Dynamic Group Link Prediction in Continuous-Time Interaction Network

Shijie Luo, **He Li*** and **Jianbin Huang**
Xidian University
sjluo@stu.xidian.edu.cn, {heli, jbhuang}@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.

1 Introduction

Link prediction, aiming to predict relationships between pairs of entities, has received wide attention as the increasing importance of network data [Lü and Zhou, 2011; Martínez *et al.*, 2016; Kumar *et al.*, 2020]. 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 [Rossi *et al.*, 2021], biochemical reaction reconstruction [Nasiri *et al.*, 2021] and content recommendation [Liu, 2022]. Almost all existing link prediction methods focus only on relationships between pairs of entities [Zhou, 2021], 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 [Stanhope *et al.*, 2019; Sha *et al.*, 2021].

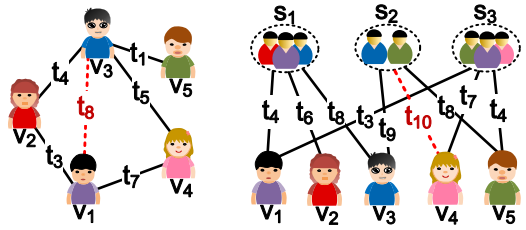


Figure 1: Difference between continuous-time link prediction and continuous-time group link prediction. (1) As shown in left part, in continuous-time link prediction, we predict the probability of an edge existing between individual v_1 and v_3 at future time t_8. (2) As shown in right part, in continuous-time group link prediction, we infer the possibility of generating a link between individual v_4 and group s_2 at future time t_{10}.

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*. The difference between continuous-time group link prediction and continuous-time link prediction is shown in Figure 1. 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

*The Corresponding Author

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.

- 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.
- 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 CTAgg is presented to bias the aggregation weights of the features of sampled neighbors.
- 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.
- 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

Link prediction. Based on the network structural similarity, heuristic link prediction methods, such as Common Neighbors (CN) [Liben-Nowell and Kleinberg, 2007] and Adamic-Adar (AA) [Adamic and Adar, 2003], 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 [Perozzi *et al.*, 2014], node2vec [Grover and Leskovec, 2016], HTNE [Zuo *et al.*, 2018], etc., calculate the possibility of generating links between nodes using the embeddings. Nevertheless, some studies [Mara *et al.*, 2020; Ghasemian *et al.*, 2020] 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 [Qu *et al.*, 2020] leverages temporal information in dynamic networks to achieve continuous-time link prediction. GNMFCA [Lv *et al.*, 2022] predicts future links using global and local information of temporal networks. GC-LSTM [Chen *et al.*, 2022] applies an embedded Long Short-Term Memory (LSTM) of Graph Convolutional Network to perform dynamic link prediction. Dyngraph2vec [Goyal *et al.*, 2020] integrates longer-term temporal information to learn node embeddings and predict future links. LP-ROBIN [Barracchia *et al.*, 2022] 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.

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 [Stanhope *et al.*, 2019] 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 [Sha *et al.*, 2021] 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 [Sha *et al.*, 2021] 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., $s_i = \{v_{i,1}^{t_1}, v_{i,2}^{t_2}, \ldots, v_{i,k}^{t_k}\} \subseteq V$, 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 $V = \{v_1, v_2, \ldots, v_N\}$ with N nodes and a group set $S = \{s_1, s_2, \ldots, s_M\}$ with M groups. For a group $s_i = \{v_{i,1}^{t_1}, v_{i,2}^{t_2}, \ldots, v_{i,k}^{t_k}\} \subseteq V$, $max(t_1, t_2, \ldots, t_k) \leq t$ with k members observed at current time t, the purpose of*

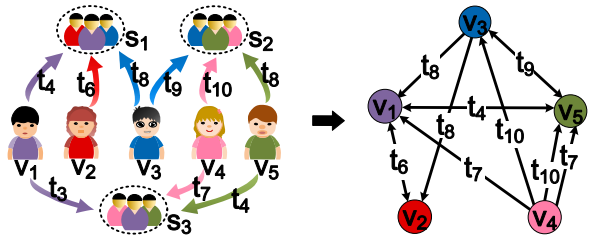


Figure 2: A toy example of the construction of a continuous-time interaction network.

continuous-time group link prediction is to predict the target $v_{i,k+1}^{t'} \in V \backslash s_i$ *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:*

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

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

3.2 Construction of Continuous-Time Interaction Network

The historical relationships between individuals and groups can be represented as a continuous-time dynamic network by decomposing the links between external individuals and groups into multiple links between the external individuals and group members. Figure 2 shows how to build a continuous-time interaction network. If individual v links to group s at time t, individual v directly connects to the existing members of group s and the timestamps of the edges are all t, as v is the initiator of the link action. For example, user v_4 links to group s_3 containing members v_1 and v_5 at time t_7, so two directed edges with timestamps of t_7 from v_4 to v_1 and v_5 are generated. Specially, if individuals v_i and v_j are the earliest two members in the group and their appearance times are t_i and t_j $(t_i < t_j)$, a bidirectional edge with a timestamp t_j will be added between them, as the appearance of the first two members indicates the formal formation of a group, and we consider the earliest two members to be visible to each other. For example, a bidirectional edge with a timestamp t_4 exists between v_1 and v_5. Note that multiple edges exist between two individuals when they link to multiple same groups. For example, there are two edges between v_4 and v_5.

4 Methodology

Figure 3 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 $\mathbf{X} = \{\vec{\mathbf{x}}_1, \vec{\mathbf{x}}_2, \ldots, \vec{\mathbf{x}}_N\}, \vec{\mathbf{x}}_i \in \mathbb{R}^D$, individual representation learning part obtains the final latent embeddings of members $\mathbf{Z} = \{\vec{\mathbf{z}}_1, \vec{\mathbf{z}}_2, \ldots, \vec{\mathbf{z}}_N\}, \vec{\mathbf{z}}_i \in \mathbb{R}^d$ 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 [Zhou *et al.*, 2020], 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 prior to the current time during aggregation. We first define the time-limited neighbor set $\Gamma_{\mathcal{T}}(u)$ of node u at time $\mathcal{T}$:

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

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:

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

where $r_\theta(\cdot)$ is the random sampling operation. θ is the neighbor sampling size and θ 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:

$$\hat{\mathcal{N}}_{\mathcal{T}}^l(u) = Samp_l\left(\Gamma_{\mathcal{T}_l}^l(\ldots Samp_1(\Gamma_{\mathcal{T}_1}^1(u)))\right), \tag{4}$$

where $Samp_l$ represents the sampling operation in the l-th convolutional layer. $\mathcal{T}_{i+1} < \mathcal{T}_i$ for $1 \le i < l$, 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 [Kipf and Welling, 2017; Hamilton *et al.*, 2017], 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 [Pham *et al.*, 2015], 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.

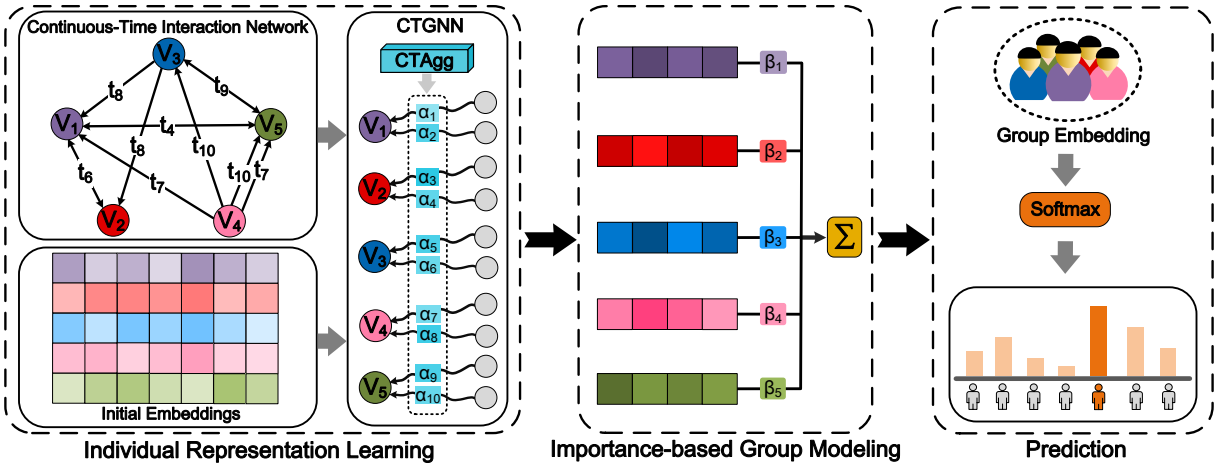


Figure 3: The overall framework of CTGLP. CTGLP is composed of three main components: individual representation learning, importance-based group modeling and prediction.

We provide an aggregation coefficient α 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 α_{ij}^t of node u_j should be greater than the aggregation coefficient α_{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:

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

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

where $\mathrm{AGG}(\cdot)$ is a function that aggregates the information of sampled neighbors and $\mathrm{COM}(\cdot)$ is a function that combines information about sampled neighborhoods and the pre-update information of the central node in the previous layer. σ 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. α_{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 α_{uv}^t is defined as:

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

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

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

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

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 [Stanhope *et al.*, 2019] 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 β 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:

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

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

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

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

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

Algorithm 1 Training of CTGLP

Input: Continuous-time interaction network $G = (V, E^T, T)$; set of node initial embeddings $\{\vec{\mathbf{x}}_v, \forall v \in V\}$; set of groups $S = (s_1, s_2, \ldots, s_M), s_i \subset V$.
Parameter: Convolutional layer depth K; neighbor sampling size $\theta_k, \forall k \in \{1, \ldots, K\}$; learnable parameters.
Output: Continuous-time group link prediction function $\mathcal{F}$.

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

latent vector $\vec{\mathbf{m}}_i$ of group s_i, the prediction process is:

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

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

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.

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:

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

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) [Harper and Konstan, 2015] and MovieLens-25M (ML25M for short) [Harper and Konstan, 2015] contains rating data from users on movies. CiaoDVD [Guo *et al.*, 2014] 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
$\text{ML100K}_{\text{w/o}}$	650	22 683	510	0
$\text{CiaoDVD}_{\text{w/o}}$	5 766	85 562	3 341	0
$\text{ML25M}_{\text{w/o}}$	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) [Stanhope *et al.*, 2019] employs LSTM to combine the historical information of groups and outputs link probabilities. **CVAE-based model** (CVAE for short) [Sha *et al.*, 2021] uses CVAE to reconstruct the membership of groups using a vector encoding an entire group and a vector encoding known members. **CVAEH** [Sha *et al.*, 2021] additionally introduces a vector encoding historical information of previous groups for prediction. (2) Two neural network-based methods. **MLP** [Taud and Mas, 2018] utilizes mini-batch gradient descent strategy to update model parameters. **GraphSAGE** (GSAGE for short) [Hamilton *et al.*, 2017] uses multi-layer aggregation functions to learn representations of nodes and obtains group vectors to make prediction. (3) Two heuristic link prediction methods. **AA** [Adamic and Adar, 2003] utilizes correlation coefficients of overlapping neighborhoods between nodes to measure the similarity between nodes. **CN** [Liben-Nowell and Kleinberg, 2007] 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) [Perozzi *et al.*, 2014] uses random walks to generate vectors of nodes. **node2vec** (n2v for short) [Grover and Leskovec, 2016] learns node representations using biased random walks. **HTNE** [Zuo *et al.*, 2018] 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.

	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	**30.5±0.9**	**42.4±0.5**	**21.5±0.9**	**24.4±0.8**	**18.6±0.7**	**19.4±0.7**
CiaoDVD	LSTM	10.6	14.7	6.0	7.0	4.5	4.8
	CVAE	**21.6±0.7**	27.1±0.8	**11.9±0.5**	13.4±0.3	**9.0±0.4**	**9.4±0.3**
	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	**28.7±0.7**	11.7±0.4	**13.7±0.2**	8.8±0.7	**9.4±0.8**
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	**30.0±0.7**	**35.8±0.8**	**19.6±0.4**	**21.0±0.5**	**16.3±0.6**	**16.7±0.6**

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

	Method	HR@K(%)		NDCG@K(%)		MRR@K(%)	
		K=10	K=20	K=10	K=20	K=10	K=20
$\text{ML100K}_{\text{w/o}}$	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	**30.0±1.6**	**38.0±1.1**	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	**22.1±0.9**	**23.5±1.0**	**20.3±0.8**	**20.7±0.8**
$\text{CiaoDVD}_{\text{w/o}}$	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	**25.5±0.9**	**30.7±1.0**	**17.0±0.5**	**18.3±0.4**	**14.3±0.9**	**14.7±1.0**
$\text{ML25M}_{\text{w/o}}$	AA	42.5	45.1	31.8	32.4	28.5	**28.7**
	CN	42.7	45	**31.9**	**32.5**	**28.6**	**28.7**
	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	**45.8±1.1**	**54.4±0.9**	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**.

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

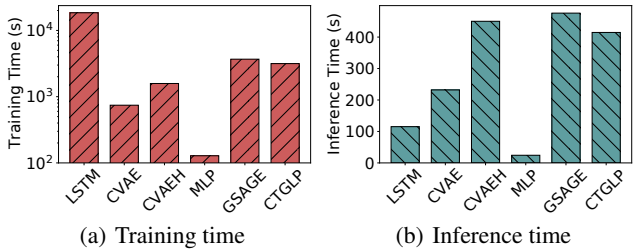
Performance on datasets *with* unseen nodes. From the overall results in Table 2, we observe that CTGLP outperforms most of the competing methods by a comfortable margin. On ML100K, our method obtains better performance than all baselines, especially in ranking targets. Specifically, CTGLP achieves average 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 $\text{ML100K}_{\text{w/o}}$, 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 $\text{CiaoDVD}_{\text{w/o}}$, 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 $\text{ML25M}_{\text{w/o}}$, 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 features of neighbors) is helpful to enhance the model performance. (3) Task-independent embedding-based methods are not suitable for group link prediction directly.

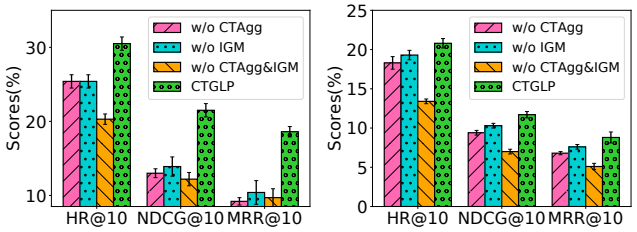
5.3 Comparison of Training and Inference Time

As shown in Figure 4, we can see that the training of our method is faster than LSTM and GSAGE and the inference is faster than CVAEH and GSAGE. MLP is always the fastest due to its simple architecture, while LSTM takes the longest time to train.



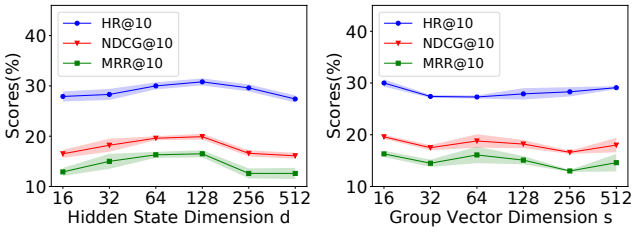
(a) Training time (b) Inference time

Figure 4: Comparison of training time and inference time of six methods on ML25M.



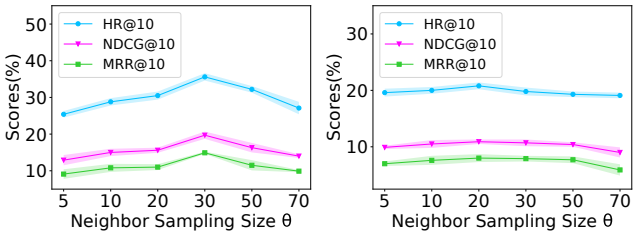
(a) On ML100K (b) On CiaoDVD

Figure 5: Impact of various components on performance under ML100K and CiaoDVD. IGM denotes the importance-based group modeling.



(a) Hidden state (b) Group vector

Figure 6: Impact of embedding dimension on method performance under ML25M.



(a) On ML100K (b) On CiaoDVD

Figure 7: Impact of embedding dimension on method performance on datasets ML100K and CiaoDVD.

5.4 Ablation Study

The ablation study results of various components on the performance of our method are shown in Figure 5. From the overall results, we observe that: (1) Pure CTGLP performs much better than its three variants in terms of all metrics. (2) The performance of CTGLP w/o CTAgg is similar to that of CTGLP w/o IGM. (3) CTGLP w/o CTAgg&IGM obtains the worst performance and is far inferior to the other variants, especially in seeking out targets. The results verify our idea that, depending on the time, different neighbors contribute differently to the feature updates of the central nodes and different members have varying importance within the group.

5.5 Hyper-parameter Study

Effect of Embedding Dimension. The experimental results of the effect of embedding dimension are shown in Figure 6. Overall, the performance of our proposed CTGLP in three metrics increases with the increasing embedding size, but the performance deteriorates after reaching a dimension bottleneck. In particular, CTGLP makes the best performance when the hidden state size d is 128 and the group vector size s is 16. The changes in embedding sizes have only a slight effect on performance, which shows the robustness of our method. Intuitively, using a larger embedding size enhances the vector representation, but it is not always optimal and increases model complexity. Therefore, we need to choose an appropriate size to trade off performance and complexity.

Effect of Neighbor Sampling Size. The experimental results of the effect of neighbor sampling size are shown in Figure 7. In general, the performance of our method first improves and then decreases as the growing of neighbor sampling size. CTGLP achieves the best performance when size θ is 30 on the small dataset ML100K, and it performs best on CiaoDVD when size θ is 20. To sum up, a small size may make the neighborhood information insufficient to capture link preference and an overly large size may introduce information noise since not all neighbors are positive. The larger the sampling size, the greater the memory usage and time consumption. To keep the balance between performance and complexity, we need to use a proper sampling size.

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 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

[Adamic and Adar, 2003] Lada A Adamic and Eytan Adar. Friends and neighbors on the web. *Social Networks*, 25(3):211–230, 2003.

[Barracchia *et al.*, 2022] Emanuele Pio Barracchia, Gianvito Pio, Albert Bifet, Heitor Murilo Gomes, Bernhard Pfahringer, and Michelangelo Ceci. Lp-robin: link prediction in dynamic networks exploiting incremental node embedding. *Information Sciences*, 606:702–721, 2022.

[Chen *et al.*, 2022] Jinyin Chen, Xueke Wang, and Xuanheng Xu. GC-LSTM: Graph convolution embedded LSTM for dynamic network link prediction. *Applied Intelligence*, 52(7):7513–7528, 2022.

[Ghasemian *et al.*, 2020] Amir Ghasemian, Homa Hosseinmardi, Aram Galstyan, Edoardo M Airoldi, and Aaron Clauset. Stacking models for nearly optimal link prediction in complex networks. *Proceedings of the National Academy of Sciences*, 117(38):23393–23400, 2020.

[Goyal *et al.*, 2020] Palash Goyal, Sujit Rokka Chhetri, and Arquimedes Canedo. dyngraph2vec: Capturing network dynamics using dynamic graph representation learning. *Knowledge-Based Systems*, 187:104816, 2020.

[Grover and Leskovec, 2016] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *SIGKDD 2016*, pages 855–864, 2016.

[Guo *et al.*, 2014] Guibing Guo, Jie Zhang, Daniel Thalmann, and Neil Yorke-Smith. Etaf: An extended trust antecedents framework for trust prediction. In *ASONAM 2014*, pages 540–547. IEEE, 2014.

[Hamilton *et al.*, 2017] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in Neural Information Processing Systems*, 30, 2017.

[Harper and Konstan, 2015] F Maxwell Harper and Joseph A Konstan. The movielens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems*, 5(4):1–19, 2015.

[Kipf and Welling, 2017] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR 2017*. OpenReview.net, 2017.

[Kumar *et al.*, 2020] Ajay Kumar, Shashank Sheshar Singh, Kuldeep Singh, and Bhaskar Biswas. Link prediction techniques, applications, and performance: A survey. *Physica A: Statistical Mechanics and its Applications*, 553:124289, 2020.

[Liben-Nowell and Kleinberg, 2007] David Liben-Nowell and Jon Kleinberg. The link-prediction problem for social networks. *Journal of the American Society for Information Science and Technology*, 58(7):1019–1031, 2007.

[Liu, 2022] Guoguang Liu. An ecommerce recommendation algorithm based on link prediction. *Alexandria Engineering Journal*, 61(1):905–910, 2022.

[Lü and Zhou, 2011] Linyuan Lü and Tao Zhou. Link prediction in complex networks: A survey. *Physica A: statistical mechanics and its applications*, 390(6):1150–1170, 2011.

[Lv *et al.*, 2022] Laishui Lv, Dalal Bardou, Peng Hu, Yanqiu Liu, and Gaohang Yu. Graph regularized nonnegative matrix factorization for link prediction in directed temporal networks using pagerank centrality. *Chaos, Solitons and Fractals*, 159:112107, 2022.

[Mara *et al.*, 2020] Alexandru Cristian Mara, Jefrey Lijffijt, and Tijl De Bie. Benchmarking network embedding models for link prediction: Are we making progress? In *DSAA 2020*, pages 138–147. IEEE, 2020.

[Martínez *et al.*, 2016] Víctor Martínez, Fernando Berzal, and Juan-Carlos Cubero. A survey of link prediction in complex networks. *ACM computing surveys (CSUR)*, 49(4):1–33, 2016.

[Nasiri *et al.*, 2021] Elahe Nasiri, Kamal Berahmand, Mehrdad Rostami, and Mohammad Dabiri. A novel link prediction algorithm for protein-protein interaction networks by attributed graph embedding. *Computers in Biology and Medicine*, 137:104772, 2021.

[Perozzi *et al.*, 2014] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *SIGKDD 2014*, pages 701–710, 2014.

[Pham *et al.*, 2015] Tuan-Anh Nguyen Pham, Xutao Li, Gao Cong, and Zhenjie Zhang. A general graph-based model for recommendation in event-based social networks. In *ICDE 2015*, pages 567–578. IEEE, 2015.

[Qu *et al.*, 2020] Liang Qu, Huaisheng Zhu, Qiqi Duan, and Yuhui Shi. Continuous-time link prediction via temporal dependent graph neural network. In *WWW 2020*, pages 3026–3032, 2020.

[Rossi *et al.*, 2021] Andrea Rossi, Denilson Barbosa, Donatella Firmani, Antonio Matinata, and Paolo Merialdo. Knowledge graph embedding for link prediction: A comparative analysis. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 15(2):1–49, 2021.

[Sha *et al.*, 2021] Hao Sha, Mohammad Al Hasan, and George Mohler. Group link prediction using conditional variational autoencoder. In *ICWSM 2021*, volume 15, pages 656–667, 2021.

[Stanhope *et al.*, 2019] Andrew Stanhope, Hao Sha, Danielle Barman, Mohammad Al Hasan, and George Mohler. Group link prediction. In *Big Data 2019*, pages 3045–3052, 2019.

[Taud and Mas, 2018] Hind Taud and JF Mas. Multilayer perceptron (MLP). In *Geomatic Approaches for Modeling Land Change Scenarios*, pages 451–455. Springer, 2018.

[Zhou *et al.*, 2020] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. *AI Open*, 1:57–81, 2020.

[Zhou, 2021] Tao Zhou. Progresses and challenges in link prediction. *Iscience*, 24(11):103217, 2021.

[Zuo *et al.*, 2018] Yuan Zuo, Guannan Liu, Hao Lin, Jia Guo, Xiaoqian Hu, and Junjie Wu. Embedding temporal network via neighborhood formation. In *SIGKDD 2018*, pages 2857–2866, 2018.