Efficient Link Prediction in Continuous-Time Dynamic Networks Using Optimal Transmission and Metropolis Hastings Sampling

Ruizhi Zhang, Wei Wei, Qiming Yang, Zhenyu Shi, Xiangnan Feng, and Zhiming Zheng

***Abstract*—Effective link prediction in continuous time dynamic networks is a challenging issue that has received much research attention in recent years. A widely used method for dynamic network link prediction is to extract the local structure of the target link through random walks on the network, and then employ encoder models to learn node features. However, this method often incorporates prior information, assuming that candidate neighbors adhere to predefined criteria for spatially adjacent, without considering the temporally proximate principles and its consistence with spatial similarity and high-order correlation. To address this limitation, we propose a framework in Continuous-time dynamic networks based on Optimal Transmission and Metropolis Hastings sampling (COM). Specifically, we utilize optimal transmission theory to calculate the distribution similarity between the current node and the time-valid candidate neighbors, aiming to minimize the loss of temporal information during node information propagation. Then we couple spatial structural information by applying the Metropolis-Hastings algorithm, which captures the local structure of target links and global spatial correlations within the network. We realize the fusion of spatiotemporal information using an encoder model. The extensive experiments performed on 14 datasets from different fields demonstrate the superiority of COM compared with representative and state-of-the-art competitors.**

***Index Terms*—Continuous-time networks, dynamic link prediction, Metropolis Hastings sampling, optimal transmission, spatiotemporal information fusion.**

I. Introduction

NETWORK data is ubiquitous across numerous fields, including social networks, academic citation and collaboration networks [1]. For dynamic graphs, entities are represented by nodes, and the connections between them, referred to as edges or links. The majority of links in these networks are continuously evolving over time, making it crucial to develop an effective continuous time dynamic network link prediction model. Such a model can reveal the underlying evolution process of the network, facilitating a better understanding of the structure, dynamics, and properties of the network. Furthermore, dynamic link prediction models can be used for various practical applications, including recommendation systems, traffic-flow prediction, and fraud detection [2], [3], [4].

Dynamic link prediction has received considerable attention in recent years, with the aim of modeling the evolution of networks and predicting link relationships between nodes in future time periods. The existing method for dynamic network link prediction can be categorized into two strategies: discretization-based and continuity-based modeling. Firstly, the discretization-based method divides the dynamic network into snapshots at multiple time points [5], [6]. These snapshots are utilized in conjunction with static link prediction methods or impact maximization techniques on multi-layer networks for link prediction. In particular, community-based context-aware influence maximization (C2IM) employs diffusion models to maximize social impact in the network, and least cost influence (LCI) addresses the challenge of least cost influence maximization by mapping a set of networks into a single one [7], [8], [9], [10]. On the other hand, the continuity-based approach considers the temporal continuity of the network, which involves methods such as time-valid random walks and graph neural networks to model the network's evolution process, such as time-relaxed temporal random walk (TxTWalk), temporal dependent graph neural network (TDGNN) [11], [12], [13].

However, when considering the temporal continuity of the network and maximizing information propagation, temporal link prediction becomes more challenging. Firstly, the interaction of node information aligns with the cohesive explanation [14], which means that temporally proximate and spatially adjacent nodes are more likely to interact with each other. Existing methods often fail to integrate temporal and spatial information effectively, mainly focusing on the length of random walks [11], [15]. In contrast, our approach focuses on evaluating the temporal distribution of nodes. By evaluating the temporal distribution of nodes, we can more precisely identify nodes that exhibit similar temporal behaviors. This approach enhances our ability to predict interactions between nodes that share similar temporal

Received 18 July 2024; revised 16 May 2025; accepted 19 June 2025. Date of publication 23 June 2025; date of current version 21 November 2025. This work was supported in part by the National Natural Science Foundation of China under Grant 62276013, Grant 62141605, and Grant 62050132, in part by Beijing Municipal Natural Science Foundation under Grant 1192012, and in part by the Fundamental Research Funds for the Central Universities. Recommended for acceptance by Dr. Y. Yang. *(Corresponding author: Wei Wei.)*

Ruizhi Zhang, Qiming Yang, and Zhenyu Shi are with the School of Mathematical Sciences, Beihang University, Beijing 100191, China.

Wei Wei is with the School of Mathematical Sciences, Beihang University, Beijing 100191, China, also with the Key Laboratory of Mathematics Informatics Behavioral Semantics, Ministry of Education, Beijing 100191, China, also with the Institute of Artificial Intelligence, Beihang University, Beijing 100083, China, and also with Zhongguancun Laboratory, Beijing 100094, China (e-mail: weiw@buaa.edu.cn).

Xiangnan Feng is with the Complexity Science Hub, 1080 Vienna, Austria.

Zhiming Zheng is with the Institute of Artificial Intelligence, Beihang University, Beijing 100083, China.

Digital Object Identifier 10.1109/TNSE.2025.3582501

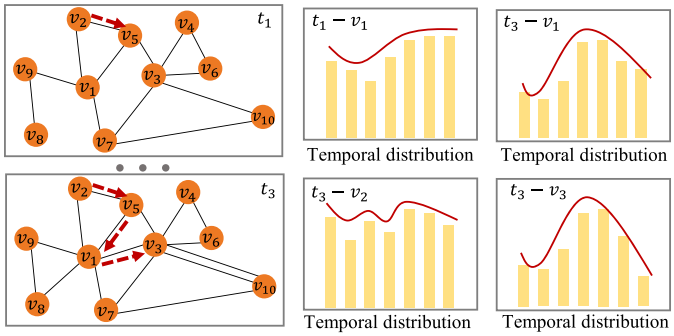

Fig. 1. A motivating example illustrating the evolution of node temporal distributions on dynamic graphs. In the left figure, as time proceeds, the edges in the network increase and the temporal distributions of nodes evolve, as observed in the variations of node v_1 in the top right figures. Distribution-based temporal random walks capture similarities, reflecting spatial adjacency and temporal proximity. For instance, in the left figure, at time t_3, node v_1 selects node v_3 instead of node v_2 due to their similar distributions, as demonstrated in the top right and bottom right distribution figures.

states, thereby minimizing the loss in the process of information propagation. As shown in Fig. 1, node v_1 is more likely to choose node v_3. This suggests that nodes tend to connect not only with spatially adjacent nodes but also with those that are temporally proximate.

Secondly, the diversity of networks emphasizes the complexity of analyzing network topology. Different networks show intricate high-order structures, like motifs or simplex structures [16], [17]. The current methods typically capture these high-order structures locally by introducing predefined criteria, which can lead to models that overly rely on specific patterns in the training data, resulting in unsatisfactory predictions. Therefore, utilizing a method of global evolution to guide the generation of local structures is advantageous. This approach reduces the dependence on predefined criteria, thereby lowering model complexity and the risk of overfitting.

To address these challenges, in this article, we introduce a novel distribution-dependent deep random walk framework aimed at addressing the aforementioned limitations and extracting potential spatiotemporal patterns from dynamic graphs for temporal link prediction. Initially, we propose an optimal transport module for node temporal information propagation based on continuous-time evolution [18]. By quantifying the Wasserstein distance between node temporal distributions, this module not only measures temporal similarity but also identifies critical topological relationships through the analysis of optimal transport paths [19]. For capturing high-order topological correlations in the spatial dimension, we employ the Metropolis-Hastings (MH) algorithm [20]. It alleviates the oversampling of hub nodes by promoting balanced exploration across the state space, thereby improving the coverage of sparse areas in long-tailed degree distributions. Then, we design a lightweight encoder to adjust the node representation, preserving both temporal dependencies and topological characteristics, thereby generating high-quality spatiotemporal topology features for the network.

Our contributions can be summarized as follows:

1) We propose a novel scalable framework, named COM, for continuous-time dynamic network link prediction, in which we simultaneously exploit both network temporal information and higher-order topological features to improve prediction accuracy.
2) We design a new method to capture spatiotemporal patterns in networks. Specifically, the optimal transmission module is introduced to capture the similarity between node state distributions through temporal information propagation. Additionally, we utilize the Metropolis-Hastings algorithm to extract high-order structures, which reflect the relationship between local structural and global networks with different distributions.
3) We conduct extensive experiments on 14 real networks from different fields, and the results indicate its superiority over several state-of-the-art methods.

The remainder of this article is organized as follows. Section II reviews some related work and problem statements are introduced in Section III. Section IV presents the proposed algorithm in detail. Section V evaluates the performance of the COM, and Section VI concludes the investigation.

II. Related Work

Dynamic network link prediction is a critical problem in network analysis, particularly in the era of Big Data where the amount of information generated by networks is increasing rapidly. Existing link prediction approaches can be broadly categorized into three main paradigms: path-based methods, embedding methods, and graph neural networks. Table I summarizes the research focuses and differences among various link prediction methods, organized by method category.

Path-Based Methods: Path-based methods are one of the traditional approaches for link prediction in dynamic networks. They aim to capture the similarity between nodes based on their common paths or walks. Commonly used path-based methods include Katz index [21] and Local Path index [22]. Several studies have proposed variations and extensions of path-based methods, such as the relation strength similarity (RSS) [23] and the PathSim algorithm [24]. Path-based methods for link prediction offer high accuracy and interpretability, but suffer from computational complexity issues and the inability to capture dynamic network properties. Some methods for influence maximization in multi-layer networks can also be applicable to link prediction. MIM2 addresses the challenge of multiple influence maximization by performing a mapping to couple a set of networks into a multiplex network via a direct linkage strategy [25]. LAPSO-IM proposes a discrete particle swarm optimization algorithm, leveraging learning automata to enhance social network influence maximization [26]. ACO-IM adopts a local influence evaluation heuristic method to approximate the local influence within the two-hop area [27]. IM-SSO introduced a global impact evaluation function for the IM optimization problem. This function offers reliable expected diffusion values of influence spread under traditional diffusion models [28].

TABLE I
SUMMARY AND COMPARISON OF LINK PREDICTION METHODS

Type	Name	Contributions and Differences	Reference
Metric-driven heuristics	CN	Local heuristics based on shared neighbors, assuming higher link probability for node pairs with more common neighbors.	[35]
	JC	Local heuristics estimating link likelihood via Jaccard index, measuring the ratio of shared neighbors to total neighbors.	[36]
	EM	Similarity index applying Dempster-Shafer theory and Shannon entropy, inferring link likelihood from local information.	[37]
	SPM	Structural heuristics evaluating network regularity by analyzing feature changes after random edge removal.	[38]
	Katz	Global heuristics quantifying node proximity by integrating all possible paths, weighted by exponential decay to emphasise shorter path contributions.	[21]
	Local Path	Enhanced CN heuristics incorporating length-3 paths, optimizing the trade-off between prediction accuracy and computational complexity.	[22]
	RSS	Path-based metric aggregating all connecting paths, where longer paths contribute less.	[23]
	PathSim	Meta-path similarity for heterogeneous networks, evaluating node similarity through symmetric meta-paths.	[24]
	MIM2	Multiplex network approach, mapping multiple networks into an integrated coupled structure via direct linkage.	[25]
	LAPSO-IM	Integrated with learning automata, discrete particle swarm optimization is used to optimize the influence maximization problem in social networks.	[26]
	ACO-IM	Heuristic optimization leveraging local influence estimation within a 2-hop neighborhood, enhancing prediction accuracy.	[27]
	IM-SSO	Global impact evaluation function for estimating influence spread in traditional diffusion models.	[28]
	DMAB	Activity-loyalty inspired dynamic prediction, utilizing activity backbone for link formation and maintain-develop for persistence.	[39]
Neural architectures	GCN	Graph convolution-based aggregation of neighboring node features to learn node representations and capture structural information.	[32]
	GAT	Enhanced GCN incorporating attention mechanisms to dynamically assign weights to neighboring nodes based on their importance.	[33]
	TDGNN	Dynamic GNN incorporating a temporal aggregator with exponential decay weights to adjust the importance of neighbor nodes over time.	[12]
	GC-LSTM	Hybrid model combining GCN and LSTM to capture both structural and temporal dependencies in dynamic networks.	[34]
	DySAT	Dynamic embedding utilizing self-attention mechanisms to capture temporal dependencies between nodes and their contexts.	[49]
	TGAT	Temporal graph attention network integrating structural information with time-aware features to model dynamic relationships.	[50]
	SE-GRU	Neural framework embedding local topological features to enhance robustness against structural variability and temporal delays.	[51]
	TREND	GCN-based model inspired by the Hawkes process, capturing both node and event popularity to model individual and collective dynamics.	[52]
Latent space embedding	Node2vec	Embeddings through biased random walks, capturing both local and global network structures.	[29]
	Deepwalk	Embeddings through uniform random walks to learn node representations, capturing structural relationships in the network.	[30]
	LINE	Embeddings preserve first-order and second-order proximities by optimizing edge likelihood in a low-dimensional space.	[31]
	AdaSim	Embeddings with tunable similarity functions, leveraging random walks to capture network structures and node relationships.	[40]
	CTDNE	Temporal embeddings leveraging time-aware node sampling to model dynamic interactions.	[15]
	COM	Distribution-aware spatiotemporal embeddings combining optimal transport for temporal consistency and Metropolis-Hastings sampling for structural correlations.	This paper.

Embedding Methods: Link prediction using embedding methods has gained much attention in recent years due to their ability to capture both structural and semantic information in networks. These methods generate low-dimensional vector representations for nodes, which can be used for predicting links. One popular embedding method is the node2vec [29] algorithm, which learns node representations by performing biased random walks on the network and then uses Skip-gram models to learn representations based on the sequences of nodes visited during the walks. Other methods, such as DeepWalk [30] and LINE [31], have also been proposed for link prediction based on node embeddings. Embedding-based methods for link prediction have the advantage of being computationally efficient and able to handle large-scale networks.

Graph Neural Networks Methods: Deep learning models can learn complex patterns and non-linear relationships from network data, and have achieved promising results in link prediction tasks. The most popular deep learning models for link prediction include graph convolutional networks (GCNs) [32], and attention-based models [33]. TDGNN integrates temporal information into GNNs through a novel temporal aggregator, which assigns aggregation weights to neighboring nodes based

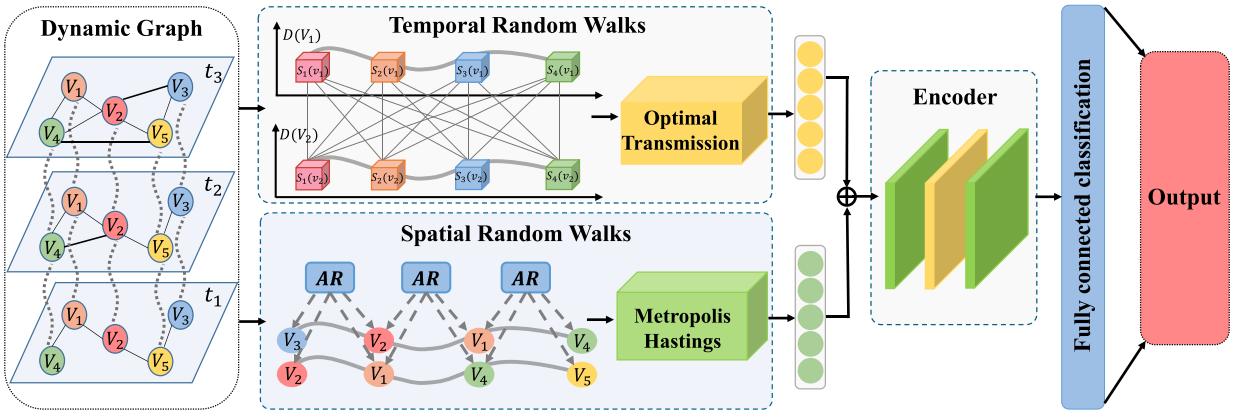

Fig. 2. Schematic of COM. COM first conducts sampling on the graph to generate temporal and spatial random walks, respectively. Within the temporal dimension, the optimal transmission module is employed to measure the similarity between node state distributions, aiming to maximize information propagation. In the spatial dimension, MH sampling is utilized to derive Markov chains that follows the global network distribution to capture structural information. Edge embeddings are then obtained using an encoding model. Finally, a fully connected classifier is employed to score and predict future link connections. The $D(V_1)$ and $D(V_2)$ in the figure represent the state distribution of nodes v_1, and v_2. AR denotes the acceptance rate, which determines each step of spatial random walk.

on an exponential distribution, thereby emphasizing the temporal differences among edges [12]. GC-LSTM is a novel end-to-end model for dynamic network link prediction, leveraging the strengths of GCN for local structural learning and LSTM for modeling temporal dependencies across network snapshots [34].

III. Preliminaries and Problem Formulation

Definition 3.1 (Continuous-Time Dynamic Networks): Continuous-time dynamic networks are networks in which the links between nodes can appear and disappear over time in a continuous fashion, rather than in discrete time steps. In mathematical terms, given a time-varying graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}_\mathcal{T}, \mathcal{T})$, where $\mathcal{V}$ is a set of nodes or vertices, and $\mathcal{E}_\mathcal{T}$ is a set of edges or links that varies with time $\mathcal{T}$. $|\mathcal{V}|$ and $|\mathcal{E}_\mathcal{T}|$ denote number of nodes and edges. This paper concerns graphs that are directed and do not have weights assigned to the edges.

The edges in $\mathcal{E}_\mathcal{T}$ can appear or disappear at any point in time, and are represented by a triplet $e^t_{i,j} = (v_i, v_j, t) \in \mathcal{E}_\mathcal{T}$, where v_i and v_j are the nodes that the edge connects. Nodes that are connected imply that they are neighbors to each other. We further define the temporal neighbors.

Definition 3.2 (Temporal Neighborhood): For a given node $v \in \mathcal{V}$ and a timestamp $t \in \mathcal{T}$, the temporal neighborhood of v is denoted as $\mathcal{N}_t(v)$. Formally,

$$\mathcal{N}_t(v) = \{(w, t') | e^{t'}_{v,w} = (v, w, t') \in \mathcal{E}_\mathcal{T} \wedge t' > t\}. \quad (1)$$

The size of the temporal neighborhood of a node v at time t is given by the cardinality of $\mathcal{N}_t(v)$, denoted as $|\mathcal{N}_t(v)|$. The temporal neighborhood provides information about the immediate network context of a given node, which can be useful in predicting its future connections.

Definition 3.3 (Link Prediction in Continuous-Time Dynamic Networks): Continuous-time dynamic network link prediction that focuses on predicting the appearance or disappearance of links over a continuous time period. This can be formulated as a random event, where the occurrence of each link is a random event that happens at a specific time with a certain probability. The goal of link prediction is to predict the presence or absence of a link between any two nodes v_i and v_j in the future graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}_\mathcal{T}, \mathcal{T} + \Delta T)$, where ΔT is the time horizon.

Definition 3.4 (Temporal Network Embedding). The goal of temporal network embedding is to learn a function that maps each node in the network to a low-dimensional vector that captures its temporal dynamics and network topology. Formally, mapping $\phi : \mathcal{V} \to \mathbb{R}^l, \phi(u) = x_u$, where x_u is called a temporal embedding of node $u \in \mathcal{V}$.

IV. Methodology

In this section, we delve into the details of generating embeddings and their utilization for link prediction. Fig. 2 illustrates an overview of the entire process.

A. Framework Overview

The key idea of COM is to establish a dynamic network link prediction framework that can capture continuously evolving spatiotemporal pattern on the graph. To achieve this, it is represented as an encoder-decoder model. COM first extracts network patterns in both the temporal and spatial dimensions and encodes them as low dimensional representation vector. Then, the decoder combines the encoder with a simple MLP to make task-specific predictions. It can be represented by the following function.

$$\mathcal{Z}_\mathcal{T} = f(g(\mathcal{G}(\mathcal{V}, \mathcal{E}_\mathcal{T}, \mathcal{T}))). \quad (2)$$

The encoder g in the equation above encodes the dynamic graph to generate node representations based on coupled temporal and spatial features, which are then synthesized to obtain edge representations (the synthesis strategy will be analyzed in the experiment). The decoder f predicts whether edges are connected based on vector decoding, and we use the simplest MLP for this task (other machine learning models could be used instead). In the equation, $\mathcal{Z}_\mathcal{T} \in \{0, 1\}$ represents the predicted

value for the edge to be predicted, where the ground truth $\mathcal{Z}_{\mathcal{T}} = 1$ if there is an interaction between two nodes and 0 otherwise.

B. Modeling Temporal Information Process

In dynamic graphs, edge changes indicate information transmission between nodes. A highly efficient approach for network representation is to integrate temporal information through random walks, ensuring minimal information loss during the process. When nodes exhibit similar structures, the loss of information propagation between them typically decreases. Furthermore, nodes sharing similar distributions often possess similar structures, neighboring nodes or connection patterns, facilitating smoother information flow between them [41]. Optimal transmission, which captures the similarity between distributions, can serve as a powerful tool for maximizing information propagation.

1) Optimal Transmission: The French mathematician Monge first introduced the optimal transmission problem, which aimed to determine the most efficient way to transport sand piles from one location to another by considering various transmission methods with different associated costs. The objective of this problem is to identify the solution with the lowest transmission cost among all possible transmission schemes [18]. It has been shown to effectively identify the most influential nodes in a network and enhance information transmission efficiency. Applying it to our tasks, optimal transmission refers to the process of transmitting information or data from one node to another in the most efficient and effective manner possible with minimal information loss. The optimal transmission problem is defined as minimizing the distance between the node and its neighbors,

$$d(u, v) = \min_{v \in \mathcal{N}_t(u)} < \mathcal{P}_t, \mathcal{G} >, \tag{3}$$

in which $\mathcal{N}_t(u)$ means the time-valid neighbors of u at time t and $\mathcal{P}_t$ represents the matrix composed of node probability distribution.

2) Wasserstein Distance: The optimal transmission problem's minimum value is represented by the Wasserstein distance, a distance measure in probability space that characterizes the minimum transmission cost from source node u to target node v [19]. It is a natural way to define distance between two probability measures in the measurement space, and provides a means to measure the difference between probability measures u and v. The Wasserstein distance formal expression for the probability distributions of u and v at time t, assumed to be p and q, is as follows:

$$\mathcal{W}(p, q) = \inf_{\gamma \sim \Pi(p,q)} E_{x,y \sim \gamma}[||x - y||]. \tag{4}$$

The set of all possible joint distributions that combine distribution p and q is denoted by $\Pi(p, q)$. For each joint distribution γ in this set, a pair of samples x and y can be sampled from it by drawing $(x, y) \sim \gamma$, and the distance $||x - y||$ between these samples can be calculated. The Wasserstein distance is defined as the infimum for the expected value of this distance over all possible joint distributions γ.

Algorithm 1: Temporal-biased Sampling..

Input: Input networks $\mathcal{G} = (\mathcal{V}, \mathcal{E}_{\mathcal{T}}, \mathcal{T})$, max_walk_length L, start_edge $e^t_{v,u} = (v, u, t)$, number of context windows C, context window size ω, number of walks for each node R, temporal context window count $\beta = R \times N \times (L - \omega + 1)$.

Output: Temporal-biased random walk sequences *walks* $\mathcal{W}_t(u)$.

```
Acquire curr_node = u; walks = [ ]
while β − C > 0 do
  for i = 1 : min{L, C} − 1 do
    Compute time-valid neighbors N_t(curr_node) with (1);
    if |N_t(curr_node)| > 0 then
      distance = [ ]
      for v_i ∈ N_t(curr_node) do
        P_t(curr_node) = p, P_t(v_i) = q
        distance.append(Wasserstein(p, q))
      index = argmin(distance)
      next_node = N_t(curr_node)[index]
      curr_node = next_node
      walks.append(curr_node)
  if |walks| > ω
    C = C + (|walks| - ω + 1)
endwhile
```

3) Temporal-Biased Dynamic Random Walks: When it comes to node information propagation in networks, the sampling strategies of CTDNE rely on biased or unbiased distribution assumptions of nodes in a time-valid space, which can result in inaccuracies or loss of information [15]. In contrast, the calculation of the Wasserstein distance based on the probability distribution of nodes offers a way to estimate the minimum loss of time information when sampling the next node. Specifically, the Wasserstein distance captures the similarity between the probability distributions of nodes in the network, enabling the identification of nodes with similar information propagation patterns and facilitating to maximize information propagation. More precisely, at a given time t, for a node u, we select the time-valid neighbors $\mathcal{N}_t(u)$ and calculate the Wasserstein distance based on the probability distribution $(\mathcal{P}_t(u), \ldots, \mathcal{P}_t(v_i)(v_i \in \mathcal{N}_t(u)))$ of node u and its neighbors at the current time t, and use it to determine the next sampling node of the random walk. The probability distribution of a node is composed of the degree information of its neighbor nodes at the current time. We finally obtain the random walk sequence $\mathcal{W}_t(u) = (v_1, \ldots, v_k)(v_1 \in \mathcal{N}_t(u), v_{i+1} \in \mathcal{N}_t(v_i))$ started from node u. In the time dimension, we limit the maximum step size of the random walk, without setting a specific walk length, as in the setting of CTDNE [15]. The overall process of the proposed temporal-biased random walk strategy is given in Algorithm 1.

C. Modeling Spatial Information Process

Network structure plays a key role in understanding network evolution mechanisms. Current methods involve extracting high-order structures by gathering neighbor nodes within 1-hop

or 2-hop distances, or by sampling nodes based on predefined criteria [42]. These methods fail to consider the correlation between local structure and global system evolution patterns, which often leads to overfitting of the model and unsatisfactory prediction. The distribution characteristics of the global network influence the formation and evolution of local structures. In our model, we employed the Metropolis-Hastings sampling algorithm to derive spatial structures, generating random walk sequences that follow the global network distribution.

1) Metropolis Hastings Sampling: Metropolis-Hastings (MH) Sampling is a Markov Chain Monte Carlo (MCMC) method, widely used for generating samples from a probability distribution that is difficult to sample directly [43]. Assuming $\pi(\cdot)$ is the target probability distribution defined on a sample space Ω. The main idea behind MCMC is to construct a Markov chain on Ω that has $\pi(\cdot)$ as its stationary distribution. The MH algorithm is a specific MCMC method used to design a Markov chain with a desired stationary distribution.

To implement the MH algorithm, a conditional probability mass function $q(y|x)$ is first specified, where $x, y \in \Omega$. At each iteration i, a candidate sample x^c is drawn from the proposal distribution $q(\cdot|x^{i-1})$, where x^{i-1} is the sample selected in the previous iteration. The candidate sample is then accepted with a probability given by the MH acceptance ratio $p(x^{i-1}, x^c)$, which depends on the ratio of the target distribution $\pi(\cdot)$ and the proposal distribution $q(\cdot|x^{i-1})$. The formula to compute the acceptance ratio is as follows,

$$p(x^{i-1}, x^c) = min\left\{1, \frac{\pi(x^c) \cdot q(x^{i-1}|x^c)}{\pi(x^{i-1}) \cdot q(x^c|x^{i-1})}\right\} \tag{5}$$

If the candidate sample x^c is accepted, then x^i is updated to x^c. Otherwise, x^i remains the same as the previous sample x^{i-1}.

2) Spatial-Biased Random Walks: Metropolis Hastings sampling provides a more flexible proposal distribution to generate local structures that satisfy the correlation with the global network evolution distribution, and improves sampling efficiency by reducing the number of rejected samples. Clearly, we present an edge sampler utilizing the Metropolis-Hastings (MH) sampling technique on graphs, where the target distribution is assumed to be the uniform distribution. To utilize the MH sampling method, we adopt the uniform distribution as the conditional probability mass function $q(\cdot|u)$, which is given by $q(\cdot|u) = 1/deg(u)$, where $deg(u)$ represents the degree of the node u. As a result of the symmetry of the uniform distribution, the acceptance ratio $p(x^{i-1}, x^c)$ for the ith candidate x^c can be reduced to

$$p(x^{i-1}, x^c) = min\left\{1, \frac{deg(x^{i-1})}{deg(x^c)}\right\} \tag{6}$$

We choose the uniform distribution as the conditional probability mass function $q(\cdot|u)$ due to its computational efficiency in calculating acceptance rates. Furthermore, the use of uniform distribution for the edge sampler allows for convergence to any discrete target distribution [20].

Our spatial-biased sampling strategy aims to generate a representative sampling set correlated with the distribution of global network evolution. By employing the MH algorithm, the resulting Markov chain is both aperiodic and recurrent, and converges to the target distribution. During the random walk based on the MH algorithm on the graph, given node u at time t, the next node is sampled from all neighbors based on the MH algorithm. Unlike temporal-biased sampling, our sampling space includes all neighbors, not just the time-valid ones. Finally, we obtain the spatial-biased random walk sequence $\mathcal{W}_s(u) = (v_1, \ldots, v_k)(v_1 \in \mathcal{N}(u), v_{i+1} \in \mathcal{N}(v_i))$ with a source node of u, where the length of the random walk is a hyperparameter. In our experiment, we set the step size to 8 based on several studies, which suggests that focusing on lower-order neighbors can provide more information for network structure [44]. The overall procedure is described in Algorithm 2.

Algorithm 2: Spatial-biased Sampling..

Input: Input networks $\mathcal{G} = (\mathcal{V}, \mathcal{E}_\mathcal{T}, \mathcal{T})$, max_length max_length, start_edge $e^t_{v,u} = (v, u, t)$.
Output: Spatial-biased random walk sequences $walks\ \mathcal{W}_s(u)$.

```
Acquire curr_node = u
walks = [ ]
while len(walks) < max_length do
    Select node w uniformly at random from neighbors
N(curr_node) of curr_node
    Generate uniformly at random a number 0 ≤ p ≤ 1
    if p < deg(curr_node)/deg(w) then
        curr_node = w
        walks.append(curr_node)
    else
        stay at curr_node
    end while
```

D. Network Topology Scheme

In contrast to traditional models, the random walk mechanism in COM is guided by both temporal node interaction patterns and global topological information. As shown in Fig. 3, node association strength is not only reflected in interaction frequency but also captured by the temporal distribution of event occurrences. By employing the Wasserstein distance to measure differences in temporal distributions, our approach quantifies the optimal transport cost to capture behavioral similarity among nodes, thereby uncovering implicit topological dependencies during temporal evolution.

For spatial exploration, employing a uniform distribution in Metropolis-Hastings (MH) sampling reduces the traditional bias toward hub nodes. This ensures balanced access probabilities throughout the state space, improving coverage in the long-tail region of the degree distribution while avoiding excessive sampling of highly connected nodes. Consequently, the sampled sequences both reflect local connection patterns and capture the statistical properties of the global topological structure.

E. Encoding Spatiotemporal Coupling Random Walks

1) Skip-Gram Embedding: Word2vec represents words as embeddings in a continuous vector space for natural language

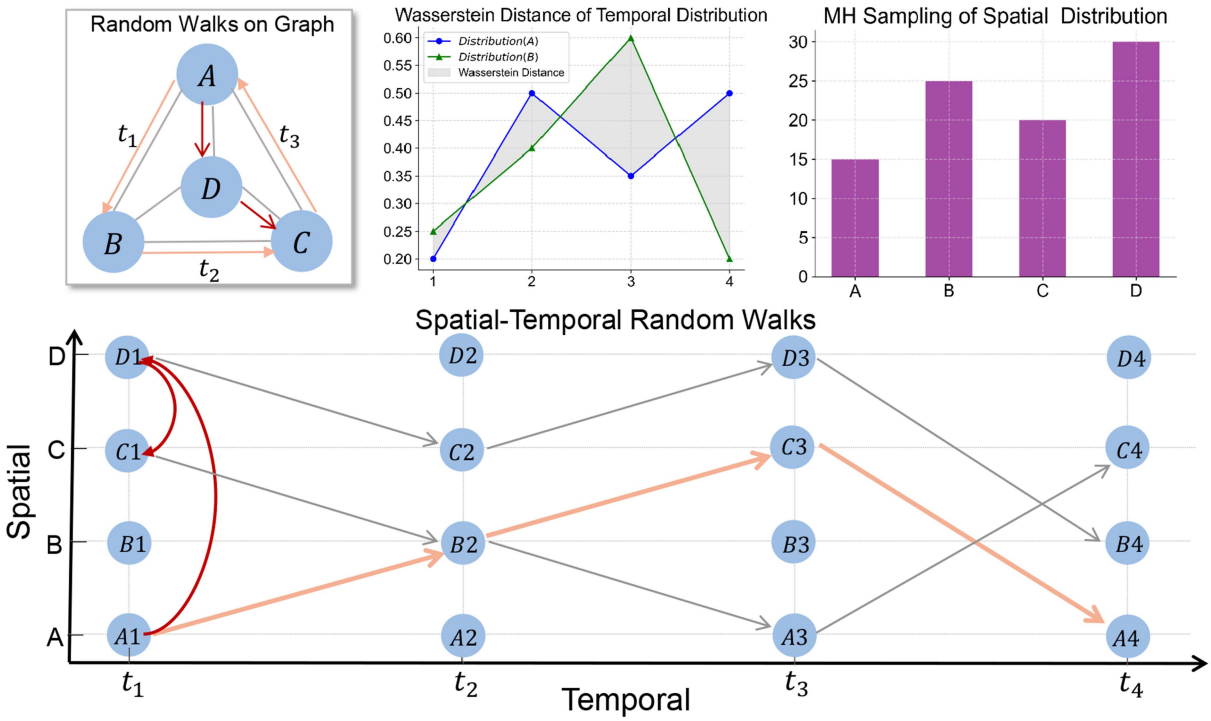

Fig. 3. Schematic of the COM network topology. This toy example illustrates the random walks process (top-left). For node objects in dynamic graphs, the temporal dimension of the random walk (orange lines) is computed via the Wasserstein distance from the temporal distribution (top-middle), while the spatial dimension (red lines) is determined by the degree distribution (top-right). The bottom figure presents random walk sequences from different starting nodes.

processing (NLP) tasks like machine translation, text classification, and sentiment analysis [45]. Skip-gram, a type of the Word2vec, predicts context words for a target word, effectively capturing semantic and syntactic relationships [46]. Building upon the obtained random walk sequences in both the temporal and spatial dimensions, we integrate the two to obtain a novel spatiotemporal random walk sequence $\mathcal{W}_{ts}$. The $\mathcal{W}_{ts}$ sequence consists of three parts that are concatenated together: first the target link, followed by the temporal-biased and spatial-biased sequences. This combined sequence allows for the exploration of both spatial and temporal information in a coherent manner, providing a more comprehensive understanding of the underlying network structure. The mathematical description of Skip-gram used in dynamic network representation learning is as follows. Given a spatiotemporal random walk sequence $\mathcal{W}_{ts}$, it maximizes the co-occurrence probability among the nodes that appear within a window,

$$\max_f \log P(\mathcal{W}_{ts} = \{v_{i-\omega}, \ldots, v_{i+\omega}\} \setminus v_i | f(v_i)), \quad (7)$$

where $f : \mathcal{V} \to \mathbb{R}^D$ is the embedding function, ω is the context window size for optimization. We assume conditional independence of the nodes, and the optimization problem can be formulated as:

$$P(\mathcal{W}_{ts} | f(v_i)) = \prod_{v_j \in \mathcal{W}_{ts}} P(v_j | f(v_i)). \quad (8)$$

Algorithm 3: Edge Representation..

Input: Input networks $\mathcal{G} = (\mathcal{V}, \mathcal{E}_\mathcal{T}, \mathcal{T})$, start_edge $e^t_{v,u} = (v, u, t)$, the number of walks for each node R, binary operations D={*Concatenation, Mean, L1, L2, Hadamard*}.

Output: Edge Representation z_{uv}.

Acquire $curr\_node = u$

For i = 1 : R **do**

$\mathcal{W}^i_{st}(u) = [(v, u), \mathcal{W}^i_t(u), \mathcal{W}^i_s(u)]$

$\mathcal{W}_{st}(u) = [\mathcal{W}^1_{st}(u), \mathcal{W}^2_{st}(u), \ldots, \mathcal{W}^R_{st}(u)]$

Carry out Skip-gram with $\mathcal{W}_{st}(w), w \in \mathcal{V}$

Obtain node embedding z_u, z_v

Choose one binary operation $\odot$ from D

Edge embedding representation $z_{uv} = (z_u \odot z_v)$

F. Edge Representation

Using the node embedding obtained from optimizing the transmission of information in the time dimension and fully exploring the surrounding spatial structure through MH sampling, we generate edge representations via binary operations. The process of generating representations of edges is shown in Algorithm 3. Specifically, we create the final edge representation z_{uv} using the embedding representations z_u and z_v of nodes u and v. We experiment with five different binary operations: Concatenation, Mean, L1, L2 and Hadamard.

TABLE II
STATISTICS OF THE DATASETS USED IN OUR EXPERIMENTS

Dataset	$\lvert\mathcal{V}\rvert$	$\lvert\mathcal{E}_\mathcal{T}\rvert$	$\bar{d}$	Timespan(days)
Contact	274	28.2K	206.2	3.9
Hypertext	113	20.8K	368.5	2.4
Enron	151	50.5K	669.8	1137.5
Radoslaw	167	82.9K	993.1	271.1
Fb-forum	899	33.7K	75	164.4
Email-eu	986	332.3K	674.1	803.9
Bitcoin-Alpha	3.7K	24.1K	12.8	1901
Bitcoin-OTC	5.8K	35.5K	12.1	1904
Slashdot-reply	51K	140.7K	2.7	977.3
Wiki-elec	7.1K	107.0K	30.1	1378.3
Wiki-talk	1.14M	7.83M	6.54	2320
Flickr	2.3M	33.1M	23.78	365
DBLP	1.2K	10.7K	18.24	3653
Prosper	2.6K	46.5K	36.34	62

V. EXPERIMENTS

A. Experimental Setup

1) Datasets: We evaluate the performance of COM using 14 real-world dynamic networks: Contact, Hypertext, Enron, Radoslaw, Fb-forum, Email-eu, Bitcoin-Alpha, Bitcoin-OTC, Slashdot-reply, Wiki-elec, Wiki-talk, Flickr, DBLP and Prosper. Table II provides a summary of the detailed information for each dataset. The datasets we use are from Stanford Network Analysis Platform (SNAP),[1] a general purpose network analysis and graph mining library.

- *Contact:* This human contact network was collected via wireless devices worn by event attendees. A link is established when two individuals contact each other at a specific timestamp.
- *Hypertext:* This network shows face-to-face interactions at the ACM Hypertext Conference 2009, with nodes representing attendees and edges denoting 20-second contacts.
- *Enron:* The Enron email network consists of emails sent between employees of Enron. Nodes in the network are individual employees and edges are individual emails.
- *Radoslaw:* This network represents email exchanges between employees of medium-sized manufacturing companies, with nodes as employees and edges as email exchanges.
- *Fb-forum:* The dataset is from an online student forum at the University of California, similar to Facebook. The network includes users as nodes and their interactions as links, covering over five months.
- *Email-eu:* The network was constructed from email data gathered at a large European research institution. Nodes represent users, and edges represent exchanged emails between them.
- *Bitcoin-Alpha:* This is a who-trusts-whom network from Bitcoin Alpha, where nodes represent users trading Bitcoin. Edges denote ratings from -10 to +10, indicating the source's rating of the target user.
- *Bitcoin-OTC:* Another who-trusts-whom network among bitcoin users trading on the OTC platform, similar to the Bitcoin-Alpha dataset.
- *Slashdot-reply:* The Slashdot reply dataset illustrates a social network platform, capturing interactions among users. It specifically includes reply relationships on the Slashdot forum.
- *Wiki-elec:* The dataset contains all administrator elections and vote history data based on the latest complete dump of Wikipedia page edit history.
- *Wiki-talk:* This dataset, collected between November 2001 and August 2007, records editing activities on user Talk pages. Each node represents a user's Talk page, and each link denotes an editing activity by one user on another user's Talk page.
- *Flickr:* The Flickr dataset, collected from the photo-sharing platform Flickr in 2008, features nodes representing users and edges indicating their friendship connections.
- *DBLP:* The DBLP citation network consists of nodes representing scientific publications, such as papers and books, with edges indicating citations between them.
- *Prosper:* A directed network from Prosper.com shows loans between members, with edges from lenders to borrowers and timestamps indicating when the loans occurred.

2) Evaluation Metrics: The accuracy of link prediction methods in continuous-time dynamic networks can be evaluated using various metrics. In this paper, we apply 1) AUC [47] and 2) AP [48], which measure the performance of the prediction method in terms of true positive rate, false positive rate, and precision.

3) Baselines: To evaluate the effectiveness of COM, we compare it with three kinds of methods: 1) heuristic methods including Common Neighbors (CN), Jaccard Coefficient (JC), EM and SPM. 2) static link prediction methods including node2vec, LINE, AdaSim. 3) dynamic link prediction methods including CTDNE, DySAT, TGAT, SE-GRU, TREND and DMAB.

- *CN [35]:* The algorithm uses the number of common neighbors as an indicator to measure the possibility of establishing a link between two nodes.

$$CN(x,y) = |N(x) \cap N(y)| \tag{9}$$

- *JC [36]:* This algorithm evaluates the probability of connecting edges also by measuring the number of common neighbors, it is the normalized version of CN.

$$JC(x,y) = \frac{|N(x) \cap N(y)|}{|N(x) \cup N(y)|} \tag{10}$$

- *EM [37]:* EM is a link prediction similarity index based on Dempster Shafer theory, which measures the link predictability through local information and Shannon entropy.
- *SPM [38]:* SPM is a structural consistency index based on adjacency matrix perturbation, applicable without prior knowledge of network organization. It assesses network regularity by comparing structural features before and after random removal of a small group of links.
- *Node2vec [29]:* Node2vec is a network representation learning algorithm that optimizes second-order proximity using biased random walks to learn low-dimensional node representations, with DFS and BFS parameters controlling node representation.

[1] https://snap.stanford.edu/

- *LINE [31]:* LINE is a scalable framework for learning high-quality node embeddings in large networks. It optimizes an objective function that incorporates both first-order and second-order proximity.
- *AdaSim [40]:* AdaSim is a link prediction framework that utilizes node features embedding in networks based on random walks. It defines a flexible similarity function, guided by a tunable parameter to optimize the model's performance.
- *CTDNE [15]:* CTDNE is a method for learning low-dimensional representations of nodes in dynamic networks based on their temporal interactions and dynamics, using a node sampling strategy incorporating time information.
- *DySAT [49]:* DySAT is a dynamic graph embedding method that leverages self-attention to capture the temporal dependencies of nodes and their surrounding contexts for link prediction in dynamic networks.
- *TGAT [50]:* TGAT is a deep learning model for dynamic graph representation learning. It employs attention mechanisms to capture temporal and structural information, demonstrating strong performance across dynamic graph prediction tasks.
- *SE-GRU [51]:* SE-GRU is a prediction framework based on neural networks, which embeds the structure of local topological features to strengthen the prediction robustness against frequency variation and occurrence delay of connections.
- *TREND [52]:* TREND is a GCN-based method inspired by the self-exciting effect of the Hawkes process, which integrates event and node popularity to capture both individual and collective event characteristics, enhancing the modeling of dynamic processes.
- *DMAB* [39]: The DMAB model predicts links by using two modules: the Activity Backbone for link formation based on activity, and the Maintain-Develop Module for maintaining or forming links based on loyalty.

We assess the effectiveness of the COM on the temporal link prediction task. For this, we sort edges in each graph by ascending chronological order and use the first 75% for representation learning. We consider the remaining 25% as positive links and randomly generate an equal number of negative edges for generating labeled examples for link prediction. Node2vec, LINE, and DySAY use the logistic regression classifier; CTDNE and TGAT use the MLP classifier. We follow the setting in the original paper for the choice of parameters in the baseline experiment. The specific settings of parameters in the COM model are as follows, the window size ω in Skip-gram model is set to 10. Besides, the maximum walk length L is set to 80, the number of walks for each node R is set to 10, and the embedding dimension D is set to 128.

B. Experimental Results

The link prediction performance of all methods was evaluated in terms of AUC and AP, and the results are summarized in Table III. The reported performance is an average over ten repetitions. To enhance aesthetic layout, we have presented the results of six models [53], as shown in Fig 4. Overall, our proposed model shows competitive performance compared to other methods. Table III shows that the heuristic method performs well on dense networks, while static models such as Node2vec, LINE and AdaSim perform poorly since they only consider node representations obtained from random walks without temporal information. Dynamic graph models, which capture temporal information in the network, outperform static models. The performance of CTDNE is improved by using a random walk algorithm that considers temporal information. On most datasets, graph neural network methods DySAT, TGAT, SE-GRU and TREND perform better than other methods, as they can aggregate neighbor information through attention mechanisms to obtain better node representations. From an interpretability perspective, DMAB achieves competitive performance by leveraging node activity and loyalty, without relying on the message-passing aggregation mechanism of GNNs. Our proposed method, which optimizes temporal information propagation while coupling spatial structures information, achieves better performance in most cases than other dynamic models. Although its AP values perform slightly worse in Hypertext and Wiki-elec, it is still competitive. In summary, our method can achieve satisfying results on both sparse and dense graphs. The promising results presented above suggest that the node embeddings obtained through our method can be utilized as favorable features for link prediction tasks.

We experiment with different training percentages and presented the AUC results of our model comparing CN, node2vec, and CTDNE in Fig. 5. We used the last 25% of the edges as the test set and gradually increased the number of training edges in reverse chronological order from 35% to 75%. Our proposed method's AUC values are indicated by red lines, and baseline methods are denoted with different colors. The figure showed that our proposed method outperformed almost all baseline methods under different percentages of training data. COM demonstrates stable AUC performance across different domains, such as Flickr, DBLP, and Prosper. This indicates that our model exhibits strong generalization ability, effectively adapting to feature variations across domains, while maintaining robustness and stability when handling complex data from various domains. The CN method's performance deteriorated with the increase of the training set proportion on the Contact, Hypertext, and Radoslaw datasets, possibly due to the addition of edges far from the current time period to the network, which introduced more noise to the model. Node2vec's performance is unstable and fluctuates significantly across most datasets. The trend of CTDNE is very similar to our model. In general, a larger training set should generally provide a better performance. COM exhibits consistent and stable performance on different training sets of most datasets, demonstrating its capacity to learn essential network features and patterns and its robustness and generalization across different data scales. Notably, for datasets Hypertext, Enron, and Wiki-elec, when increasing the training rate from 65% to 75%, there is a notable enhancement in the model's performance, which emphasizes the relevance of recent data for precise predictions.

TABLE III
EXPERIMENTAL RESULTS OF DIFFERENT METHODS ON DIFFERENT DATASETS

Datasets	Contact		Hypertext		Enron		Radoslaw		Fb-forum		Email-eu		Bitcoin-Alpha	
Metrics	***AUC***	***AP***	***AUC***	***AP***	***AUC***	***AP***	***AUC***	***AP***	***AUC***	***AP***	***AUC***	***AP***	***AUC***	***AP***
CN	0.969	0.973	0.761	0.753	0.936	0.933	0.942	0.935	0.700	0.687	0.962	0.959	0.721	0.721
JC	0.901	0.817	0.788	0.754	0.883	0.867	0.862	0.820	0.684	0.616	0.965	0.964	0.792	0.748
EM	0.913	0.845	0.785	0.634	0.898	0.867	0.903	0.858	0.714	0.660	0.946	0.864	0.805	0.762
SPM	0.895	0.872	0.735	0.709	0.856	0.831	0.896	0.872	0.703	0.685	0.934	0.901	0.787	0.757
Node2vec	0.878	0.835	0.693	0.662	0.721	0.703	0.794	0.772	0.722	0.735	0.717	0.709	0.836	0.847
Line	0.736	0.718	0.621	0.607	0.551	0.536	0.609	0.584	0.642	0.612	0.650	0.649	0.750	0.724
AdaSim	0.849	0.841	0.727	0.718	0.863	0.851	0.774	0.723	0.766	0.762	0.743	0.740	0.799	0.795
CTDNE	0.907	0.918	0.893	0.897	0.843	0.837	0.841	0.846	0.818	0.825	0.729	0.691	0.837	0.820
DySAT	0.947	0.918	0.715	0.709	0.878	0.855	0.817	0.798	0.821	0.803	0.959	0.951	0.846	0.822
TGAT	0.921	0.906	**0.959**	**0.927**	0.786	0.774	0.905	0.882	0.878	0.831	0.719	0.691	0.872	0.676
SE-GRU	0.963	0.948	0.951	0.914	0.831	0.802	0.973	0.898	0.894	0.832	0.933	0.927	0.906	0.828
TREND	0.935	0.653	0.682	0.537	0.627	0.682	0.769	0.741	0.724	0.643	0.811	0.762	0.863	0.844
DMAB	0.958	0.746	0.764	0.797	0.736	0.613	0.912	0.829	0.919	0.862	0.974	0.841	**0.933**	0.829
COM	**0.989**	**0.982**	**0.959**	0.923	**0.953**	**0.946**	**0.986**	**0.982**	**0.923**	**0.893**	**0.979**	**0.976**	0.878	**0.863**

Datasets	Bitcoin-OTC		Slashdot-reply		Wiki-elec		Wiki-talk		Flickr		DBLP		Prosper	
Metrics	***AUC***	***AP***	***AUC***	***AP***	***AUC***	***AP***	***AUC***	***AP***	***AUC***	***AP***	***AUC***	***AP***	***AUC***	***AP***
CN	0.718	0.717	0.856	0.849	0.780	0.776	0.761	0.760	0.742	0.732	0.873	0.867	0.821	0.784
JC	0.742	0.693	0.757	0.755	0.844	0.825	0.687	0.639	0.712	0.682	0.609	0.615	0.746	0.783
EM	0.751	0.745	0.776	0.762	0.867	0.839	0.773	0.744	0.779	0.742	0.607	0.714	0.848	0.603
SPM	0.710	0.683	0.769	0.742	0.842	0.815	0.756	0.748	0.723	0.711	0.885	0.646	0.752	0.628
Node2vec	0.755	0.693	0.794	0.656	0.601	0.737	0.783	0.641	0.807	0.784	0.688	0.593	0.740	0.700
Line	0.726	0.699	0.782	0.771	0.625	0.619	0.766	0.748	0.772	0.756	0.642	0.642	0.747	0.656
AdaSim	0.737	0.709	0.825	0.805	0.718	0.724	0.836	0.802	0.835	0.816	0.868	0.685	0.871	0.708
CTDNE	0.807	0.776	0.891	0.872	0.826	0.726	0.905	0.881	0.857	0.845	0.913	0.798	0.823	0.799
DySAT	0.856	0.829	0.920	0.913	0.816	0.804	**0.937**	0.892	0.921	0.909	0.861	**0.885**	**0.903**	0.826
TGAT	0.823	0.808	0.942	0.924	0.953	**0.956**	0.916	0.901	0.924	0.911	0.825	0.828	0.843	0.736
SE-GRU	0.876	0.866	0.949	0.897	0.904	0.891	0.908	0.895	0.949	0.926	0.904	0.708	0.746	0.713
TREND	0.692	0.687	0.785	0.685	0.723	0.609	0.748	0.737	0.884	0.624	0.702	0.875	0.700	0.835
DMAB	**0.909**	0.671	0.871	0.739	0.782	0.602	0.803	0.664	0.708	0.715	0.852	0.761	0.869	0.779
COM	0.885	**0.893**	**0.956**	**0.932**	**0.956**	0.938	0.921	**0.905**	**0.968**	**0.937**	**0.934**	0.869	0.883	**0.874**

The best-performing methods are bold.

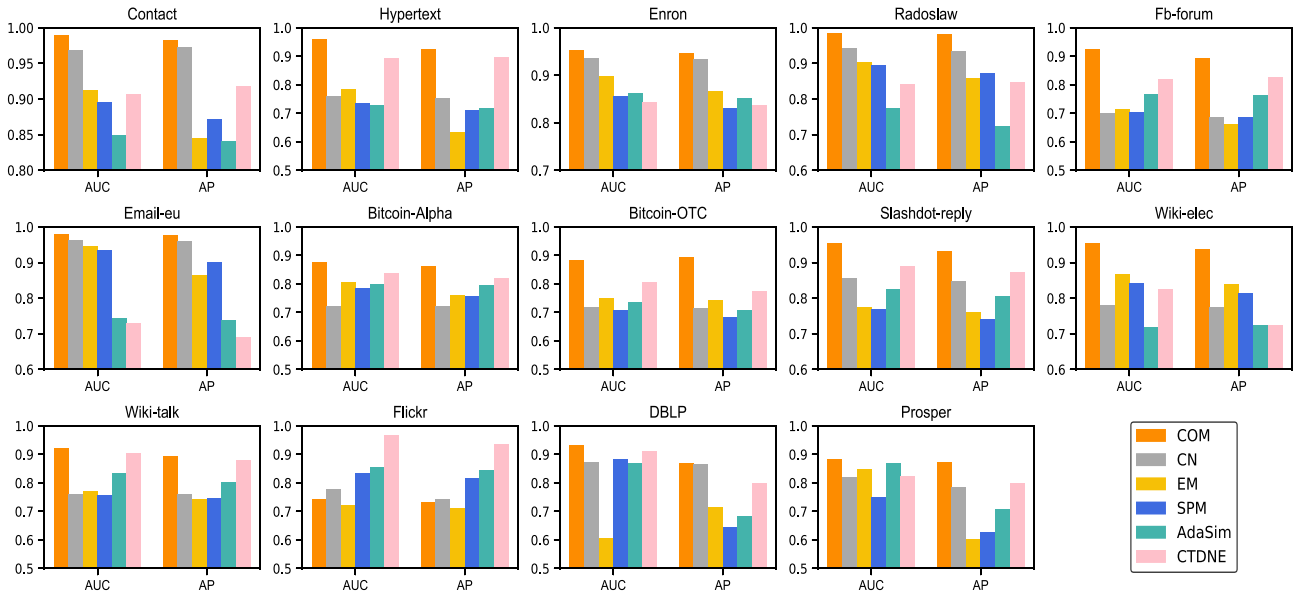

Fig. 4. AUC and AP results for models CN, EM, SPM, AdaSim, CTDNE, and COM on all datasets.

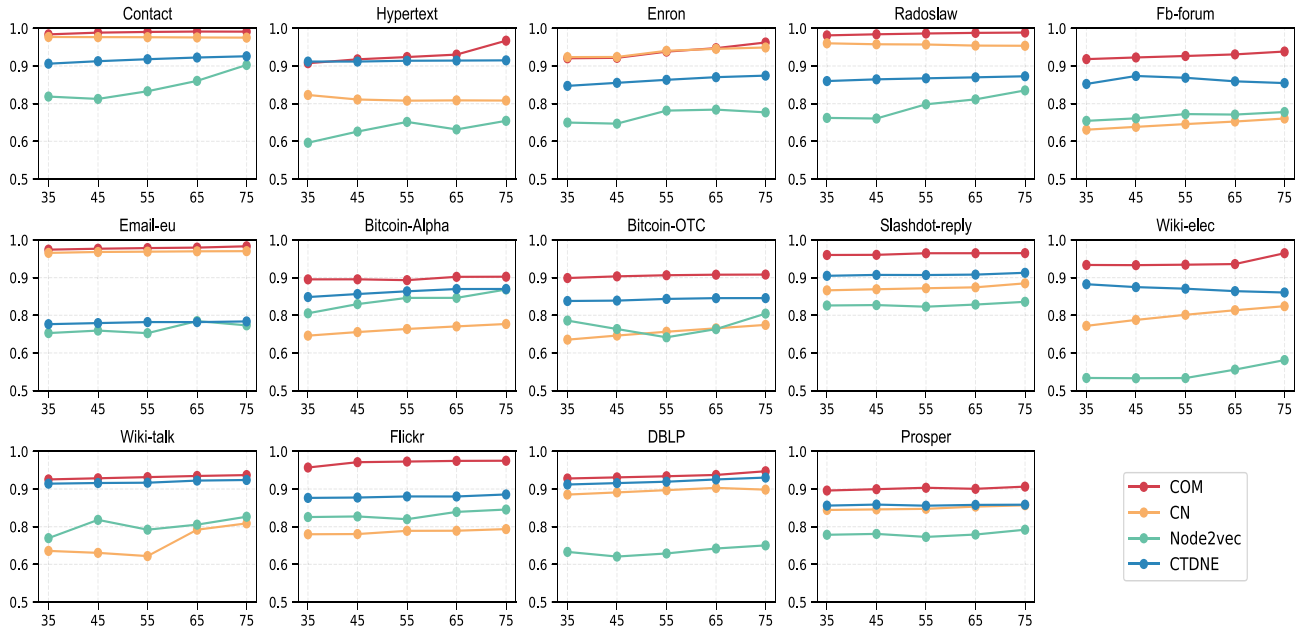

Fig. 5. AUC comparison was performed on all datasets of CN, Node2vec, CTDNE and COM using different percentage training links. For each dataset, 35%, 45%, 55%, 65%, and 75% of all links in networks were used as training sets.

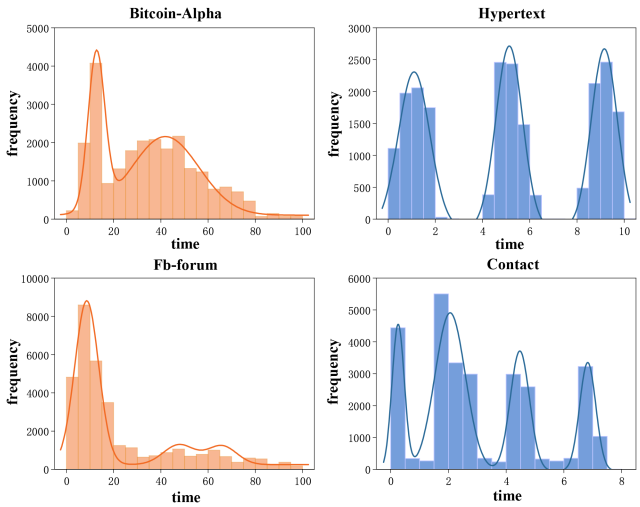

Fig. 6. Network growth pattern of different datasets. We normalize the time of the network to start from 0. The horizontal axis represents the evolution time of the network, and the vertical axis represents the frequency of edges appearing during the corresponding time period.

Additionally, we conducted a comprehensive analysis of the model's performance across different datasets. To provide further insights into these differences, we visualized the data distribution of edges in various networks, as shown in Fig. 6, and fitted them with curves. Notably, networks Contact and Hypertext, depicted in blue, showcased superior performance on our proposed COM model, while networks Bitcoin-Alpha and Fb-forum, represented in orange, exhibited relatively poor performance. The figure reveals that the evolution and growth pattern of the network that performs well in the model shows regularity. Conversely, networks with lower performance display erratic distribution and higher volatility, with noticeably slower growth rates in later stages. These observations further confirm the effectiveness of the COM model in capturing information pertaining to the growth and evolution patterns of dynamic networks.

C. Embedding Analysis

We use binary operations to combine node representations into edge representations and compare the performance of different methods. Fig. 7 shows that the effectiveness of each method varies across datasets. The experimental results show that the concatenation operation outperforms other edge representation methods on most datasets, indicating its effectiveness in preserving structural information. In Radoslaw, Email-eu, and Wiki-talk, the Mean operation performs more efficiently, highlighting its suitability for certain types of network structures. Meanwhile, the L2 operation exhibits poor performance on most datasets, especially on Bitcoin-Alpha and Bitcoin-OTC, suggesting a substantial loss of distinctive information. Based on these results, it can be concluded that other binary operations lead to information loss when compared to concatenation. Thus, we selected concatenation as the binary operation for our model. Notably, the performance of different methods on the Forum, Bitcoin-Alpha datasets shows significant variability, possibly due to the network's sparsity and the slow growth of the network in the later stage as shown in the network growth pattern of different datasets in Fig. 6.

D. Statistical Test

In this section, we apply the Friedman test and post-hoc Bonferroni-Dunn test to examine whether the performance of COM has statistical significance [54]. The Friedman test is used

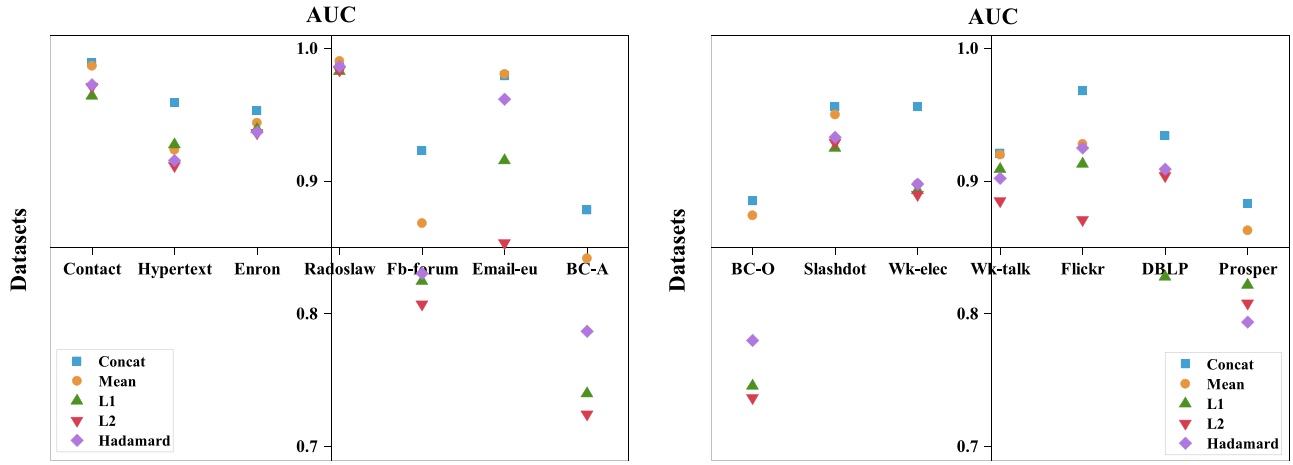

Fig. 7. Different edge representation methods test in COM across all datasets, measured by AUC.

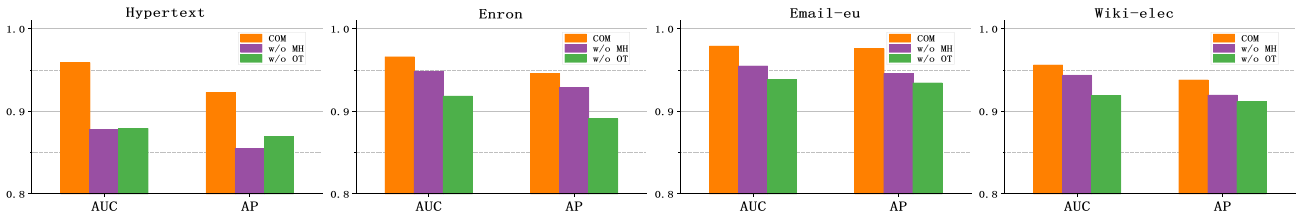

Fig. 8. Comparative analysis of AUC and AP results. Evaluating the impact of only considers optimal temporal information propagation (*w/o OT*) and spatial structure information (*w/o MH*) on COM performance, respectively.

to check the outperformance of COM is statistically significant, with the null hypothesis being that all models perform equally. Subsequently, the Bonferroni-Dunn test is employed to further investigate whether the performance of the focused model (COM, in our case) significantly differs from other models after rejecting the null hypothesis of the Friedman test. Both tests are based on the ranks of models, where the best-performing model is ranked 1, the second-best is ranked 2, and so forth.

The tests are conducted among the four well-performing models: COM, CTDNE, EM, and AdaSim. The ranks of these models are determined based on their performance on each dataset, which is measured using specific metrics. For instance, considering AUC as the evaluation metric, on the Contact dataset, the ranks of COM, CTDNE, EM, and AdaSim are 1, 3, 2, and 4 respectively. Friedman and Bonferroni-Dunn tests are performed based on these ranks and the result is shown in Fig. 9. The significance level α is set to 0.05.

As observed in the results, the p-value $1.8e-5$ of the Friedman test is significantly less than α, indicating the rejection of the null hypothesis that these four models have the same performance. Therefore, the Bonferroni-Dunn test is needed for further comparison. As depicted, the average rankings of COM, CTDNE, EM, and AdaSim are 1.000, 2.583, 2.916, and 3.500 respectively. All other models have average ranks greater than COM's average rank, with differences exceeding the critical difference (CD) value of 1.26174. In the Bonferroni-Dunn test, any model differing from the focused model by at least CD is considered significantly different in performance. Hence, the test results indicate that COM significantly outperforms other models in terms of prediction accuracy. In summary, these results verify that the superior performance of COM over baseline models holds statistical significance, which is proof of the effectiveness of COM from a statistical perspective.

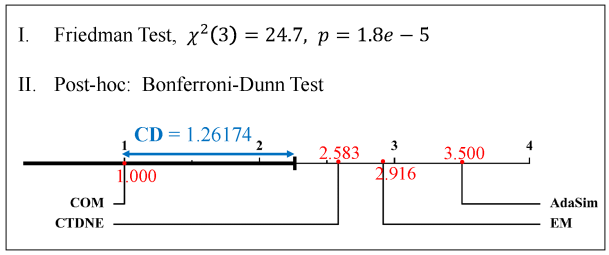

Fig. 9. Results of the statistical test. Step I shows the result of the Friedman test, including χ^2 statistics and the corresponding p-value. Meanwhile, step II illustrates the critical difference diagram derived from the Bonferroni-Dunn test, where the critical difference (CD) is marked in blue, and the average rank of each model is highlighted in red.

E. Ablation Study

In Fig. 8, we present the results of ablation studies conducted on the Hypertext, Enron, Email-eu and Wiki-elec in both temporal and spatial dimensions to examine the impact of temporal-biased and spatial-biased sampling strategy on dynamic link

prediction. Specifically, we additionally demonstrate the ablation test in terms of discarding the temporal validity and shortest distance of information transmission based on optimal transmission when extracting spatial structure information (denoted as *w/o OT*), and removing the extraction of spatial structure based on Metropolis Hastings when sampling temporal-biased walks (denoted as *w/o MH*). Our findings indicate that considering only temporal or spatial information alone leads to a decrease in link prediction performance. This highlights the effectiveness of our spatiotemporal biased walking sampler, and provides valuable insights into the intricate interplay between temporal dynamics and spatial structure in complex networks. Furthermore, when the optimal transmission of information between nodes is not considered in sampling, the prediction performance decreases significantly, demonstrating the importance of our proposed temporal-biased strategy in reducing the loss of information propagation in the network.

VI. Conclusion

In this paper, we introduce a dynamic link prediction framework based on distribution and random walks named COM. The framework employs the optimal transport theory to address temporal information loss in node information propagation by using the Wasserstein distance to reveal latent topological dependencies. Effectively integrating both local and global topology is crucial for improving link prediction performance. To achieve this, we leverage the Metropolis-Hastings algorithm to extract high-order spatial structures surrounding target links, ensuring correlation with the global evolution distribution while mitigating topological sampling bias toward hub nodes. And the encoding model is used to obtain the final node feature representation. Experiments on 14 datasets from different fields show that COM outperforms a wide range of methods in both heuristic and graph neural networks methods.

In future work, we aim to enhance our model's performance by integrating external information, such as text data and user attributes. Moreover, developing a multi-task link prediction model that can predict different types of links simultaneously will provide more comprehensive insights into network dynamics and structure.

References

[1] P.-Y. Chen, "Academic social networks and collaboration patterns," *Library Hi Tech*, vol. 38, no. 2, pp. 293–307, 2020.

[2] W. Wang et al., "User-context collaboration and tensor factorization for GNN-based social recommendation," *IEEE Trans. Netw. Sci. Eng.*, vol. 10, no. 6, pp. 3320–3330, Nov.–Dec. 2023.

[3] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, "Traffic flow prediction with Big Data: A deep learning approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 865–873, Apr. 2015.

[4] T. Pourhabibi, K.-L. Ong, B. H. Kam, and Y. L. Boo, "Fraud detection: A systematic literature review of graph-based anomaly detection approaches," *Decis. Support Syst.*, vol. 133, 2020, Art. no. 113303.

[5] R. Zhang, Q. Wang, Q. Yang, and W. Wei, "Temporal link prediction via adjusted sigmoid function and 2-simplex structure," *Sci. Reports*, vol. 12, no. 1, 2022, Art. no. 16585.

[6] L. Zou, X.-X. Zhan, J. Sun, A. Hanjalic, and H. Wang, "Temporal network prediction and interpretation," *IEEE Trans. Netw. Sci. Eng.*, vol. 9, no. 3, pp. 1215–1224, May–Jun. 2021.

[7] S. S. Singh, V. Srivastava, A. Kumar, S. Tiwari, D. Singh, and H.-N. Lee, "Social network analysis: A survey on measure, structure, language information analysis, privacy, and applications," *ACM Trans. Asian Low-Resource Lang. Inf. Process.*, vol. 22, no. 5, pp. 1–47, 2023.

[8] S. S. Singh, D. Srivastva, M. Verma, and J. Singh, "Influence maximization frameworks, performance, challenges and directions on social network: A theoretical study," *J. King Saud Univ.- Comput. Inf. Sci.*, vol. 34, no. 9, pp. 7570–7603, 2022.

[9] S. S. Singh, A. Kumar, K. Singh, and B. Biswas, "C2IM: Community based context-aware influence maximization in social networks," *Physica A: Stat. Mech. Appl.*, vol. 514, pp. 796–818, 2019.

[10] H. Zhang, D. T. Nguyen, H. Zhang, and M. T. Thai, "Least cost influence maximization across multiple social networks," *IEEE/ACM Trans. Netw.*, vol. 24, no. 2, pp. 929–939, Apr. 2016.

[11] Y. Song, D. Lai, Z. Chong, and Z. Pan, "Dynamic network embedding by time-relaxed temporal random walk," in *Proc. Neural Inf. Process. 28th Int. Conf.*, Sanur, Bali, Indonesia: Springer, Dec. 2021, pp. 426–437.

[12] L. Qu, H. Zhu, Q. Duan, and Y. Shi, "Continuous-time link prediction via temporal dependent graph neural network," in *Proc. Web Conf.*, 2020, pp. 3026–3032.

[13] C. Liu, S. Yu, Y. Huang, and Z.-K. Zhang, "Effective model integration algorithm for improving link and sign prediction in complex networks," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 3, pp. 2613–2624, Jul.–Sep. 2021.

[14] J. Chen and R. Ying, "TempME: Towards the explainability of temporal graph neural networks via motif discovery," in *Proc. Adv. Neural Inf. Process. Syst.*, 2023, vol. 36, pp. 29005–29028.

[15] G. H. Nguyen, J. B. Lee, R. A. Rossi, N. K. Ahmed, E. Koh, and S. Kim, "Continuous-time dynamic network embeddings," in *Proc. Companion Web Conf.* 2018, pp. 969–976.

[16] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon, "Network motifs: Simple building blocks of complex networks," *Science*, vol. 298, no. 5594, pp. 824–827, 2002.

[17] I. Iacopini, G. Petri, A. Barrat, and V. Latora, "Simplicial models of social contagion," *Nature Commun.*, vol. 10, no. 1, 2019, Art. no. 2485.

[18] C. Villani, "Topics in optimal transportation," *Amer. Math. Soc.*, 2021, vol. 58.

[19] Y. Rubner, C. Tomasi, and L. J. Guibas, "The Earth mover's distance as a metric for image retrieval," *Int. J. Comput. Vis.*, vol. 40, no. 2, 2000, Art. no. 99.

[20] X. Yao, Y. Shao, B. Cui, and L. Chen, "UniNet: Scalable network representation learning with metropolis-hastings sampling," in *Proc. IEEE 37th Int. Conf. Data Eng.*, 2021, pp. 516–527.

[21] L. Katz, "A new status index derived from sociometric analysis," *Psychometrika*, vol. 18, no. 1, pp. 39–43, 1953.

[22] L. Lü, C.-H. Jin, and T. Zhou, "Similarity index based on local paths for link prediction of complex networks," *Phys. Rev. E*, vol. 80, no. 4, 2009, Art. no. 046122.

[23] H.-H. Chen, L. Gou, X. Zhang, and C. L. Giles, "Discovering missing links in networks using vertex similarity measures," in *Proc. 27th Annu. ACM Symp. Appl. Comput.*, 2012, pp. 138–143.

[24] Y. Sun, J. Han, X. Yan, P. S. Yu, and T. Wu, "PathSim: Meta path-based top-K similarity search in heterogeneous information networks," *Proc. VLDB Endowment*, vol. 4, no. 11, pp. 992–1003, 2011.

[25] S. S. Singh, K. Singh, A. Kumar, and B. Biswas, "MIM2: Multiple influence maximization across multiple social networks," *Physica A: Stat. Mech. Appl.*, vol. 526, 2019, Art. no. 120902.

[26] S. S. Singh, A. Kumar, K. Singh, and B. Biswas, "LAPSO-IM: A learning-based influence maximization approach for social networks," *Appl. Soft Comput.*, vol. 82, 2019, Art. no. 105554.

[27] S. R. Mandala, S. R. Kumara, C. R. Rao, and R. Albert, "Clustering social networks using ant colony optimization," *Oper. Res.*, vol. 13, pp. 47–65, 2013.

[28] S. S. Singh, A. Kumar, K. Singh, and B. Biswas, "IM-SSO: Maximizing influence in social networks using social spider optimization," *Concurr. Comput.: Pract. Exp.*, vol. 32, no. 2, 2020, Art. no. e5421.

[29] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proc. 22nd ACM Int. Conf. Knowl. Discov. data mining*, 2016, pp. 855–864.

[30] B. Perozzi, R. Al-Rfou, and S. Skiena, "DeepWalk: Online learning of social representations," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discov. data mining*, 2014, pp. 701–710.

[31] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "Line: Large-scale information network embedding," in *Proc. 24th Int. Conf. World Wide Web*, 2015, pp. 1067–1077.

[32] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv:1609.02907*.
[33] P. Velickovic et al., "Graph attention networks," *Stat*, vol. 1050, no. 20, pp. 10–48550, 2017.
[34] J. Chen, X. Wang, and X. Xu, "GC-LSTM: Graph convolution embedded LSTM for dynamic network link prediction," *Appl. Intell.*, vol. 52, no. 7, pp. 7513–7528, 2022.
[35] M. E. Newman, "The structure and function of complex networks," *SIAM Rev.*, vol. 45, no. 2, pp. 167–256, 2003.
[36] D. Liben-Nowell and J. Kleinberg, "The link prediction problem for social networks," in *Proc. 12th Int. Conf. Inf. Knowl. Manage.*, 2003, pp. 556–559.
[37] L. Yin, H. Zheng, T. Bian, and Y. Deng, "An evidential link prediction method and link predictability based on Shannon entropy," *Physica A: Stat. Mechan. Appl.*, vol. 482, pp. 699–712, 2017.
[38] L. Lü, L. Pan, T. Zhou, Y.-C. Zhang, and H. E. Stanley, "Toward link predictability of complex networks," *Proc. Nat. Acad. Sci.*, vol. 112, no. 8, pp. 2325–2330, 2015.
[39] J. Wu, L. He, T. Jia, and L. Tao, "Temporal link prediction based on node dynamics," *Chaos, Solitons Fract.*, vol. 170, 2023, Art. no. 113402.
[40] C. Zhang, K.-K. Shang, and J. Qiao, "Adaptive similarity function with structural features of network embedding for missing link prediction," *Complexity*, vol. 2021, pp. 1–15, 2021.
[41] J. Park and A.-L. Barabási, "Distribution of node characteristics in complex networks," *Proc. Nat. Acad. Sci.*, vol. 104, no. 46, pp. 17916–17920, 2007.
[42] T. Zhou, Y.-L. Lee, and G. Wang, "Experimental analyses on 2-hop-based and 3-hop-based link prediction algorithms," *Physica A: Stat. Mech. Appl.*, vol. 564, 2021, Art. no. 125532.
[43] S. Chib and E. Greenberg, "Understanding the metropolis-hastings algorithm," *Amer. Statistician*, vol. 49, no. 4, pp. 327–335, 1995.
[44] J. M. Kleinberg, "Navigation in a small world," *Nature*, vol. 406, no. 6798, pp. 845–845, 2000.
[45] Y. Goldberg and O. Levy, "Word2vec explained: Deriving mikolov's negative-sampling word-embedding method," 2014, *arXiv:1402.3722*.
[46] D. Karani, "Introduction to word embedding and word2vec," *Towards Data Sci.*, vol. 1, 2018.
[47] H. Wang, W. Hu, Z. Qiu, and B. Du, "Nodes' evolution diversity and link prediction in social networks," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 10, pp. 2263–2274, Oct. 2017.
[48] S. Robertson, "A new interpretation of average precision," in *Proc. 31st Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. retrieval*, 2008, pp. 689–690.
[49] A. Sankar, Y. Wu, L. Gou, W. Zhang, and H. Yang, "DySAT: Deep neural representation learning on dynamic graphs via self-attention networks," in *Proc. 13th Int. Conf. web search data mining*, 2020, pp. 519–527.
[50] D. Xu, C. Ruan, E. Korpeoglu, S. Kumar, and K. Achan, "Inductive representation learning on temporal graphs," 2020, *arXiv:2002.07962*.
[51] Y. Yin, Y. Wu, X. Yang, W. Zhang, and X. Yuan, "SE-GRU: Structure embedded gated recurrent unit neural networks for temporal link prediction," *IEEE Trans. Netw. Sci. Eng.*, vol. 9, no. 4, pp. 2495–2509, july.–Aug. 2022.
[52] Z. Wen and Y. Fang, "Trend: Temporal event and node dynamics for graph representation learning," in *Proc. ACM Web Conf.* 2022, pp. 1159–1169.
[53] L. Lü and T. Zhou, "Link prediction in complex networks: A survey," *Physica A: Stat. Mech. Appl.*, vol. 390, no. 6, pp. 1150–1170, 2011.
[54] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, 2006.

Ruizhi Zhang is currently working toward the Ph.D. degree with the School of Mathematical Sciences, Beihang University, Beijing, China. Her current research interests include complex networks, link prediction and graph representation learning.

Wei Wei received the Ph.D. degree in mathematics from the School of Mathematical Sciences, Peking University, Beijing, China, in 2009. He is currently an Associate Professor with the School of Mathematical Sciences, Beihang University, Beijing, China. His research interests include dynamical system and complexity, complex networks, and artificial intelligence.

Qiming Yang received the B.Sc. degree in 2021, from Beihang University, Beijing, China, where he is currently working toward the master's degree with the School of Mathematical Sciences. His current research interests include complex networks, link prediction and graph representation learning.

Zhenyu Shi received the B.Sc. degree in 2019, from Beihang University, Beijing, where he is currently working toward the Ph.D. degree with the School of Mathematical Sciences and studying in Kyushu University as an Exchange Student. His current research interests include complex networks and evolutionary game theory.

Xiangnan Feng received the Ph.D. degree in mathematics from the School of Mathematical Sciences, Beihang University, Beijing, China, in 2020. He is currently a Postdoctoral Fellow with the Complexity Science Hub, Vienna, Austria. His research interests include complex networks, computing social science, and artificial intelligence.

Zhiming Zheng received the Ph.D. degree in mathematics from the School of Mathematical Sciences, Peking University, Beijing, China, in 1987. He is currently a Professor with the Institute of Artificial Intelligence, Beihang University, Beijing. He is also the Member of Chinese Academy of Sciences. His research interests include refined intelligence theory, information security, and complex information systems.