2024 IEEE International Conference on Evolving and Adaptive Intelligent Systems (EAIS) | 979-8-3503-6623-5/24/$31.00 ©2024 IEEE | DOI: 10.1109/EAIS58494.2024.10570036

Correlation-enhanced Dynamic Graph Learning for Temporal Link Prediction

1st Junzhe Chen
School of Mathematics
Nanjing University of Aeronautics and Astronautics
Nanjing, China
ccjjzz@nuaa.edu.cn

2st Zhiqiang Pan†
Science and Technology on Information Systems Engineering Laboratory
National University of Defense Technology
Changsha, China
panzhiqiang@nudt.edu.cn

3st Honghui Chen
Science and Technology on Information Systems Engineering Laboratory
National University of Defense Technology
Changsha, China
chenhonghui@nudt.edu.cn

***Abstract*—Temporal networks represent the evolving complex systems by regarding the contained elements as nodes and their connections as edges, respectively, which are both time-varying. Link prediction on temporal networks is an essential problem in real-world applications, which aims to forecast the evolution of temporal networks by predicting the future links to appear. However, existing methods generally focus on modeling the individual historical temporal features of source node and target node, while neglecting the complex correlations between them, thus leading to the suboptimal performance. In this paper, we propose a Correlation-enhanced Dynamic Graph learning (CoDyG) method to simultaneously take the individual features of source/target nodes and their correlations into consideration. Specifically, we achieve this by (1) introducing a co-attention network in the source/target node representation learning and (2) designing a temporal difference encoding strategy to model the temporal correlations between source/target nodes. Comprehensive experiments conducted on two widely adopted real-world temporal network datasets demonstrate that our proposed CoDyG can achieve the state-of-the-art performance in terms of the Average Precision (AP) and Area Under the Curve (AUC) metrics on the temporal link prediction task.**

***Index Terms*—temporal link prediction, dynamic graph learning, temporal network, co-attention mechanism**

I. Introduction

Graph networks are meaningful tools to model complex systems by regarding the contained elements as nodes and the interactions between nodes as edges, respectively [1]–[4]. Moreover, graph networks are usually time-varying in real-world applications, such as the social networks on social platforms [5]–[7] and the bipartite user-item graphs in recommender systems [8], [9]. Such time-varying characteristics of graph networks lead to temporal networks, in which the contained nodes and edges are continuously changing with time [10]–[14]. Though extensive researches have been conducted on static graph networks [15]–[22], the problems on temporal networks still requires further investigation. As an important and fundamental problem in forecasting the evolution of temporal networks, link prediction on temporal networks aims to predict the future edges, including (1) transductive link prediction forecasting the link formation probability between two historically seen nodes, like edge (v_3, v_4, t_4) in Fig. 1; and (2) inductive link prediction forecasting the link formation probability associated with at least one historically unseen node, such as edges (v_3, v_6, t_6) and (v_5, v_6, t_7) in Fig. 1. Considering the complex temporal characteristics reflected on temporal networks, temporal link prediction is a challenging problem to be investigated and solved.

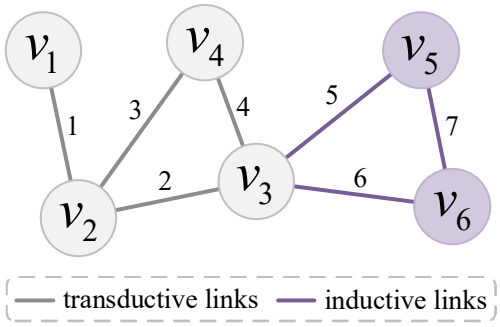


Fig. 1: An example of temporal network, where the continuous edge timestamps are marked using the numbers above edges.

Existing methods for link prediction on temporal networks mainly focus on learning the dynamic node representations of source node and target node for making future link prediction between them [23]–[28]. For example, recurrent neural networks (RNNs) are adopted to iteratively update the representations of nodes after each interaction in temporal networks [29], [30]. Moreover, Xu et al. design a temporal graph attention layer to learn dynamic node representations from the temporal and topological features and design a time encoding function for continuous time embedding [31]. Furthermore, Cong et al. design an effective and efficient method named GraphMixer to apply the multi-layer perceptron (MLP) Mixers [32] for learning from the neighbor sequence with adopting fixed time encoding function for embedding temporal features. In addition, Wang et al. propose to take the neural motifs on temporal networks into consideration by designing a causal anonymous walk method [33]–[36] for modeling the structural characteristics between source node and target node.

Although satisfactory results have been achieved, there still

†Corresponding author.

remain two challenging problems to be solved in existing researches. First, though the historical temporal features of source node and target node are well modeled in the previous dynamic graph learning methods, this is generally achieved in an individual way. That is, the dynamic correlations between source node and target node are not taken into consideration. In addition, as the pivotal features in dynamic graph learning, for both source node and target node, the interaction timestamps of historical neighbors are modeled by embedding their relative timespans between the current timestamp. However, the relative timespans between the historical neighbors of source node and those of target node, which can represent the different activity levels between them, are not considered.

In order to solve the above problems, we propose a Correlation-enhanced Dynamic Graph learning (CoDyG) method for link prediction on temporal networks. Specifically, given the source node and target node between which the temporal link is to predict, we first obtain the historical neighbors of source and target nodes by selecting their recent interacted neighbors, respectively. Then, we compute the relative timespans between the neighbors of source node and target node, which are then combined with the node features, the edge features and the neighbor timestamp features to generate the combined features for both source node and target node. After that, we apply a co-attention mechanism to dynamically select the information from the target node to inject into the source node, and vice versa, to fuse the information from source node and target node. Finally, we respectively input the features of source node and target node into a MLP-mixer to obtain their final dynamic representations, which are then combined and inputted to a MLP for generating the link formation probability.

To verify the effectiveness of our proposed CoDyG method, we compare the performance of CoDyG and the baselines on two widely adopted real-world temporal network datasets, i.e., Enron and UCI. Experimental results verify that CoDyG can outperform the competitive baselines in terms of the Average Precision (AP) and Area Under the Curve (AUC) metrics.

We summarize our contributions in this paper as follows:

1) We adopt the co-attention mechanism to integrate the information from source node and target node for modeling the dynamic correlations between them.
2) We design a temporal difference encoding strategy to consider the relative timespans between the neighbor interaction timestamps of source node and target node.
3) Extensive experiments conducted on two public datasets demonstrate the superiority of CoDyG over the competitive baselines in terms of AUC and AP.

The rest of this paper is organized as follows: We first formulate the investigated research task and detail our proposed CoDyG method in Section II. After that, we introduce the related experimental details such as the datasets, the included baselines, etc, in Section III. Next, we report our experimental results and further conduct deep analysis in Section IV. Finally, we summarize our work and point out the possible future research directions in Section V.

TABLE I: Main notations used in this paper, where $v_x \in \{v_s, v_t\}$ denotes the source node v_s or the target node v_t.

Notation	Description
$\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$	the temporal network with nodes and edges
$\mathcal{E}_T$	the tranductive temporal links to predict
$\mathcal{E}_I$	the inductive temporal links to predict
(v_s, v_t, t_{st})	the temporal link connecting v_s and v_t at the timestamp t_{st}
$\mathcal{N}_x$	the selected neighbors of v_x
$\mathbf{x}_i^x$	the node features of neighbor v_i^x
$\mathbf{e}_i^x$	the edge features brought by neighbor v_i^x
$\mathbf{t}_i^x$	the time features brought by neighbor v_i^x
$\mathbf{c}_i^x$	the combined features of neighbor v_i^x
$\mathbf{z}_i^x$	the final representation of neighbor v_i^x
$\mathbf{C}$	the affinity matrix between source/target nodes
$\mathbf{H}^s$	the neighbor representations of v_s fusing information from v_t
$\mathbf{H}^t$	the neighbor representations of v_t fusing information from v_s
$\mathbf{H}_{out}^x$	the neighbor representations of v_x outputted by MLP-Mixer
$\mathbf{o}^x$	the final dynamic node representation of node v_x
y_{st}	the predicted link formation probability between v_s and v_t
$\mathbf{F}_{time}$	the continuous time encoding function
$\mathbf{F}_{mean}$	the mean pooling operation function
K	the number of neighbors sampled for link prediction

II. Approach

We first give the formulation of the temporal link prediction task. Given the temporal network denoted as $\mathcal{G} = \{\mathcal{N}, \mathcal{E}\}$, where $\mathcal{N}$ and $\mathcal{E}$ are the contained time-varying nodes and edges in the temporal network. Each temporal link $(v_s, v_t, t_{st}) \in \mathcal{E}$ denotes that node v_s interacts with node v_t at the timestamp t_{st}. Link prediction on temporal networks aims to predict the future edges to appear in the temporal network, including transductive setting and inductive setting. Specifically, given the node set in the training set as $\mathcal{N}_{train}$, the transductive setting aims to predict the transductive temporal links connecting two nodes historically seen in the training set, i.e., $\mathcal{E}_T = \{(v_s, v_t, t_{st}) | v_s, v_t \in \mathcal{N}_{train}\}$; and the inductive setting aims to predict the inductive temporal links associated with at least one historically unseen node, i.e., $\mathcal{E}_I = \{(v_s, v_t, t_{st}) | v_s \notin \mathcal{N}_{train} || v_t \notin \mathcal{N}_{train}\}$. In addition, we summarize the main notations used in this paper in Table I.

Our proposed Correlation-enhanced Dynamic Graph learning (CoDyG) method mainly consists of three main components, i.e, node representation combination (see Section II-A), correlation enhanced learning (see Section II-B), prediction and optimization (see Section II-C). Specifically, given the source node and target node between which the future link is to predict, we first combine the node features, the edge features and the time features for both source node and target node individually. Then, to take the correlations between source/target nodes into consideration, we model the relative interaction timespans between the neighbors of source/target nodes using the designed temporal difference encoding strategy; and adopt a co-attention mechanism to dynamically fuse information from the neighbor sequence of the target node to that of the source node, and vice versa; then a two-layer MLP-Mixer is applied to summarize information from different neighbors for source node and target node, respectively. Finally, the mean pooling is utilized to combine information from different

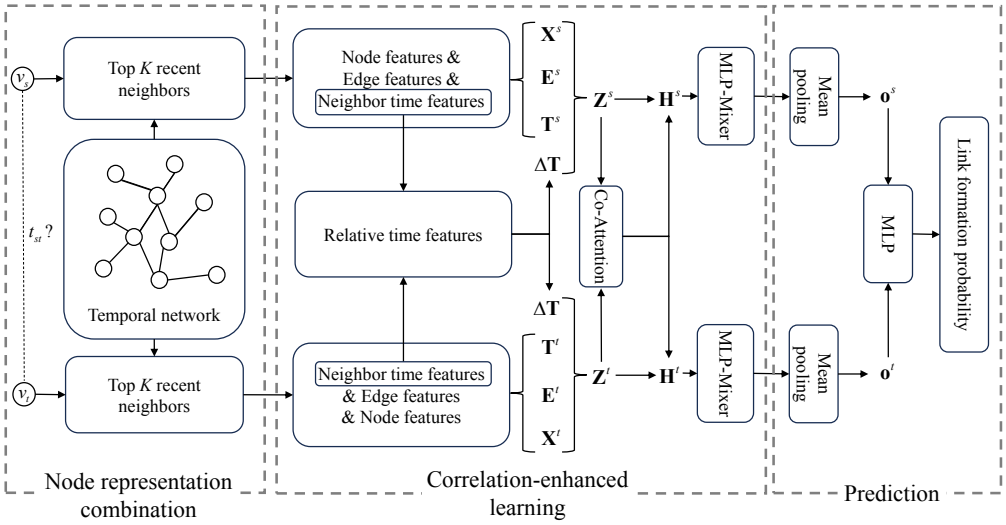


Fig. 2: Framework of our proposed Correlation-enhanced Dynamic Graph learning (CoDyG) method.

neighbors to generate the dynamic node representations for source node and target node, which are then adopted to generate the link formation probability.

A. Node representation combination

Given the source node v_s and target node v_t among which the future edge is to predict at the timestamp t_{st}, we first obtain the neighbor sequence for v_s and v_t, respectively. Here following [37], we adopt a simple and effective sampling strategy which directly takes the recent neighbors, i.e., the top K most recently interacted neighbors, as the neighbor sequence. We denote the neighbor sequence of source node and target node as $\mathcal{N}_s = \{v_1^s, v_2^s, \ldots, v_K^s\}$ and $\mathcal{N}_t = \{v_1^t, v_2^t, \ldots, v_K^t\}$, respectively. Then, taking the source node v_s as an example, for each neighbor $v_i^s \in \mathcal{N}_s$, we can directly obtain its node features as $\mathbf{x}_i^s \in \mathbb{R}^{d_N}$ and edge features as $\mathbf{e}_i^s \in \mathbb{R}^{d_E}$ from the temporal network, respectively. In addition, for the time encoding, we embedding the time interval between the current timestamp t_{st} and the interaction time t_i^s using the continuous time encoding function as follows:

$$\begin{aligned}\mathbf{t}_i^s &= \mathcal{F}_{time}(\Delta t_i^s) = \mathcal{F}_{time}(t_{st} - t_i^s) \\ &= [cos(\omega_1 \Delta t_i), sin(\omega_1 \Delta t_i), \ldots, cos(\omega_d \Delta t_i), sin(\omega_d \Delta t_i)],\end{aligned} \tag{1}$$

where $\mathbf{t}_i^s \in \mathbb{R}^{d_T}$ is the generating time encoding vector, $[\omega_1, \omega_2, \ldots, \omega_d]$ are the fixed parameters in the time encoding function $\mathcal{F}_{time}$, where $\mathcal{F}_{time}$ is defined by utilizing the random Fourier features [31], [38].

Then, we concatenate the node features, edge feature and time features of each neighbor v_i^s of source node v_s to obtain the combined features as $\mathbf{c}_i^s = [\mathbf{x}_i^s, \mathbf{e}_i^s, \mathbf{t}_i^s] \in \mathbb{R}^{d_N + d_E + d_T}$. Similarly, each neighbor $v_i^t \in \mathcal{N}_t$ of the target node can be represented as $\mathbf{c}_i^t = [\mathbf{x}_i^t, \mathbf{e}_i^t, \mathbf{t}_i^t] \in \mathbb{R}^{d_N + d_E + d_T}$.

B. Correlation enhanced learning

1) Temporal difference incorporation: After obtaining the individual representation of each neighbor of source node v_s and target node v_t, we take the correlations between v_s and v_t in to consideration. Specifically, we design a temporal difference encoding strategy to model the relative timespans between the interaction timestamps of the neighbors of v_s and v_t. Specifically, we calculate the timespans between the corresponding positions in the neighbor sequence of source node and target node as $\Delta\mathcal{T} = \{\Delta t_1, \Delta t_2, \ldots, \Delta t_K\}$, which are then embedded following the same way as in Eq. (1), generating the temporal difference embedding of the position i for both nodes v_s and v_t as $\Delta\mathbf{t}_i \in \mathbb{R}^{t_T}$.

Then, taking the source node as an example, we concatenate the temporal difference embeddings $\Delta\mathbf{t}_i \in \mathbb{R}^{t_T}$ to the combined representation $\mathbf{c}_i^s = [\mathbf{x}_i^s, \mathbf{e}_i^s, \mathbf{t}_i^s] \in \mathbb{R}^{d_N + d_E + d_T}$ of neighbor v_i^s, generating the final dynamic node representation of v_i^s as $\mathbf{z}_i^s = [\mathbf{x}_i^s, \mathbf{e}_i^s, \mathbf{t}_i^s, \Delta\mathbf{t}_i] \in \mathbb{R}^{d_N + d_E + d_T + d_T}$. Similarly, the final dynamic node representation of v_i^t can be generated as $\mathbf{z}_i^t = [\mathbf{x}_i^t, \mathbf{e}_i^t, \mathbf{t}_i^t, \Delta\mathbf{t}_i] \in \mathbb{R}^{d_N + d_E + d_T + d_T}$.

2) Co-attentive information fusion: After obtaining the representations of neighbors of the source node v_s as $\mathbf{Z}^s = \{\mathbf{z}_1^s, \mathbf{z}_2^s, \ldots, \mathbf{z}_K^s\}$ and those of the target node v_t as $\mathbf{Z}^t = \{\mathbf{z}_1^t, \mathbf{z}_2^t, \ldots, \mathbf{z}_K^t\}$, we propose to incorporate a co-attention mechanism to exploit the dynamic correlations between the final dynamic node representations of source node v_s and target node v_t.

Specifically, we first compute the affinity matrix $\mathbf{C}$ between $\mathbf{Z}^s$ and $\mathbf{Z}^t$ as follows:

$$\mathbf{C} = \tanh(\mathbf{Z}^s \mathbf{W}_c \mathbf{Z}^{t\top}), \tag{2}$$

where $\mathbf{C} \in \mathbf{R}^{K \times K}$ is the generated affinity matrix, and $\mathbf{W}_c \in \mathbf{R}^{d \times d}$ is the trainable matrix.

Then, we selectively incorporate information from the target neighbor sequence to the source neighbor sequence using the affinity matrix as follows:

$$\mathbf{H}^s = \text{Concat}[\mathbf{Z}^s, \text{Softmax}(\mathbf{C})\mathbf{Z}^t], \tag{3}$$

where $\mathbf{H}^s \in \mathbb{R}^{2K \times (d_N + d_E + d_T + d_T)}$ is the fused representations of source node v_s's neighbor sequence. Similarly, the fused

TABLE II: **Statistics of the preprocessed Enron and UCI datasets used in our experiments.**

Statistics	Nodes	Edges	Ave. Degree	Duration
Enron	184	125,235	680.63	3 years
UCI	1,899	59,835	31.51	196 days

representation of target node v_t's neighbor sequence can be generated as $\mathbf{H}^t \in \mathbb{R}^{2K\times(d_N+d_E+d_T+d_T)}$ as follows:

$$\mathbf{H}^t = \text{Concat}[\mathbf{Z}^t, \text{Softmax}(\mathbf{C}^\top)\mathbf{Z}^s]. \tag{4}$$

3) Information summation: After generating the fused representations of source node v_s's neighbors as $\mathbf{H}^s \in \mathbb{R}^{2K\times(d_N+d_E+d_T+d_T)}$ and those of target node v_t's neighbors as $\mathbf{H}^t \in \mathbb{R}^{2K\times(d_N+d_E+d_T+d_T)}$, we adopt a two-layer MLP-Mixer [32] to summarize the temporal information at different positions of $\mathbf{H}^s$ and $\mathbf{H}^t$, respectively. Specifically, each MLP-Mixer can be formulated as follows:

$$\begin{aligned}\mathbf{H}_{to} &= \mathbf{H}_{in} + \mathbf{W}^2_{to}\text{GeLU}(\mathbf{W}^1_{to}\text{LayerNorm}(\mathbf{H}_{in})),\\ \mathbf{H}_{ch} &= \mathbf{H}_{to} + \text{GeLU}(\text{LayerNorm}(\mathbf{H}_{to})\mathbf{W}^1_{ch})\mathbf{W}^2_{ch},\end{aligned} \tag{5}$$

where in, to, ch are short for input, token and channel, respectively. Here taking the source node as an example, $\mathbf{H}_{in}$ is transformed from $\mathbf{H}^s$ as $\mathbf{H}_{in} = \mathbf{W}_m\mathbf{H}^s$, and the representation $\mathbf{H}_{channel}$ at the second MLP-Mixer layer is deemed as the output of the two-layer MLP-Mixer, which can be denoted as $\mathbf{H}^s_{out}$. Similarly, the output for the target node v_t can be represented as $\mathbf{H}^t_{out}$.

C. Prediction and Optimization

After summarizing the information for source node v_s and target node v_t for obtaining the outputs as $\mathbf{H}^s_{out}$ and $\mathbf{H}^t_{out}$, respectively, we combine the information from different neighbors to generate the final dynamic node representation of v_s and v_t as follows:

$$\begin{aligned}\mathbf{o}^s &= \mathcal{F}_{mean}(\mathbf{H}^s_{out}),\\ \mathbf{o}^t &= \mathcal{F}_{mean}(\mathbf{H}^t_{out}),\end{aligned} \tag{6}$$

where $\mathbf{o}^s, \mathbf{o}^t \in \mathbb{R}^d$ are the generated dynamic node representations of source node v_s and target node v_t, respectively, $\mathcal{F}_{mean}$ denotes the mean pooling function.

Then, we combine them together, which are then inputted into a MLP to generate the link formation probability between v_s and v_t as follows:

$$y_{st} = \text{MLP}([\mathbf{o}^s, \mathbf{o}^t]) = \mathbf{W}_2(\text{ReLU}(\mathbf{W}_1[\mathbf{o}^s, \mathbf{o}^t]+\mathbf{b}_1))+\mathbf{b}_2, \tag{7}$$

where $\mathbf{W}_1 \in \mathbb{R}^{d\times 2d}$ and $\mathbf{W}_2 \in \mathbb{R}^{1\times d}$ are the learnable parameters, and $\mathbf{b}_1, \mathbf{b}_2 \in \mathbb{R}^d$ are the biased vectors.

Finally, the prediction score can be compared to the ground-truth label using the cross-entropy function as the optimization objective, which is then optimized by the Back-Propagation Through Time (BPTT) algorithm.

III. Experiments

A. Research Questions

We verify the effectiveness of our proposed CoDyG method by answering the following two research questions:

1) Can the proposal CoDyG beat the competitive baselines and achieve the state-of-the-art performance on the temporal link prediction task?
2) Hhow is the performance of CoDyG with different number of historical neighbors sampled for the dynamic node representation learning?

B. Datasets

In order to examine the effectiveness of our proposed CoDyG, we conduct experiments by comparing the performance of CoDyG and the baselines on two temporal network datasets, i.e., Enron and UCI. The details of two adopted datasets are as follows:

1) **Enron:** The Enron dataset comprises approximately 50,000 emails that were exchanged between employees of ENRON, an energy company, during a timespan lasting for three years.
2) **UCI:** The UCI dataset is a social network platform similar to Facebook, where students from the University of California at Irvine engage in online communication anonymously. Enron also includes timestamps with second-level temporal granularity.

Moreover, the temporal edges in both two temporal network datasets are chronologically separated as the training set, the validation set and the test set with the proportions of 70%, 15% and 15%, respectively. We summarize the statistics of two preprocessed datasets in Table II.

C. Model Summary

We select the following baselines for comparing the performance with CoDyG on the temporal link prediction task:

- **JODIE** [29] is tailored for temporal bipartite networks that involve interactions between users and items, which utilizes two coupled recurrent neural networks to update the status of users and items. Additionally, it incorporates a projection operation to forecast the future representation trajectory of each user or item.
- **DyRep** [30] is an RNN-based technique that continuously adjusts the node states following each interaction. Additionally, it features a temporal-attentive aggregation component to account for the changing structural information in dynamic graphs over time.
- **TGAT** [31] calculates the node representations by combining characteristics from the temporal-topological neighbors of each node using self-attention mechanism [39]. Moreover, it includes a time encoding function [40] to capture the temporal patterns effectively.
- **TGN** [41] utilizes an evolving memory for each node, which is updated whenever the node is involved in an interaction. This update process is facilitated by the message function, message aggregator, and memory updater.

TABLE III: Overall performance in terms of AP and AUC on transductive temporal link prediction, where the best and second best methods in each line are underlines and bold, respectively.

NSS	Datasets	Metrics	JODIE	DyRep	TGAT	TGN	TCL	GraphMixer	CoDyG
Random	Enron	AP	84.77 ± 0.30	82.38 ± 3.36	71.12 ± 0.97	86.53 ± 1.11	79.70 ± 0.71	82.25 ± 0.16	**87.67 ± 0.09**
		AUC	87.96 ± 0.52	84.89 ± 3.00	68.89 ± 1.10	**88.32 ± 0.99**	75.74 ± 0.72	84.38 ± 0.21	88.19 ± 0.20
	UCI	AP	89.43 ± 1.09	65.14 ± 2.30	79.63 ± 0.70	92.34 ± 1.04	89.57 ± 1.63	93.25 ± 0.57	**94.03 ± 0.25**
		AUC	90.44 ± 0.49	68.77 ± 2.34	78.53 ± 0.74	92.03 ± 1.13	87.82 ± 1.36	91.81 ± 0.67	**92.17 ± 0.27**
Historical	Enron	AP	69.85 ± 2.70	71.19 ± 2.76	64.07 ± 1.05	73.91 ± 1.76	70.66 ± 0.39	77.98 ± 0.92	**79.57 ± 1.42**
		AUC	75.39 ± 2.37	74.69 ± 3.55	61.85 ± 1.43	77.09 ± 2.22	67.95 ± 0.88	75.27 ± 1.14	**78.43 ± 2.39**
	UCI	AP	75.24 ± 5.80	55.10 ± 3.14	68.27 ± 1.37	80.43 ± 2.12	80.25 ± 2.74	84.11 ± 1.35	**88.18 ± 0.46**
		AUC	78.64 ± 3.50	57.91 ± 3.12	58.89 ± 1.57	77.25 ± 2.68	72.25 ± 3.46	77.54 ± 2.02	**83.71 ± 1.00**
Inductive	Enron	AP	68.96 ± 0.98	67.79 ± 1.53	63.94 ± 1.36	70.89 ± 2.72	71.29 ± 0.32	75.01 ± 0.79	**76.44 ± 1.10**
		AUC	70.92 ± 1.05	68.73 ± 1.34	60.45 ± 2.12	71.34 ± 2.46	67.64 ± 0.86	71.53 ± 0.85	**74.21 ± 1.74**
	UCI	AP	65.99 ± 1.40	54.79 ± 1.76	68.67 ± 0.84	70.94 ± 0.71	76.01 ± 1.11	80.10 ± 0.51	**81.54 ± 0.70**
		AUC	64.14 ± 1.26	54.25 ± 2.01	60.80 ± 1.01	64.11 ± 1.04	70.05 ± 1.86	74.59 ± 0.74	**77.26 ± 0.45**

TABLE IV: Overall performance in terms of AP and AUC on inductive temporal link prediction, where the best and second best methods in each line are underlines and bold, respectively.

NSS	Datasets	Metrics	JODIE	DyRep	TGAT	TGN	TCL	GraphMixer	CoDyG
Random	Enron	AP	80.72 ± 1.39	74.55 ± 3.95	67.05 ± 1.51	77.94 ± 1.02	76.14 ± 0.79	75.88 ± 0.48	**81.98 ± 0.33**
		AUC	81.96 ± 1.34	76.34 ± 4.20	64.63 ± 1.74	78.83 ± 1.11	72.33 ± 0.99	76.51 ± 0.71	**80.33 ± 0.57**
	UCI	AP	79.86 ± 1.48	57.48 ± 1.87	79.54 ± 0.48	88.12 ± 2.05	87.36 ± 2.03	91.19 ± 0.42	**92.54 ± 0.19**
		AUC	78.80 ± 0.94	58.08 ± 1.81	77.64 ± 0.38	86.68 ± 2.29	84.49 ± 1.82	89.30 ± 0.57	**90.54 ± 0.15**
Historical	Enron	AP	65.86 ± 3.71	62.08 ± 2.27	61.40 ± 1.31	62.91 ± 1.16	67.11 ± 0.62	72.37 ± 1.37	**74.48 ± 1.44**
		AUC	65.32 ± 3.57	61.50 ± 2.50	57.84 ± 2.18	62.68 ± 1.09	64.06 ± 1.02	68.20 ± 1.62	**70.12 ± 1.92**
	UCI	AP	63.11 ± 2.27	52.47 ± 2.06	70.52 ± 0.93	70.78 ± 0.78	76.71 ± 1.00	81.66 ± 0.49	**82.28 ± 0.78**
		AUC	60.24 ± 1.94	51.25 ± 2.37	62.32 ± 1.18	62.69 ± 0.90	70.46 ± 1.94	75.98 ± 0.84	**78.12 ± 0.40**
Inductive	Enron	AP	65.86 ± 3.71	62.08 ± 2.27	61.40 ± 1.30	62.90 ± 1.16	67.11 ± 0.62	72.37 ± 1.38	**74.48 ± 1.44**
		AUC	65.32 ± 3.57	61.50 ± 2.50	57.83 ± 2.18	62.68 ± 1.09	64.05 ± 1.02	68.19 ± 1.63	**70.12 ± 1.92**
	UCI	AP	63.16 ± 2.27	52.47 ± 2.09	70.49 ± 0.93	70.73 ± 0.79	76.65 ± 0.99	81.64 ± 0.49	**82.32 ± 0.78**
		AUC	60.27 ± 1.94	51.26 ± 2.40	62.29 ± 1.17	62.66 ± 0.91	70.42 ± 1.93	75.97 ± 0.85	**78.17 ± 0.39**

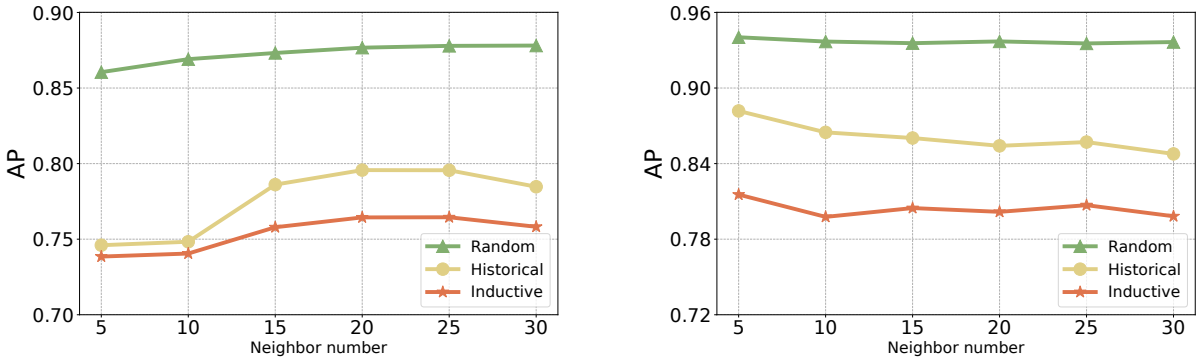
Additionally, TGN employs an embedding module to generate the temporal representations of nodes.

- **TCL** [42] utilizes a self-attention approach, beginning with the creation of each node's interaction sequence via a breadth-first search algorithm on the temporal dependency interaction sub-graph. Subsequently, TCL employs a graph transformer that takes into account both topological and temporal characteristics, and further integrates a cross-attention operation to model the interconnections between two interaction nodes.
- **GraphMixer** [37] is a straightforward MLP-based architecture that adopts a fixed time encoding function and integrates it into a MLP-Mixer based link encoder to extract features from temporal links, and further employs a node encoder for neighbor features summarization.

D. Experimental Setup

We compare the performance of CoDyG and the baselines on temporal link prediction with two settings: (1) transductive setting, which predicts the link formation probability between two historically seen nodes in the training set; (2) inductive setting, which predicts the link formation probability associated with at least one historically unseen node. In addition, we adopt Average Precision (AP) and Area Under the Curve (AUC) as the metrics for evaluating the temporal link prediction performance. In addition, we evaluate the model with three different negative sampling strategies, including (1) random, which selects the negative samples randomly from the full edges of the temporal network; (2) historical, which selects the negative samples from the historically observed edges in the full edges of the temporal network; (3) inductive, which selects the negative samples from the historically observed edges in the inductive edges of the temporal network.

Moreover, all models are trained with a maximum epoch number of 200 equipped with an early stopping strategy with a patience of 20, and the model achieving the best performance on the validation set is selected for evaluating the performance on the test set. And the Adam optimizer is adopted for optimizing the model parameters, with the batch size and the learning rate setting to $1e^{-4}$ and 200, respectively. In addition, for our proposed CoDyG, we set all the dimensions of the node embedding, edge embedding, neighbor time encoding and difference time encoding to 100, and the number of MLP-Mixer layers is set to 2. Furthermore, we range the number of sampled historical neighbors for source node and target node



(a) **Transductive performance on Enron.** (b) **Transductive performance on UCI.**

Fig. 3: Model performance of CoDyG with different number of neighbors sampled for dynamic node representation learning.

in $\{5, 10, 15, 20, 25, 30\}$ for both the Enron and UCI datasets.

IV. Results

A. Overall Performance

To verify the superiority of our proposal above the baselines, we compare the performance between CoDyG and the baselines in terms of AP and AUC on two real-world temporal network datasets, i.e., Enron and UCI. The experimental results in the transductive setting and inductive setting are presented in Table III and Table IV, respectively.

From Table III and Table IV, we can observe that CoDyG can achieve the best performance in almost all cases, validating the effectiveness of our proposed CoDyG on the temporal link prediction task. Moreover, among the baselines, we can see that GraphMixer generally outperforms the other baselines. We analyze this may be due to that the adopted fixed time encoding function and the MLP-Mixer based link encoder in GraphMixer can well model the temporal characteristics reflected by the neighbors of either source node or target node. However, it faces with a major issue, i.e., neglect the dynamic correlations between source node and target node, which also exists in the other baselines. Though solving this issue by introducing the co-attention mechanism and designing a temporal difference encoding strategy on the basis of GraphMixer, CoDyG largely improves the performance of GraphMixer and achieves the state-of-the-art performance.

B. Parameter Analysis

To examine the sensitivity of CoDyG to the neighbor number, i.e., analyzing the impact of the neighbor number on the performance of CoDyG, we range the neighbor number in $\{5, 10, 15, 20, 25, 30\}$ for testing the performance of CoDyG. We present the results on the transductive setting in terms of AP in Fig. 3, where the results in other cases are similar.

From Fig. 3a, we can observe that with the neighbor number increasing, the performance of CoDyG generally increases on both the transductive and inductive settings on the Enron dataset. However, from Fig. 3b, we can see that the phenomenon is opposite to that on Enron, i.e., the performance of CoDyG slightly decreases when the neighbor number increases. We analyze the difference may be caused by that the average degree on two datasets are different, which are respectively 680.63 and 31.51 as we can observe from Table. II. That is, the larger average degree in Enron leads to a stronger demand of long neighbor sequences, while a shorter neighbor sequence is enough for accurate dynamic representation learning on UCI.

V. Conclusion

In this paper, we focus on the temporal link prediction task, which aims to forecast the future edges to appear on temporal networks, including predicting transductive temporal links connecting two historically seen nodes and inductive temporal links associated with at least one historically unseen node. Existing methods mainly focus on modeling the individual temporal characteristics of source node and target node, while failing to adequately consider the complex correlations between them. Thus, we propose a Correlation enhance Dynamic Graph learning (CoDyG) method for the temporal link prediction task. Specifically, CoDyG adopts a co-attention mechanism to fuse the dynamic correlations between source node and target node, and designs a temporal difference encoding strategy to take the relative timespans between the neighbor interaction timestamps between source/target nodes. We conduct extensive experiments on two widely adopted real-world temporal network datasets, i.e., Enron and UCI. The experimental results demonstrate that our proposed CoDyG can achieve the state-of-the-art performance on predicting future links on temporal networks.

As to future work, we are interested in introducing the graph condensation technique [43], [44] to simplify large-scale temporal networks for achieving simultaneously effective and efficient temporal link prediction. In addition, we also would like to extend our proposal to make CoDyG scalable to extremely large industrial temporal networks [45].

Acknowledgment

This work was supported in part by the Independent Innovation Science Foundation Project of National University of Defense Technology under Grant 23-ZZCX-JDZ-43 and in part by the Postgraduate Scientific Research Innovation Project of Hunan Province under Grant CX20230083.

REFERENCES

[1] Z. Wu, S. Pan, F. Chen *et al.*, "A comprehensive survey on graph neural networks," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 32, no. 1, pp. 4–24, 2021.

[2] S. Khoshraftar and A. An, "A survey on graph representation learning methods," *arXiv preprint arXiv:2204.01855*, 2022.

[3] G. Du, J. Zhang, M. Jiang *et al.*, "Graph-based class-imbalance learning with label enhancement," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 34, no. 9, pp. 6081–6095, 2023.

[4] G. Du, J. Zhang, N. Zhang *et al.*, "Semi-supervised imbalanced multi-label classification with label propagation," *Pattern Recognit.*, vol. 150, p. 110358, 2024.

[5] Y. Wang, P. Li, C. Bai *et al.*, "TEDIC: neural modeling of behavioral patterns in dynamic social interaction networks," in *The Web Conference*. ACM / IW3C2, 2021, pp. 693–705.

[6] L. A. Adamic and E. Adar, "Friends and neighbors on the web," *Soc. Networks*, vol. 25, no. 3, pp. 211–230, 2003.

[7] S. Kumar, A. Mallik, A. Khetarpal, and B. S. Panda, "Influence maximization in social networks using graph embedding and graph neural network," *Inf. Sci.*, vol. 607, pp. 1617–1636, 2022.

[8] J. Wang, K. Ding, L. Hong *et al.*, "Next-item recommendation with sequential hypergraphs," in *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*. ACM, 2020.

[9] C. Wei, J. Liang, B. Bai, and D. Liu, "Dynamic hypergraph learning for collaborative filtering," in *Proceedings of the 31st ACM International Conference on Information and Knowledge Management*. ACM, 2022, pp. 2108–2117.

[10] S. M. Kazemi, R. Goel, K. Jain *et al.*, "Representation learning for dynamic graphs: A survey," *J. Mach. Learn. Res.*, vol. 21, pp. 70:1–70:73, 2020.

[11] C. D. T. de Barros, M. R. F. Mendonça, A. B. Vieira, and A. Ziviani, "A survey on embedding dynamic graphs," *ACM Comput. Surv.*, vol. 55, no. 2, pp. 10:1–10:37, 2023.

[12] A. Li, L. Zhou, Q. Su *et al.*, "Evolution of cooperation on temporal networks," *Nature communications*, vol. 11, no. 1, p. 2259, 2020.

[13] J. Wang, G. Song, Y. Wu *et al.*, "Streaming graph neural networks via continual learning," in *Proceedings of the 29th ACM International Conference on Information and Knowledge Management*. ACM, 2020, pp. 1515–1524.

[14] J. Li, H. Dani, X. Hu *et al.*, "Attributed network embedding for learning in a dynamic environment," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. ACM, 2017, pp. 387–396.

[15] Y. Li, D. Tarlow, M. Brockschmidt *et al.*, "Gated graph sequence neural networks," in *4th International Conference on Learning Representations*, 2016.

[16] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *5th International Conference on Learning Representations*, 2017.

[17] P. Velickovic, G. Cucurull, A. Casanova *et al.*, "Graph attention networks," in *6th International Conference on Learning Representations*, 2018.

[18] W. L. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Annual Conference on Neural Information Processing Systems*, 2017, pp. 1024–1034.

[19] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2016, pp. 855–864.

[20] R. Ying, R. He, K. Chen *et al.*, "Graph convolutional neural networks for web-scale recommender systems," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2018, pp. 974–983.

[21] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: online learning of social representations," in *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2014, pp. 701–710.

[22] Y. Hao, X. Cao, Y. Fang *et al.*, "Inductive link prediction for nodes having only attribute information," in *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*. ijcai.org, 2020, pp. 1209–1215.

[23] L. Zhou, Y. Yang, X. Ren *et al.*, "Dynamic network embedding by modeling triadic closure process," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*. AAAI Press, 2018, pp. 571–578.

[24] A. Pareja, G. Domeniconi, J. Chen *et al.*, "Evolvegcn: Evolving graph convolutional networks for dynamic graphs," in *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence*. AAAI Press, 2020, pp. 5363–5370.

[25] S. Tian, R. Wu, L. Shi *et al.*, "Self-supervised representation learning on dynamic graphs," in *Proceedings of the 30th ACM International Conference on Information and Knowledge Management*. ACM, 2021, pp. 1814–1823.

[26] U. Singer, I. Guy, and K. Radinsky, "Node embedding over temporal graphs," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*. ijcai.org, 2019, pp. 4605–4612.

[27] E. Hajiramezanali, A. Hasanzadeh, K. R. Narayanan *et al.*, "Variational graph recurrent neural networks," in *Annual Conference on Neural Information Processing Systems*, 2019, pp. 10 700–10 710.

[28] P. Goyal, S. R. Chhetri, and A. Canedo, "dyngraph2vec: Capturing network dynamics using dynamic graph representation learning," *Knowl. Based Syst.*, vol. 187, 2020.

[29] S. Kumar, X. Zhang, and J. Leskovec, "Predicting dynamic embedding trajectory in temporal interaction networks," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2019, pp. 1269–1278.

[30] R. Trivedi, M. Farajtabar, P. Biswal *et al.*, "Dyrep: Learning representations over dynamic graphs," in *7th International Conference on Learning Representations*, 2019.

[31] D. Xu, C. Ruan, E. Körpeoglu *et al.*, "Inductive representation learning on temporal graphs," in *8th International Conference on Learning Representations*, 2020.

[32] I. O. Tolstikhin, N. Houlsby, A. Kolesnikov *et al.*, "Mlp-mixer: An all-mlp architecture for vision," in *Annual Conference on Neural Information Processing Systems*, 2021, pp. 24 261–24 272.

[33] H. Yin, M. Zhang, Y. Wang *et al.*, "Algorithm and system co-design for efficient subgraph-based graph representation learning," *arXiv preprint arXiv:2202.13538*, 2022.

[34] Y. Liu, J. Ma, and P. Li, "Neural predicting higher-order patterns in temporal networks," in *The Web Conference*. ACM, 2022, pp. 1340–1351.

[35] M. Jin, Y. Li, and S. Pan, "Neural temporal walks: Motif-aware representation learning on continuous-time dynamic graphs," in *Annual Conference on Neural Information Processing Systems*, 2022.

[36] Y. Wang, Y. Chang, Y. Liu *et al.*, "Inductive representation learning in temporal networks via causal anonymous walks," in *9th International Conference on Learning Representations*, 2021.

[37] W. Cong, S. Zhang, J. Kang *et al.*, "Do we really need complicated model architectures for temporal networks?" in *The Eleventh International Conference on Learning Representations*, 2021.

[38] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," in *Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems*, 2007, pp. 1177–1184.

[39] A. Vaswani, N. Shazeer, N. Parmar *et al.*, "Attention is all you need," in *Annual Conference on Neural Information Processing Systems*, 2017, pp. 5998–6008.

[40] D. Xu, C. Ruan, E. Körpeoglu *et al.*, "Self-attention with functional time representation learning," in *Annual Conference on Neural Information Processing Systems*, 2019, pp. 15 889–15 899.

[41] E. Rossi, B. Chamberlain, F. Frasca *et al.*, "Temporal graph networks for deep learning on dynamic graphs," *arXiv preprint arXiv:2006.10637*, 2020.

[42] L. Wang, X. Chang, S. Li *et al.*, "TCL: transformer-based dynamic graph modelling via contrastive learning," *arXiv preprint arXiv:2105.07944*, 2021.

[43] X. Gao, T. Chen, Y. Zang *et al.*, "Graph condensation for inductive node representation learning," *arXiv preprint arXiv:2307.15967*, 2023.

[44] W. Jin, L. Zhao, S. Zhang *et al.*, "Graph condensation for graph neural networks," in *The Tenth International Conference on Learning Representations*. OpenReview.net, 2022.

[45] Y. Zheng, Z. Wei, and J. Liu, "Decoupled graph neural networks for large dynamic graphs," *Proc. VLDB Endow.*, vol. 16, no. 9, pp. 2239–2247, 2023.