>	<b>19.4±0.7</b>
CiaoDVD	LSTM	10.6	14.7	6.0	7.0	4.5	4.8
	CVAE	<b>21.6±0.7</b>	27.1±0.8	<b>11.9±0.5</b>	13.4±0.3	<b>9.0±0.4</b>	<b>9.4±0.3</b>
	CVAEH	16.1±0.8	23.5±1.9	10.0±1.1	11.9±1.3	8.2±1.1	8.7±1.2
	MLP	15.2±1.5	20.5±2.6	7.2±0.6	8.6±0.8	4.8±0.5	5.2±0.4
	GSAGE	17.6±0.8	24.8±0.8	8.8±0.7	10.6±0.5	6.1±0.5	6.5±0.6
	CTGLP	20.8±0.6	<b>28.7±0.7</b>	11.7±0.4	<b>13.7±0.2</b>	8.8±0.7	<b>9.4±0.8</b>
ML25M	LSTM	20.5	25.3	13.9	15.1	11.9	12.2
	CVAE	22.6±2.1	26.9±2.1	16.6±1.6	17.7±1.7	14.7±1.5	15.0±1.5
	CVAEH	19.4±0.8	23.4±0.6	14.3±1.1	15.3±0.8	12.7±1.0	12.9±0.8
	MLP	19.6±1.4	23.4±3.1	14.4±1.5	15.4±2.0	12.8±1.6	13.1±1.8
	GSAGE	27.1±0.4	31.9±0.6	17.1±0.3	18.3±0.4	14.0±0.2	14.3±0.3
	CTGLP	<b>30.0±0.7</b>	<b>35.8±0.8</b>	<b>19.6±0.4</b>	<b>21.0±0.5</b>	<b>16.3±0.6</b>	<b>16.7±0.6</b>

**Table 2.** Performance of various methods on datasets *with* unseen nodes. Items with the highest values are marked in **bold**.

features of neighbors) is helpful to enhance the model performance. (3) Task-independent embedding-based methods are not suitable for group link prediction directly.

## 6 Conclusion

In this paper, to address the dynamic group link prediction problem which concentrates on the future relationships between individuals and groups in continuous-time dynamic settings, we propose a novel continuous-time group link prediction method CTGLP. We first build continuous-time interaction networks based on the historical interactions between individuals and groups and present a new graph neural network CTGNN to learn the node embeddings, where a novel aggregation function is designed to jointly capture network structural features and temporal information. Then, we provide an importance-based group modeling

	Method	HR@K(%)		NDCG@K(%)		MRR@K(%)	
		K=10	K=20	K=10	K=20	K=10	K=20
ML100K <sub>w/o</sub>	AA	8.8	18.7	4.3	6.8	3.1	3.8
	CN	7.7	18.7	3.5	6.3	2.2	3.0
	DW	0.0	2.0	0.0	0.5	0.0	0.1
	n2v	0.0	4.0	0.0	1.0	0.0	0.3
	HTNE	0.0	8.0	0.0	2.0	0.0	0.5
	LSTM	4.7	7.0	2.0	2.6	1.2	1.3
	CVAE	<b>30.0±1.6</b>	<b>38.0±1.1</b>	17.7±1.1	19.6±1.1	13.8±1.3	14.3±1.3
	CVAEH	24.0±3.0	28.0±2.3	12.2±1.9	13.1±1.8	8.4±1.9	8.7±1.9
	CTGLP	28.0±1.0	34.0±0.6	<b>22.1±0.9</b>	<b>23.5±1.0</b>	<b>20.3±0.8</b>	<b>20.7±0.8</b>
CiaoDVD <sub>w/o</sub>	AA	20.4	28.7	12.6	14.8	10.3	10.9
	CN	19.3	30.1	12.1	14.8	9.9	10.6
	DW	0.9	1.9	0.3	0.6	0.2	0.3
	n2v	0.5	1.0	0.2	0.4	0.1	0.2
	HTNE	0.5	2.0	0.2	0.5	0.1	0.2
	LSTM	13.0	16.6	8.2	9.1	6.7	7.0
	CVAE	22.1±1.7	26.8±1.0	12.6±1.1	13.8±0.9	9.7±0.9	10.0±0.9
	CVAEH	22.5±0.9	27.7±1.0	13.1±0.7	14.4±0.6	10.2±0.6	10.5±0.6
	CTGLP	<b>25.5±0.9</b>	<b>30.7±1.0</b>	<b>17.0±0.5</b>	<b>18.3±0.4</b>	<b>14.3±0.9</b>	<b>14.7±1.0</b>
ML25M <sub>w/o</sub>	AA	42.5	45.1	31.8	32.4	28.5	<b>28.7</b>
	CN	42.7	45	<b>31.9</b>	<b>32.5</b>	<b>28.6</b>	<b>28.7</b>
	DW	2.1	5.4	0.7	1.5	0.3	0.5
	n2v	1.3	3.5	0.5	1.0	0.2	0.4
	HTNE	1.3	3.1	0.5	0.9	0.2	0.3
	LSTM	27.3	32.6	19.8	21.1	17.5	17.8
	CVAE	27.8±0.5	34.2±0.6	19.3±0.4	20.9±0.7	16.6±0.7	17.1±0.5
	CVAEH	27.2±0.4	32.1±0.7	19.3±0.5	20.6±0.4	16.9±0.4	17.2±0.4
	CTGLP	<b>45.8±1.1</b>	<b>54.4±0.9</b>	28.3±0.8	30.5±1.0	22.9±1.1	23.5±1.3

**Table 3.** Performance of various methods on datasets *without* unseen nodes. Items with the highest values are marked in **bold**.

function to model latent representations of groups, which can differentiate the contribution of members to group formation. Finally, the targets can be predicted using CTGLP based on group vectors. We compare CTGLP with ten baselines on various datasets and the experimental results show that CTGLP outperforms the state-of-the-art method. We also conduct a series of comprehensive experiments to analyze the effects of model components and hyperparameters on performance.

## Acknowledgments

The work was supported by STI 2030' Major Projects (2021ZD0201300) and the Concept Grant of Hangzhou Institute of Technology of Xidian University (No.GNYZ2023XJ0409-2).

## References

1. Adamic, L.A., Adar, E.: Friends and neighbors on the web. *Social Networks* **25**(3), 211–230 (2003)
2. Barracchia, E.P., Pio, G., Bifet, A., Gomes, H.M., Pfahringer, B., Ceci, M.: Lp-robin: link prediction in dynamic networks exploiting incremental node embedding. *Information Sciences* **606**, 702–721 (2022)
3. Chen, J., Wang, X., Xu, X.: GC-LSTM: Graph convolution embedded LSTM for dynamic network link prediction. *Applied Intelligence* **52**(7), 7513–7528 (2022)
4. Ghasemian, A., Hosseinmardi, H., Galstyan, A., Airoidi, E.M., Clauset, A.: Stacking models for nearly optimal link prediction in complex networks. *Proceedings of the National Academy of Sciences* **117**(38), 23393–23400 (2020)
5. Goyal, P., Chhetri, S.R., Canedo, A.: dyngraph2vec: Capturing network dynamics using dynamic graph representation learning. *Knowledge-Based Systems* **187**, 104816 (2020)
6. Grover, A., Leskovec, J.: node2vec: Scalable feature learning for networks. In: *SIGKDD 2016*. pp. 855–864 (2016)
7. Guo, G., Zhang, J., Thalmann, D., Yorke-Smith, N.: Etaf: An extended trust antecedents framework for trust prediction. In: *ASONAM 2014*. pp. 540–547. *IEEE* (2014)
8. Hamilton, W., Ying, Z., Leskovec, J.: Inductive representation learning on large graphs. *Advances in Neural Information Processing Systems* **30** (2017)
9. Harper, F.M., Konstan, J.A.: The movielens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems* **5**(4), 1–19 (2015)
10. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: *ICLR 2017*. *OpenReview.net* (2017), <https://openreview.net/forum?id=SJU4ayYgl>
11. Kumar, A., Singh, S.S., Singh, K., Biswas, B.: Link prediction techniques, applications, and performance: A survey. *Physica A: Statistical Mechanics and its Applications* **553**, 124289 (2020)
12. Liben-Nowell, D., Kleinberg, J.: The link-prediction problem for social networks. *Journal of the American Society for Information Science and Technology* **58**(7), 1019–1031 (2007)
13. Liu, G.: An ecommerce recommendation algorithm based on link prediction. *Alexandria Engineering Journal* **61**(1), 905–910 (2022)
14. Lü, L., Zhou, T.: Link prediction in complex networks: A survey. *Physica A: Statistical mechanics and its applications* **390**(6), 1150–1170 (2011)
15. Lv, L., Bardou, D., Hu, P., Liu, Y., Yu, G.: Graph regularized nonnegative matrix factorization for link prediction in directed temporal networks using pagerank centrality. *Chaos, Solitons and Fractals* **159**, 112107 (2022)
16. Mara, A.C., Lijffijt, J., De Bie, T.: Benchmarking network embedding models for link prediction: Are we making progress? In: *DSAA 2020*. pp. 138–147. *IEEE* (2020)

17. Martínez, V., Berzal, F., Cubero, J.C.: A survey of link prediction in complex networks. *ACM computing surveys (CSUR)* **49**(4), 1–33 (2016)
18. Nasiri, E., Berahmand, K., Rostami, M., Dabiri, M.: A novel link prediction algorithm for protein-protein interaction networks by attributed graph embedding. *Computers in Biology and Medicine* **137**, 104772 (2021)
19. Perozzi, B., Al-Rfou, R., Skiena, S.: Deepwalk: Online learning of social representations. In: *SIGKDD 2014*. pp. 701–710 (2014)
20. Pham, T.A.N., Li, X., Cong, G., Zhang, Z.: A general graph-based model for recommendation in event-based social networks. In: *ICDE 2015*. pp. 567–578. IEEE (2015)
21. Qu, L., Zhu, H., Duan, Q., Shi, Y.: Continuous-time link prediction via temporal dependent graph neural network. In: *WWW 2020*. pp. 3026–3032 (2020)
22. Rossi, A., Barbosa, D., Firmani, D., Matinata, A., Merialdo, P.: Knowledge graph embedding for link prediction: A comparative analysis. *ACM Transactions on Knowledge Discovery from Data (TKDD)* **15**(2), 1–49 (2021)
23. Sha, H., Al Hasan, M., Mohler, G.: Group link prediction using conditional variational autoencoder. In: *ICWSM 2021*. vol. 15, pp. 656–667 (2021)
24. Stanhope, A., Sha, H., Barman, D., Hasan, M.A., Mohler, G.: Group link prediction. In: *Big Data 2019*. pp. 3045–3052 (2019)
25. Taud, H., Mas, J.: Multilayer perceptron (MLP). In: *Geomatic Approaches for Modeling Land Change Scenarios*, pp. 451–455. Springer (2018)
26. Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., Sun, M.: Graph neural networks: A review of methods and applications. *AI Open* **1**, 57–81 (2020)
27. Zhou, T.: Progresses and challenges in link prediction. *Iscience* **24**(11), 103217 (2021)
28. Zuo, Y., Liu, G., Lin, H., Guo, J., Hu, X., Wu, J.: Embedding temporal network via neighborhood formation. In: *SIGKDD 2018*. pp. 2857–2866 (2018)