Disentangling Node Metric Factors for Temporal Link Prediction

Tianli Zhang[1,2], Tongya Zheng[1,3,4], Yuanyu Wan[1,3,4(✉)], Ying Li[5], and Wenqi Huang[6]

[1] Zhejiang University, Hangzhou 310027, China
{zhangtianli,tyzheng,wanyy}@zju.edu.cn
[2] Zhejiang University - China Southern Power Grid Joint Research Centre on AI, Hangzhou 310058, China
[3] ZJU-Bangsun Joint Research Center, Hangzhou, China
[4] Shanghai Institute for Advanced Study of Zhejiang University, Hangzhou, China
[5] Zhejiang Bangsun Technology Co. Ltd., Hangzhou 310012, China
li_ying@bsfit.com.cn
[6] Digital Grid Research Institute, China Southern Power Grid, Guangzhou 510663, China
huangwq@csg.cn

Abstract. Temporal Link Prediction (TLP), as one of the highly concerned tasks in graph mining, requires predicting the future link probability based on historical interactions. On the one hand, traditional methods based on node metrics, such as Common Neighbor, achieve satisfactory performance in the TLP task. On the other hand, node metrics overly focus on the global impact of nodes while neglecting the personalization of different node pairs, which can sometimes mislead link prediction results. However, mainstream TLP methods follow the standard paradigm of learning node embedding, entangling favorable and harmful node metric factors in the representation, reducing the model's robustness. In this paper, we propose a plug-and-play plugin called **N**ode **M**etric **D**isentanglement, which can apply to most TLP methods and boost their performance. It explicitly accounts for node metrics and disentangles them from the embedding representations generated by TLP methods. We adopt the attention mechanism to reasonably select information conducive to the TLP task and integrate it into the node embedding. Experiments on various *state-of-the-art* methods and dynamic graphs verify the effectiveness and universality of our NMD plugin.

Keywords: Dynamic Graphs · Temporal Link Prediction · Disentangled Representation Learning · Node Metric

1 Introduction

Graph data, particularly dynamic graphs, have grown exponentially in recent years, arousing widespread attention from researchers and practitioners [25,27,28]. Dynamic graphs can represent the changeable interactions of nodes with

B. Luo et al. (Eds.): ICONIP 2023, LNCS 14448, pp. 346–357, 2024.
https://doi.org/10.1007/978-981-99-8082-6_27

time elapsing in real-graph scenarios, such as social networks [23], academic networks [31], transaction networks [16,17], etc. One of the most challenging issues related to dynamic graphs is temporal link prediction (TLP), which aims to estimate the likelihood of paired nodes being connected in the future [6]. It helps to track the dynamic characteristics and reveal the evolutionary patterns of the system. Some applications of TLP tasks include analyzing community clusters in social networks [5], revealing author collaboration trends in academic networks [15], and predicting commercial intercourse in transaction networks [18].

Traditional link prediction methods utilize the property metrics of nodes to predict links, such as similarity-based algorithms using degree or common neighbors [19]. Preferential Attachment [3] explicitly considers the degree centrality of nodes. Several heuristic algorithms like the Jaccard Coefficient [2], Adamic Adar Index [1], and Resource Allocation index [37] consider the common neighbors to predict a connection. On the one hand, node metrics reflect the global impact and promote link prediction in specific scenarios. For example, a new manuscript may prefer papers with more citations in their references. On the other hand, node metrics neglect the personalization of different node pairs, such as music application users preferring niche songs over popular songs on the list. It is crucial to handle node metrics information in link prediction tasks properly.

The interaction of many complex factors, including node metrics, usually drives temporal link changes in dynamic graph systems. Existing *state-of-the-art* methods, including GCN-GRU [29], EvolveGCN [25], DySAT [28], etc., predict temporal link existence by learning node embedding from the ego network based on various specific Graph Neural Networks. However, these TLP methods mix different factors into a shared embedding representation space, which may introduce harmful node metrics to temporal link prediction tasks. Numerous works in multiple fields [21,34] show that disentangling various information factors in embedding is conducive to better representation learning and model generalization. It inspires us to disentangle node metrics from embedding representations and integrate information conducive to link prediction into the embedding by a fusion module. In this paper, we aim to design a disentanglement plugin that can be applied to most TLP methods, disentangling node metric factors and improving their performance on different dynamic graphs.

To this end, we propose a dual-branch framework consisting of a temporal link prediction (TLP) branch and a node metric disentanglement (NMD) branch. The former describes a general structural-temporal framework for existing TLP methods, which can generate node representations based on historical graph snapshots. The latter uses multiple optional types of node metrics as factors, such as degree centrality, closeness centrality [8], PageRank [24], etc. We introduce a similarity decoupling loss to disentangle the node representations of different time slots under multiple metrics. It helps two branch focus on edge-level personalization and node-level global functionality, respectively. The TLP branch is optimized for an edge-level classification objective, and the NMD branch is optimized for a node-level regression objective. Finally, an attention-based fusion module is adopted for the two sets of historical embedding obtained from the

two branches, integrating information beneficial to the TLP task and exploring fine-grained representations. Our contributions are summarized as follows:

- To the best of our knowledge, we are the first to consider the positive and negative effects of node metric factors in representation learning on the temporal link prediction task.
- We propose a Node Metric Disentanglement (NMD) plugin that can disentangle node metrics in most TLP methods and generate fine-grained and more beneficial representations by the attention mechanism.
- Experiments on five dynamic graphs validate the universal improvement of our NMD plugin for existing SOTA methods. Elaborate ablation studies further verify the effectiveness of different components.

2 Related Work

Temporal Link Prediction. Some heuristics strategies based on graph topology are proposed to solve link prediction tasks, such as Katz [10], singular value decomposition (SVD) [7], and non-negative matrix factorization (NMF) [36]. Learning low-dimensional embedding on dynamic graphs is an emerging topic under investigation [4]. Dyngraph2Vec [9] uses recurrent neural networks (RNN) to learn each vertex's complex transformation. Inspired by the progress of Graph Neural Networks (GNNs) [12], EvolveGCN [25] trains RNN parameters at each time step to dynamically update GCN parameters. STGSN [22] introduces the attention mechanism to model social networks' spatial and temporal dynamicity.

Disentangled Representation Learning. [20] proposes a disentangled multichannel convolutional layer and devises a neighborhood routing mechanism to disentangle the underlying factors behind a graph. [33] establishes a set of intent-aware graphs and chunked representations, disentangling representations of users and items at the granularity of user intents. [21] proposes disentangled variational autoencoder and a beam-search strategy to explicitly models the separation of macro and micro factors. [34] utilizes a mechanism that incorporates the micro-and macro-disentanglement in knowledge graphs.

Node Metrics. Centrality is a classical measure of nodes in complex networks, including k-shell decomposition [13], closeness centrality [8], or betweenness centrality [8]. [32] uses the concept of *diversity entropy* to describe the relative frequency of access received by nodes and study the internal and external accessibility of nodes. [14] defines *dynamic influence* as the leading left eigenvector of a characteristic matrix that encodes the interaction between graph topology and dynamics. Adamic Adar index [1] is a similarity measure that assigns higher importance to common neighbors with lower degrees in a graph representation.

3 Method

3.1 Problem Statement

Most dynamic networks can be described as a weighted temporal graph $G(V, E)$ with a node set $V = \{v_i\}_{i=1}^{N}$ and an edge set $E \subseteq |V| \times |V|$, where $N = |V|$

is the number of unique vertices. An edge between nodes v_i and v_j with weight $w_{ij} \in \mathbf{R}$ at time step $t \in \mathbf{R}^+$ is written as $e_{ij} = (v_i, v_j, t, w_{ij}) \in E$. A graph G can be split into a series of snapshots $G = \{G_1, ..., G_T\}$. We transform E_t into an adjacency matrix $A_t \in \mathbf{R}^{N \times N}$ according to the rule: $(A_t)_{ij} = w_{ij}$ otherwise 0. The d-dimensional node feature matrix in G_t is defined as $X_t \in \mathbf{R}^{N \times d}$.

Temporal **L**ink **P**rediction (TLP) task predicts future link status based on historical snapshots. Formally, given δ snapshots $\{A_\tau, X_\tau\}_{\tau=t-\delta}^{t-1}$ at time t, our goal is to learn a function $f(\cdot)$ that predicts the adjacency matrix A_t:

$$\{A_\tau, X_\tau\}_{\tau=t-\delta}^{t-1} \xrightarrow{f(\cdot)} A_t, t \in \{1+\delta, ..., T\} \tag{1}$$

3.2 Temporal Link Prediction Branch

The general paradigm of TLP methods is a sequential combination of a multi-layer structural encoder, a temporal encoder, and a link predictor [27]. For each historical snapshot G_τ at time t, an l-th layer of the structural encoder takes the adjacency matrix A_τ and the $(l-1)$-th output embedding Z_τ^{l-1} as input:

$$Z_\tau^l = \text{StructuralEncoder}^l(A_\tau, Z_\tau^{l-1}), l = 1, ..., L, \tag{2}$$

where L is the number of layers, and the initial embedding matrix comes from the raw node features, *i.e.* $Z_\tau^0 = X_\tau$. Then a temporal encoder combines the L-th layer embedding of δ historical snapshots $\{Z_\tau^L\}_{\tau=t-\delta}^{t-1}$ and generates the current snapshot's node embedding Z_t for downstream tasks:

$$Z_t = \text{TemporalEncoder}(Z_{t-1}^L, ..., Z_{t-\delta}^L). \tag{3}$$

The TLP task can be regarded as a binary classification of positive and negative edges. As for an edge e_{ij} with its label $y = (A_t)_{ij} \in \{0, 1\}$, we use the following classification layer to obtain the prediction probability $p_{ij} \in [0, 1]$:

$$p_{ij}^t = \sigma\left(\text{MLP}\left(\mathbf{z}_i^t || \mathbf{z}_j^t\right)\right), \tag{4}$$

where σ is the softmax function, MLP is a multi-layer perceptron (MLP), and $||$ represents the concatenation operation. $\mathbf{z}_i^t = (Z_t)_i \in \mathbf{R}^{d_1}$ is the d_1-dimensional embedding of node v_i, *i.e.*, the i-th row of the embedding matrix Z_t. The TLP branch calculates the link prediction loss at t time by the Cross-Entropy (CE):

$$\mathcal{L}_{TLP}^t = K \sum_{e_{ij} \in E_t^+} -\log p_{ij}^t - \sum_{e_{ij} \in E_t^-} 1 - \log p_{ij}^t, \tag{5}$$

where E_t^+ and E_t^- are the set of positive and negative edges, respectively. K balances the loss of E_t^+ and E_t^-, usually set to the ratio of $|E_t^+|$ and $|E_t^-|$.

3.3 Node Metric Disentanglement Branch

As shown in Fig. 1, the proposed node metric disentanglement (NMD) branch includes a structural encoder with the same architecture as the TLP branch,

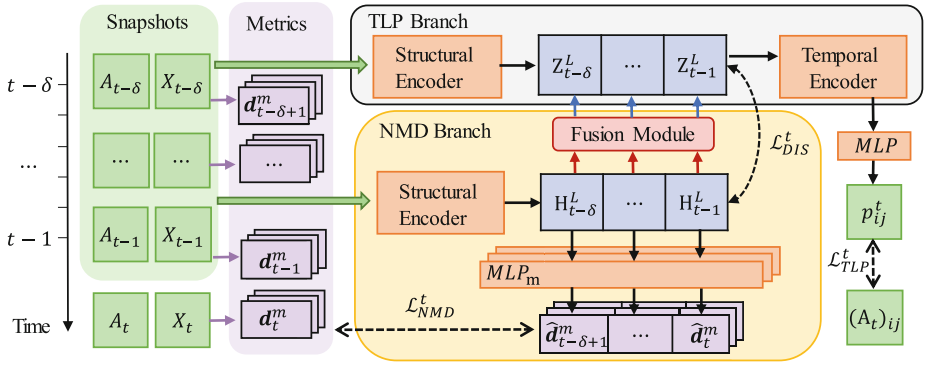

Fig. 1. Framework of our NMD plugin based on the structural-temporal paradigm. TLP Branch, applicable to any TLP method, predicts the link status at time t based on δ snapshots $\{A_\tau, X_\tau\}_{\tau=t-\delta}^{t-1}$. NMD Branch generates node-level embedding $\{H_\tau^L\}_{\tau=t-\delta}^{t-1}$ by predicting temporal node metrics $\{\mathbf{d}_\tau^m\}_{\tau=t-\delta}^{t}$. It disentangles the node representations of different time slots by the decoupling loss $\mathcal{L}_{DIS}^t$ and integrates beneficial factors into $\{Z_\tau^L\}_{\tau=t-\delta}^{t-1}$ by an attention-based fusion module.

multiple MLP predictors, and an attention-based fusion module. It uses multiple self-selectable node metrics as decoupling factors, such as degree centrality, closeness centrality [8], PageRank [24], etc.

Formally, given a node metric $m \in \mathcal{M}$, $\mathcal{M}$ is a set of various node metrics selected according to different contexts. We calculate the node metric vector $\mathbf{d}_\tau^m \in \mathbf{R}^{|V|}$ for each historical snapshot at time t and obtain the set of node metrics $\{\mathbf{d}_\tau^m\}_{\tau=t-\delta}^{t}$.

Then, we perform regression optimization on multiple temporal node metrics to explicitly model metric-aware representations. Precisely similar to the process described in Eq. 2, an L-layer structural encoder generates the representation of nodes in each historical snapshot G_τ a time t:

$$H_\tau^l = \text{StructEncoder}^l(A_\tau, H_\tau^{l-1}), l = 1, ..., L, \tag{6}$$

where H_τ^l is the l-th layer embedding matrix, and H_τ^0 is initialized with the raw feature matrix X_t. Our NMD branch tries to predict the node metric $\mathbf{d}_{\tau+1}^m$ of the next snapshot $G_{\tau+1}$ by each historical embedding H_τ^L:

$$\widehat{\mathbf{d}}_{\tau+1}^m = \text{MLP}_\text{m}(H_\tau^L), \tau = t-\delta, ..., t, \tag{7}$$

where MLP_m is a multi-layer perceptron predictor, which can project H_τ^L into representation spaces of different node metrics. Naturally, we calculate the average MSE (Mean Squared Error) between the ground-truth metric $\mathbf{d}_{\tau+1}^m$ and the predicted metric $\widehat{\mathbf{d}}_{\tau+1}^m$ over the period $[t-\delta, t]$:

$$\mathcal{L}_{NMD}^t = \frac{1}{|\mathcal{M}|} \sum_{m \in \mathcal{M}} \frac{1}{\delta} \sum_{\tau=t-\delta}^{t-1} \text{MSE}\left(\mathbf{d}_{\tau+1}^m, \widehat{\mathbf{d}}_{\tau+1}^m\right), \tag{8}$$

where $\mathcal{L}_{NMD}^{t}$ is our NMD branch loss at time t, $\mathcal{M}$ is the set of multiple optional types of node metrics, and δ is the number of historical snapshots. The above prediction strategy fully utilizes the multiple historical snapshots' information, favoring each historical embedding to establish a non-linear relationship with the node metric of the following snapshot. Our NMD branch builds a fine-grained metric-aware embedding $\{H_{\tau}^{L}\}_{\tau=t-\delta}^{t-1}$ that can represent multiple metric factors of different periods by the regression optimization.

The TLP branch's embedding is mixed with many complex factors, including node metrics. Considering that node metrics may mislead the link prediction results, we devise a decoupling loss to disentangle them from $\{Z_{\tau}^{L}\}_{\tau=t-\delta}^{t-1}$. It can use statistical measures such as distance correlation or mutual information as a regularizer to encourage specialization in representation space. Distance correlation can characterize the independence of any two paired vectors in their linear and nonlinear relationships. Here we calculate the decoupling loss $\mathcal{L}_{DIS}^{t}$ by the commonly-used cosine similarity:

$$\mathcal{L}_{DIS}^{t} = \sum_{\tau=t-\delta}^{t-1} \sum_{i=1}^{N} \frac{\mathbf{z}_{i}^{\tau} \cdot \mathbf{h}_{i}^{\tau}}{||\mathbf{z}_{i}^{\tau}||||\mathbf{h}_{i}^{\tau}||} \tag{9}$$

where $\mathbf{z}_{i}^{\tau} = (Z_{\tau})_{i} \in \mathbf{R}^{d_1}$, $\mathbf{h}_{i}^{\tau} = (H_{\tau})_{i} \in \mathbf{R}^{d_1}$, and $||\cdot||$ is the vector norm.

Under the guidance of $\mathcal{L}_{DIS}^{t}$, the representation function of $\{Z_{\tau}^{L}\}_{\tau=t-\delta}^{t-1}$ and $\{H_{\tau}^{L}\}_{\tau=t-\delta}^{t-1}$ are deconstructed and refined. The former focuses more on personalized aggregation information of the node's multi-hop neighbors, while the latter pays more attention to the global impact of the node itself. If node-level and edge-level factors are highly intertwined without processing, the inconsistency between the TLP and metric regression tasks will lead to a seesaw phenomenon [30]. The negative correlation information can degrade the representation ability, thereby damaging performance. In short, the TLP branch and the NMD branch are devoted to generating representations that do not contain mutual information as much as possible, achieving the disentanglement of the two levels of information.

3.4 Attention-Based Fusion Module

In this section, we need to integrate beneficial information of the metric-aware representation $\{H_{\tau}^{L}\}_{\tau=t-\delta}^{t-1}$ into the TLP embedding $\{Z_{\tau}^{L}\}_{\tau=t-\delta}^{t-1}$, as it only leaves non-metric information after disentangling. We employ the attention mechanism to learn the fusion mode between $\{Z_{\tau}^{L}\}_{\tau=t-\delta}^{t-1}$ and $\{H_{\tau}^{L}\}_{\tau=t-\delta}^{t-1}$, which can discard the damaging information and capture meaningful information.

For each historical snapshot G_{τ} at time t, we first learn N nodes' attention vector $\mathbf{a}^{Z}, \mathbf{a}^{H} \in \mathbf{R}^{N\times 1}$:

$$\mathbf{a}^{Z} = tanh(Z_{\tau}^{L}W)\mathbf{q}, \mathbf{a}^{H} = tanh(H_{\tau}^{L}W)\mathbf{q}, \tag{10}$$

where $tanh$ is a activation function, $W \in \mathbf{R}^{d1\times d2}$ and $\mathbf{q} \in \mathbf{R}^{d2\times 1}$ are learnable weights. As for node $v_i \in V$, we normalize its two attention coefficients a_i^{Z}, a_i^{H} into 2-dimensional probability $\mathbf{p}_i \in \mathbf{R}^{2}$ by the softmax function.

$$\mathbf{p}_i = softmax([a_i^Z, a_i^H]), i = 1, ..., N, \tag{11}$$

Then we linearly combine v_i's two embeddings, $\mathbf{z}_i^\tau = (Z_\tau)_i \in \mathbf{R}^{d_1}$ and $\mathbf{h}_i^\tau = (H_\tau)_i \in \mathbf{R}^{d_1}$, with the probability $\mathbf{p}_i = [p_{i0}, p_{i1}]$:

$$\widetilde{\mathbf{z}}_i^\tau = p_{i0}\mathbf{z}_i^\tau + p_{i1}\mathbf{h}_i^\tau, i = 1, ..., N, \tag{12}$$

where $\widetilde{\mathbf{z}}_i^\tau$ is v_i's embedding after fusing, which Eq. 4 will use to predict the probability of temporal links.

Finally, our method collaboratively infers temporal links and node metrics and jointly optimizes the loss functions of two branches to simulate network evolution in an interdependent manner. The total loss is summarized as follows:

$$\mathcal{L} = \sum_{t=1+\delta}^{t=T} \mathcal{L}_{TLP}^t + \alpha\mathcal{L}_{NMD}^t + \beta\mathcal{L}_{DIS}^t, \tag{13}$$

where δ is the number of snapshots, $\mathcal{L}_{TLP}^t$ and $\mathcal{L}_{NMD}^t$ are the TLP and NMD branch's losses, $\mathcal{L}_{DIS}^t$ is the decoupling loss, and α and β are trade-off weights.

4 Experiment

4.1 Experimental Setup

Datasets and Baselines. We conducted experiments on five dynamic graphs, including UCI [23], BC-Alpha [16,17], BC-OTC [16,17], DBLP [31], and APS[1]. Aiming to evaluate our NMD plugin's broad applicability, the compared baselines include three different types of TLP methods: one RNN-Based method (Dyngraph2Vec [9]), two GCN-Based methods (GCN-GRU [29], EvolveGCN [25]), and three Attention-Based methods (STGSN [22], DySAT [28], HTGN [35]).

Training Details. The common-used deep learning framework PyTorch [26] is adopted to implement all our experiments. We split the datasets into the training, validation, and testing set with ratios 7:1:2 chronologically. Referring to the general setting of the link prediction tasks, we randomly select 100 negative samples for each edge in each snapshot during data loading. The parameters of model architectures are fixed: all methods' structural encoder layers are 2, and the hidden layer dimension is 128. For a fair comparison, the downstream classifier for all methods is a trainable two-layer perceptron. The Adam optimizer [11] and an early-stopping strategy are adopted for model training.

4.2 Overall Performance

Table 1 shows the performance comparison and improvement of the original TLP methods and our method on five dynamic graphs. The default node metric of the experiment is degree centrality. Our NMD plugin achieves significant and

[1] https://journals.aps.org/datasets.

Table 1. AUC (%) performance of the original temporal link prediction methods (**Base**) and our method (**Ours**) on five dynamic graphs. The **Gain (%)** row reports the mean gain percentage of the six baselines. The reported results are the average scores on all timestamps of the testing set.

Case	UCI		BC-Alpha		BC-OTC		DBLP		APS	
	Base	Ours	Base	Ours	Base	Ours	Base	Ours	Base	Ours
Dyn2Vec	$83.62_{\pm 1.03}$	$\mathbf{89.08_{\pm 1.12}}$	$83.18_{\pm 1.00}$	$\mathbf{87.56_{\pm 1.01}}$	$75.93_{\pm 1.01}$	$\mathbf{84.24_{\pm 1.06}}$	$72.31_{\pm 1.22}$	$\mathbf{74.75_{\pm 1.26}}$	$71.71_{\pm 0.18}$	$\mathbf{76.24_{\pm 0.21}}$
GCN-GRU	$86.73_{\pm 1.10}$	$\mathbf{87.55_{\pm 1.11}}$	$85.43_{\pm 1.26}$	$\mathbf{91.57_{\pm 1.43}}$	$76.06_{\pm 1.10}$	$\mathbf{85.06_{\pm 1.41}}$	$73.33_{\pm 1.02}$	$\mathbf{74.73_{\pm 1.02}}$	$73.17_{\pm 0.14}$	$\mathbf{78.61_{\pm 0.17}}$
EvolveGCN	$87.09_{\pm 1.18}$	$\mathbf{89.95_{\pm 1.12}}$	$82.37_{\pm 1.29}$	$\mathbf{84.82_{\pm 1.46}}$	$81.05_{\pm 1.34}$	$\mathbf{82.65_{\pm 1.29}}$	$75.06_{\pm 0.10}$	$\mathbf{77.02_{\pm 0.10}}$	$72.35_{\pm 0.12}$	$\mathbf{77.25_{\pm 0.13}}$
STGSN	$88.04_{\pm 1.06}$	$\mathbf{90.76_{\pm 1.02}}$	$80.48_{\pm 1.03}$	$\mathbf{83.79_{\pm 1.03}}$	$78.85_{\pm 1.09}$	$\mathbf{81.92_{\pm 1.13}}$	$78.39_{\pm 0.09}$	$\mathbf{79.62_{\pm 0.08}}$	$71.14_{\pm 0.09}$	$\mathbf{77.91_{\pm 0.11}}$
DySAT	$87.51_{\pm 0.04}$	$\mathbf{89.61_{\pm 1.14}}$	$82.19_{\pm 1.16}$	$\mathbf{85.96_{\pm 1.33}}$	$78.24_{\pm 1.15}$	$\mathbf{83.19_{\pm 1.25}}$	$73.89_{\pm 1.08}$	$\mathbf{75.60_{\pm 0.21}}$	$72.14_{\pm 0.14}$	$\mathbf{78.77_{\pm 0.24}}$
HTGN	$88.60_{\pm 0.15}$	$\mathbf{91.36_{\pm 1.16}}$	$90.84_{\pm 1.18}$	$\mathbf{94.31_{\pm 1.32}}$	$80.97_{\pm 1.17}$	$\mathbf{84.94_{\pm 1.24}}$	$73.44_{\pm 0.12}$	$\mathbf{78.05_{\pm 0.26}}$	$73.62_{\pm 0.19}$	$\mathbf{79.97_{\pm 0.29}}$
Gain(%)	+2.79		+3.92		+5.15		+2.22		+5.77	

consistent improvements in six different baselines, with the mean gain of 2.79%, 3.92%, 5.15%, 2.22%, and 5.77% on five datasets, respectively. Part of the reason for our success is a more fine-grained embedding representation than baselines, which fully exploits the advantages of node metric factors by the decoupling loss and the attention-based fusion module. It reveals that our work is conducive to promoting the development of link predictions and recommender systems.

Table 2. AUC(%) scores and performance improvement of three loss terms and three node metrics on UCI and BC-Alpha, where DG, CN, and PG are abbreviations for degree centrality, closeness centrality, and PageRank, respectively.

Loss Term			Node Metric			UCI			BC-Alpha		
$\mathcal{L}^t_{TLP}$	$\mathcal{L}^t_{NMD}$	$\mathcal{L}^t_{DIS}$	DG	CN	PG	Dyn2Vec	GCN-GRU	DySAT	Dyn2Vec	GCN-GRU	DySAT
✓						83.62	86.73	87.51	83.18	85.43	82.19
✓	✓		✓			86.46 (+2.84)	83.34 (−3.39)	88.38 (+0.87)	86.66 (+3.48)	89.57 (+4.14)	83.43 (+1.24)
✓	✓	✓	✓			89.08 (+5.46)	87.55 (+0.82)	89.61 (+2.10)	87.56 (+4.38)	91.57 (+6.14)	85.96 (+3.77)
✓	✓	✓		✓		88.65 (+5.03)	87.36 (+0.63)	89.44 (+1.93)	89.03 (+5.85)	92.05 (+6.62)	83.79 (+1.60)
✓	✓	✓			✓	89.48 (+5.86)	88.32 (+1.59)	90.76 (+3.25)	84.68 (+1.50)	90.47 (+5.04)	85.56 (+3.37)
✓	✓	✓	✓	✓		90.65 (+7.03)	88.23 (+1.50)	91.41 (+3.90)	89.13 (+5.95)	93.79 (+8.36)	87.98 (+5.79)
✓	✓	✓		✓	✓	90.08 (+6.46)	89.86 (+3.13)	92.14 (+4.63)	89.71 (+6.53)	93.73 (+8.30)	89.06 (+6.87)
✓	✓	✓	✓	✓	✓	90.16 (+6.45)	90.06 (+3.33)	93.33 (+5.82)	90.29 (+7.11)	94.18 (+8.75)	89.57 (+7.38)

4.3 Ablation Study

To intuitively understand the effect of each part of our NMD plugin, we implement the ablation experiment consisting of three loss terms and three kinds of node metrics. The former includes $\mathcal{L}^t_{TLP}$, $\mathcal{L}^t_{NMD}$, and $\mathcal{L}^t_{DIS}$. The latter includes degree centrality (DG), closeness centrality (CN), and PageRank (PR). Table 2 shows the AUC scores and performance improvement under different experimental settings on UCI and BC-Alpha datasets.

Using only $\mathcal{L}^t_{NMD}$ without $\mathcal{L}^t_{DIS}$ may lead to performance degradation, such as a decrease of 3.39% of the GCN-GRU baseline on the UCI dataset. It indicates a seesaw phenomenon [30] in the TLP and NMD tasks, *i.e.*, a negative

correlation between the edge-level and node-level tasks, affecting the quality of the fused embedding. Hence we decouple the global properties of the nodes from the edge-level embedding of the TLP branch by the decoupling loss $\mathcal{L}_{DIS}^{t}$, which brings 0.82% improvements of the GCN-GRU baseline on the UCI dataset. It emphasizes the importance of disentangling node metric factors. In addition, the performance of different node metrics is very similar due to the general consistency of node attribute information in graph data. From the results, we find that our plugin achieves greater relative improvement as the number of node metric factors increases. We conjecture that disentangling more node metrics can obtain more fine-grained embedding with better representation ability.

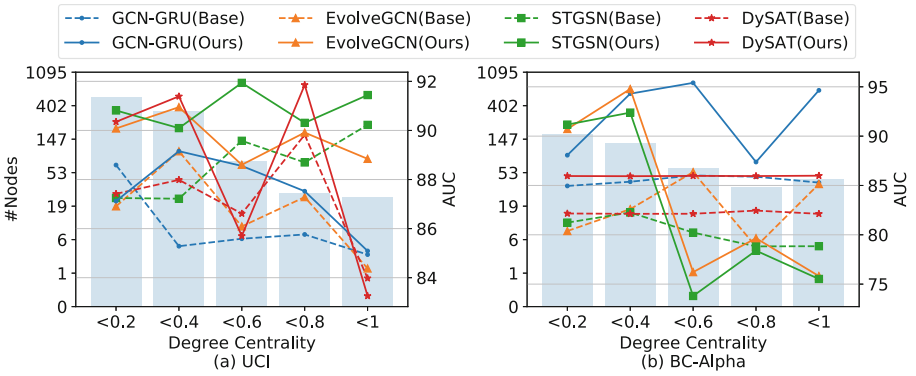

Fig. 2. AUC comparison over the degree centrality of node groups on UCI and BC-Alpha datasets, where the background histograms indicate the number of nodes (#Nodes) involved in each group. The dotted line and the solid line represent the TLP baselines and our improved methods, respectively.

4.4 Performance on Different Node Groups

To carefully explore the effectiveness of our plugin on nodes with different metrics values, we divide the node set into five groups according to the degree centrality of nodes. Figure 2 displays the AUC performance on five node groups with different levels on UCI and BC-Alpha datasets. As for the two GCN-Based methods (GCN-GRU and EvolveGCN), the performance improvement of medium to high degree nodes is more significant than those with low degree. While as for the two Attention-Based methods (STGSN and DySAT), the performance of the five groups of nodes has significantly improved. One possible reason is that the attention mechanism is better at capturing the correlation between node embedding and metrics in the historical link state to promote the generation of high-quality representations. In short, our plugin consistently promotes the representation learning of node groups with different metrics distribution ranges.

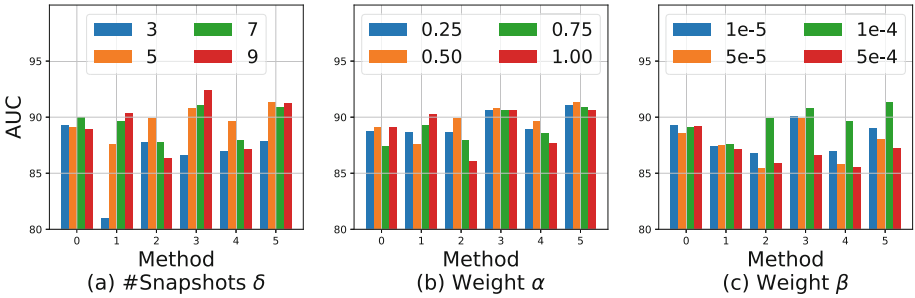

Fig. 3. Parameter sensitivity of six baselines based on our plugin. Methods are indexed in the following order: Dyn2vec, GCN-GRU, EvolveGCN, STGSN, DySAT, and HTGN.

4.5 Parameter Sensitivity

In this section, our parameter sensitivity experiment includes the number of historical snapshots (#Snapshots) and the trade-off weights α and β. The three hyper-parameters' default settings are 5, 0.50, and $1e^{-4}$. Figure 3 shows the performance comparison of different values over six baselines with our plugin on the UCI dataset. Firstly, most methods benefit from more historical snapshots because of more temporal information. Secondly, the trade-off weight α, defined as Eq. 13, controls the relative importance of the edge-level TLP loss and the node-level NMD loss in the dual-branch framework. The higher α performance is better, which proved that the NMD plugin achieved our expected effect, namely, learning fine-grained node embedding. Finally, the performance of different values of β is similar. It reflects that the decoupling loss can steadily improve the diversity and specialization of embedding representations.

5 Conclusion

In this paper, we point out the benefits and harmfulness of node metric factors in temporal link prediction representation. The proposed NMD plugin disentangles node metrics from the node embedding generated by most TLP methods. We further devise an attention-based module to explore fine-grain and high-quality embedding representation. In the future, we will extend our auxiliary components to predict continuous-valued timestamped links and explore the problem of mutual promotion and restriction between node-level and edge-level tasks.

Acknowledgements. This work is funded by the National Key Research and Development Project (Grant No: 2022YFB2703100), the Starry Night Science Fund of Zhejiang University Shanghai Institute for Advanced Study (Grant No. SN-ZJU-SIAS-001), and the Fundamental Research Funds for the Central Universities (2021FZZX001-23, 226-2023-00048).

References

1. Adamic, L.A., Adar, E.: Friends and neighbors on the web. Soc. Netw. **25**(3), 211–230 (2003)
2. Ahmad, I., Akhtar, M.U., Noor, S., Shahnaz, A.: Missing link prediction using common neighbor and centrality based parameterized algorithm. Sci. Rep. **10**(1), 1–9 (2020)
3. Barabási, A.L., Albert, R.: Emergence of scaling in random networks. Science **286**(5439), 509–512 (1999)
4. Barros, C.D., Mendonça, M.R., Vieira, A.B., Ziviani, A.: A survey on embedding dynamic graphs. ACM Comput. Surv. (CSUR) **55**(1), 1–37 (2021)
5. Bedi, P., Sharma, C.: Community detection in social networks. Wiley Interdiscip. Rev. Data Mining Knowl. Discov. **6**(3), 115–135 (2016)
6. Divakaran, A., Mohan, A.: Temporal link prediction: a survey. N. Gener. Comput. **38**, 213–258 (2020)
7. Dunlavy, D.M., Kolda, T.G., Acar, E.: Temporal link prediction using matrix and tensor factorizations. ACM Trans. Knowl. Discov. Data (TKDD) **5**(2), 1–27 (2011)
8. Freeman, L.C.: Centrality in social networks conceptual clarification. Soc. Netw. **1**(3), 215–239 (1978)
9. Goyal, P., Chhetri, S.R., Canedo, A.M.: Capturing network dynamics using dynamic graph representation learning. US Patent App. 16/550,771 (2020)
10. Katz, L.: A new status index derived from sociometric analysis. Psychometrika **18**(1), 39–43 (1953)
11. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
12. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)
13. Kitsak, M., et al.: Identification of influential spreaders in complex networks. Nat. Phys. **6**(11), 888–893 (2010)
14. Klemm, K., Serrano, M., Eguíluz, V.M., Miguel, M.S.: A measure of individual role in collective dynamics. Sci. Rep. **2**(1), 1–8 (2012)
15. Kong, X., Shi, Y., Yu, S., Liu, J., Xia, F.: Academic social networks: modeling, analysis, mining and applications. J. Netw. Comput. Appl. **132**, 86–103 (2019)
16. Kumar, S., Hooi, B., Makhija, D., Kumar, M., Faloutsos, C., Subrahmanian, V.: Rev2: fraudulent user prediction in rating platforms. In: Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, pp. 333–341. ACM (2018)
17. Kumar, S., Spezzano, F., Subrahmanian, V., Faloutsos, C.: Edge weight prediction in weighted signed networks. In: 2016 IEEE 16th International Conference on Data Mining (ICDM), pp. 221–230. IEEE (2016)
18. Lin, D., Wu, J., Xuan, Q., Chi, K.T.: Ethereum transaction tracking: inferring evolution of transaction networks via link prediction. Phys. A **600**, 127504 (2022)
19. Lorrain, F., White, H.C.: Structural equivalence of individuals in social networks. J. Math. Sociol. **1**(1), 49–80 (1971)
20. Ma, J., Cui, P., Kuang, K., Wang, X., Zhu, W.: Disentangled graph convolutional networks. In: International Conference on Machine Learning, pp. 4212–4221. PMLR (2019)
21. Ma, J., Zhou, C., Cui, P., Yang, H., Zhu, W.: Learning disentangled representations for recommendation. In: Advances in Neural Information Processing Systems, vol. 32 (2019)

22. Min, S., Gao, Z., Peng, J., Wang, L., Qin, K., Fang, B.: Stgsn-a spatial-temporal graph neural network framework for time-evolving social networks. Knowl.-Based Syst. **214**, 106746 (2021)
23. Opsahl, T., Panzarasa, P.: Clustering in weighted networks. Soc. Netw. **31**(2), 155–163 (2009)
24. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: bring order to the web. Technical report, Stanford University (1998)
25. Pareja, A., et al.: Evolvegcn: evolving graph convolutional networks for dynamic graphs. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 5363–5370 (2020)
26. Paszke, A., et al.: Pytorch: an imperative style, high-performance deep learning library. In: Advances in Neural Information Processing Systems, vol. 32 (2019)
27. Qin, M., Yeung, D.Y.: Temporal link prediction: a unified framework, taxonomy, and review. arXiv preprint arXiv:2210.08765 (2022)
28. Sankar, A., Wu, Y., Gou, L., Zhang, W., Yang, H.: Dysat: deep neural representation learning on dynamic graphs via self-attention networks. In: Proceedings of the 13th International Conference on Web Search and Data Mining, pp. 519–527 (2020)
29. Seo, Y., Defferrard, M., Vandergheynst, P., Bresson, X.: Structured sequence modeling with graph convolutional recurrent networks. In: Cheng, L., Leung, A.C.S., Ozawa, S. (eds.) ICONIP 2018. LNCS, vol. 11301, pp. 362–373. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-04167-0_33
30. Tang, H., Liu, J., Zhao, M., Gong, X.: Progressive layered extraction (PLE): a novel multi-task learning (MTL) model for personalized recommendations. In: Proceedings of the 14th ACM Conference on Recommender Systems, pp. 269–278 (2020)
31. Tang, J., Zhang, J., Yao, L., Li, J., Zhang, L., Su, Z.: Arnetminer: extraction and mining of academic social networks. In: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 990–998 (2008)
32. Travençolo, B.A.N., Costa, L.D.F.: Accessibility in complex networks. Phys. Lett. A **373**(1), 89–95 (2008)
33. Wang, X., Jin, H., Zhang, A., He, X., Xu, T., Chua, T.S.: Disentangled graph collaborative filtering. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1001–1010 (2020)
34. Wu, J., et al.: Disenkgat: knowledge graph embedding with disentangled graph attention network. In: Proceedings of the 30th ACM International Conference on Information & Knowledge Management, pp. 2140–2149 (2021)
35. Yang, M., Zhou, M., Kalander, M., Huang, Z., King, I.: Discrete-time temporal network embedding via implicit hierarchical learning in hyperbolic space. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, pp. 1975–1985 (2021)
36. Yu, W., Aggarwal, C.C., Wang, W.: Temporally factorized network modeling for evolutionary network analysis. In: Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, pp. 455–464 (2017)
37. Zhou, T., Lü, L., Zhang, Y.C.: Predicting missing links via local information. Eur. Phys. J. B **71**, 623–630 (2009)