Complex & Intelligent Systems (2024) 10:3763–3788
https://doi.org/10.1007/s40747-024-01352-z

ORIGINAL ARTICLE

Bi-DNE: bilayer evolutionary pattern preserved embedding for dynamic networks

Xu Gu[1] · Xiaoliang Chen[1,2] · Min Yang[3] · Yajun Du[1] · Mingwei Tang[1]

Received: 31 August 2023 / Accepted: 23 December 2023 / Published online: 23 February 2024

Abstract

Network embedding is a technique used to generate low-dimensional vectors representing each node in a network while maintaining the original topology and properties of the network. This technology enables a wide range of learning tasks, including node classification and link prediction. However, the current landscape of network embedding approaches predominantly revolves around static networks, neglecting the dynamic nature that characterizes real social networks. Dynamics at both the micro- and macrolevels are fundamental drivers of network evolution. Microlevel dynamics provide a detailed account of the network topology formation process, while macrolevel dynamics reveal the evolutionary trends of the network. Despite recent dynamic network embedding efforts, a few approaches accurately capture the evolution patterns of nodes at the microlevel or effectively preserve the crucial dynamics of both layers. Our study introduces a novel method for embedding networks, i.e., bilayer evolutionary pattern-preserving embedding for dynamic networks (Bi-DNE), that preserves the evolutionary patterns at both the micro- and macrolevels. The model utilizes strengthened triadic closure to represent the network structure formation process at the microlevel, while a dynamic equation constrains the network structure to adhere to the densification power-law evolution pattern at the macrolevel. The proposed Bi-DNE model exhibits significant performance improvements across a range of tasks, including link prediction, reconstruction, and temporal link analysis. These improvements are demonstrated through comprehensive experiments carried out on both simulated and real-world dynamic network datasets. The consistently superior results to those of the state-of-the-art methods provide empirical evidence for the effectiveness of Bi-DNE in capturing complex evolutionary patterns and learning high-quality node representations. These findings validate the methodological innovations presented in this work and mark valuable progress in the emerging field of dynamic network representation learning. Further exploration demonstrates that Bi-DNE is sensitive to the analysis task parameters, leading to a more accurate representation of the natural evolution process during dynamic network embedding.

Keywords Representation learning · Dynamic network embedding · Power law · Network analysis

Introduction

Primary motivations

Citizens have shifted from traditional media to online social networks (OSNs) for obtaining, exchanging, and sharing information. The vast amount of data collected on OSNs are invaluable for academics and economics, making these platforms truly remarkable. This situation has led to the creation of many machine learning algorithms that address different application tasks, such as community detection [1, 2], link prediction [3, 4], node classification [5, 6], data visualization [7], and user alignment [8]. The volume of available social network data and the reliance of methods on the quality of the input pose challenges for implementing machine learning algorithms [9, 10]. To overcome these obstacles, various representation learning techniques have been proposed.

Network embedding (NE) is a type of representation learning strategy that has become popular for solving social network tasks. NE is further divided into static network embedding (SNE) and dynamic network embedding (DNE).

✉ Xiaoliang Chen
chexiaol@iro.umontreal.ca

[1] School of Computer and Software Engineering, Xihua University, Chengdu 610039, People's Republic of China

[2] Department of Computer Science and Operations Research, University of Montreal, Montreal, QC H3C3J7, Canada

[3] College of Computer Science, Sichuan University, Chengdu 610065, People's Republic of China

SNE is commonly used in the OSN research and is employed to extract various user node features, such as text sentiments and user relationships in social networks. In SNE methods, the vertices and edges remain unchanged during the time evolution process. DSRE [11], an unsupervised method introduced by Xiao et al., is a cost-effective way to extract specific knowledge from the abundant unstructured text information available in OSNs. Its efficacy has been proven to facilitate efficient OSN research while minimizing experimental costs. However, real-world networks are dynamic and constantly evolving. OSNs, such as Facebook and microblogging sites, offer opportunities for users to meet new people and build lasting connections, resulting in the constant formation of new friendships. The constantly changing nature of network development poses new challenges for SNE problems. Therefore, representation learning algorithms that overlook these dynamic features face limitations in terms of accurately capturing the evolving characteristics of hidden nodes, thus impeding their effectiveness in various practical applications.

A DNE approach extends a conventional static network model to capture the evolving patterns within networks [12, 13]. Li et al. [14] employed matrix perturbation theory to iteratively update low-dimensional node vectors, considering the temporal dependencies between different network representations. Goyal et al. [15] utilized a deep autoencoder called DynGEM to capture the nonlinear information between neighboring nodes, enhancing the learning of first-order and second-order relationships. While these DNE methods have provided some insights into network evolution, they still fall short of discerning the distinctive characteristics of the various node types within a network. Moreover, there is a dearth of comprehensive knowledge surrounding the fundamental dynamic mechanisms that drive network structure shifts.

Previous research has demonstrated that network dynamics are heavily influenced by evolution, with triadic closure being one of the most important structures [16]. Closed triads are formed when all pairs of users have connections in subsequent snapshots. The significance of triadic closure cannot be underestimated in terms of the growth and progression of OSNs. Importantly, this mechanism is not exclusive to OSNs; it also applies to the natural patterns that are present in various types of networks. Figure 1 depicts the evolutionary patterns observed in the vertex and community structure of a local social network. i and j are initially unconnected users, and they establish a friendship with user $k$at time a. This pattern unfolds in three distinct phases. Initially, user k exhibits relatively low connectivity, indicating limited popularity. However, user k can its increase attention and effort by maintaining strong connections with these two friends. Subsequently, user k evaluates the relevance and similarity between themselves and users i and j. This evaluation guides user k's social strategy, determining whether it is advantageous to introduce their two friends to each other. In OSNs, communities are typically subgraphs with higher internal link densities than that of the overall network. The likelihood of forming connections within a community surpasses that of forming connections between different communities. In Fig. 1, i and j are not initially part of the same community, necessitating k's role in fostering connections across communities. As time passes and triadic closure occurs, users i and j may change their link structures. This can lead to significant dissimilarity in their low-dimensional representations. Consequently, the community structure that they inhabit also undergoes an alteration.

Although past studies have recognized the impacts of alterations on network dynamics and the significance of incorporating events such as triadic closure into the development of social networks, little research has adequately merged DNE with triadic closure to address problems within OSNs. Yang et al. [17] presented DNETC, a novel DNE model that combines DNE coordination and triadic closure to effectively capture network evolution patterns. Their approach employs an ensemble technique using vertex similarity and emphasizes two factors, vertex popularity and community structure, enhancing the learning of a discriminative low-dimensional network representation. By incorporating this feature-enhanced representation, the authors optimized the evolution process of triadic closure within the network. However, understanding network evolution requires the consideration of both the micro- and macrolevels. The microlevel captures the intricate formation process of the network structure, involving the gradual generation of new edges through strengthened ternary closure. It provides insights into the specific structure adopted by the network at a given time, explaining the evolution process of the network. In contrast, macrolevel dynamics focus on network size changes, which are measured by the quantity of edges. This dynamic evolution process exhibits discernible nonrandom patterns and aims to answer the question of "how many edges should be generated in total by the dynamic evolution process of the network at the microlevel at time t." Understanding the dynamics at both the micro- and macrolevels is crucial for comprehending the evolution of a network, as these factors significantly influence how networks grow and transform.

Innovation aspect

This study proposes bilayer evolutionary pattern-preserving embedding for dynamic networks (Bi-DNE), which is a novel approach for dynamic network representation learning that captures evolution patterns at both the micro- and macrolevels, thus addressing the challenges associated with network representation learning in dynamic networks. Bi-DNE offers methodological innovations in terms of network representation learning by integrating triadic closure and dense power laws. We use enhanced triadic closure

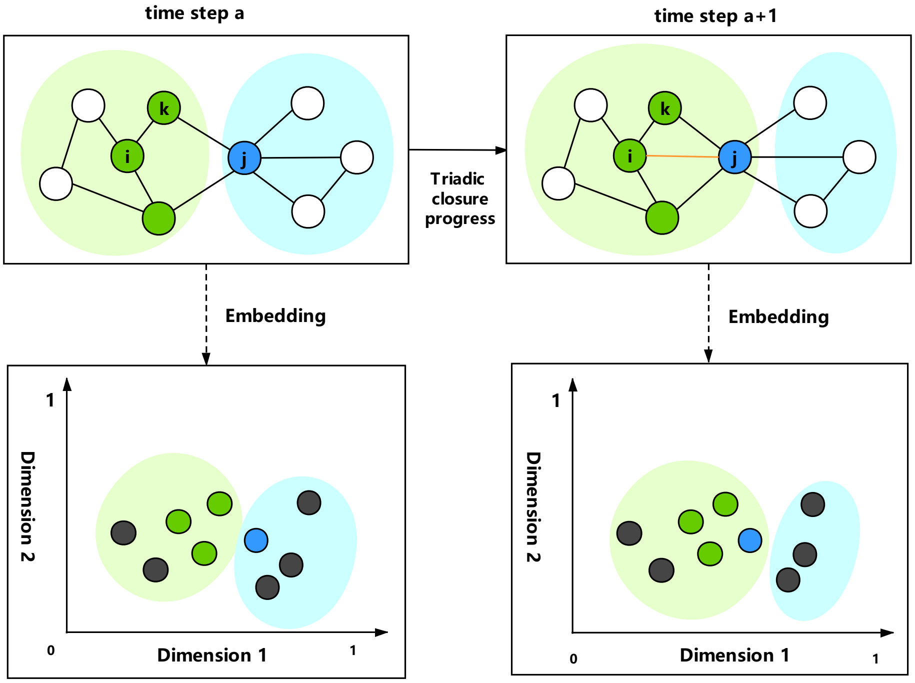


Fig. 1 Evolutionary patterns of vertices and the resulting community structures in localized social networks. The red solid line represents the addition of a new link at the subsequent moment. The circles with different colors represent the discrete vertices of two communities

to simulate how dynamic network structures change at the microlevel. At the macrolevel, we use dense power-law principles to impose constraints on how a network structure evolves. Our method effectively captures the dynamic characteristics of the network evolution process through low-dimensional representations, showcasing its methodological advancements. The visualized pipeline offers a clear and concise representation of the Bi-DNE method, as described in Fig. 2. The process begins with the input of a dynamic network and then divides it into two primary components that represent the micro- and macrolevel dynamics. It then learns the community structure of the network and finally converges to the Bi-DNE embedding process. The result is an evolved network representation that effectively demonstrates the fusion of micro- and macrodynamics during dynamic network embedding.

The Bi-DNE model is tested through various experiments that involve link prediction, link reconstruction, changed link prediction, and reconstruction tasks conducted on real datasets. Through these experiments, it is shown that the model can accurately preserve dynamic network evolution patterns. Our proposed model successfully captures high-quality feature representations from social network data. These findings underscore the methodological innovations and practical value of the Bi-DNE algorithm.

Contributions

The specific contributions of this study are listed below:

- Bi-DNE, a new approach that integrates micro- and macrodynamics to learn embeddings for dynamic networks, captures the natural evolution process of a network and improves network analysis tasks, is presented.
- Microlevel modeling is performed via enhanced triadic closure to capture granular network formation.
- Macrolevel modeling constrains the overall growth trends through a power law.
- Multiscale dynamics unification enables the evolution process to be more accurately represented.
- The proposed approach significantly outperforms the baselines on predictive tasks concerning time-evolving networks.
- This study advances the state-of-the-art representation learning research for dynamic networks.
- The sensitivity of the parameters in Bi-DNE is validated for the analysis task, demonstrating that the network representations learned by Bi-DNE are capable of addressing different types of variations.

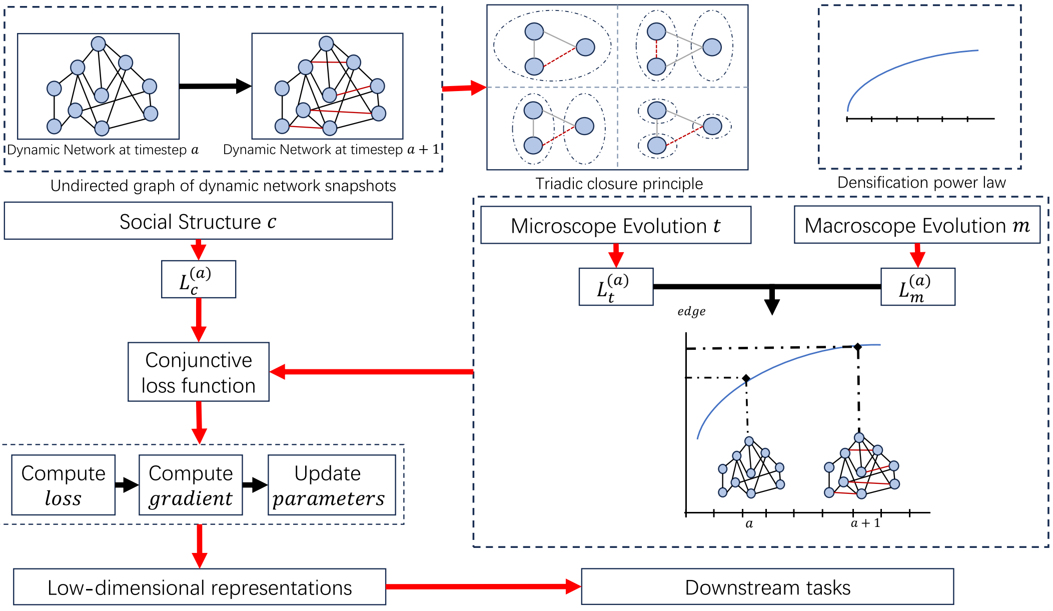


Fig. 2 A visualization pipeline of Bi-DNE

Manuscript structure

The rest of the paper is organized into the following sections. The section "Related work" provides a brief review of related work. The notation and definition of the DNE problem are given in the section "Problem definition". The section "Framework of Bi-DNE model" presents the Bi-DNE model in detail and the section "Experiment settings" presents the experimental setup. The section "Results" analyzes the experimental results and the section "Conclusion" concludes the paper.

Related work

In this section, we will discuss the various approaches to NE models, both static and dynamic, as explored in previous research. Abbreviations involved in the full text are enumerated in Table 1 for easier understanding.

SNE models

In the early stages, matrix factorization techniques, such as Laplacian eigenmapping [18] and locally linear embedding (LLE) [19], were commonly used. These approaches aim to obtain node representations of dimensionality k by computing eigenvectors. However, they often face challenges, including high computational costs and statistical performance limitations, when dealing with large-scale networks.

Deep learning has emerged as a prominent approach for network representation learning, leveraging neural network-based models to capture the intricate patterns and relationships within a network. DeepWalk [20] draws inspiration from the skip-gram word representation model [21] and learns latent representations. Node2Vec [22] extends DeepWalk by introducing a bias parameter that enhances the walking strategy by controlling the preference for depth- or breadth-first search. Line [23] focuses on capturing the second-order proximity information by considering the common neighbors of indirectly connected nodes, complementing the information acquired from the first-order proximity. SDNE [24] utilizes a deep neural network, modeling the highly nonlinear relationships between node representations, and uses the middle layer of a deep autoencoder to represent network nodes. GraRep [25] incorporates a specific relation matrix and reduces its dimensionality through singular value decomposition, resulting in a k-step network representation. These deep learning-based models have advanced the field of network representation learning by effectively capturing complex network structures.

The existing methods primarily focus on preserving the pairwise relationships between nodes while overlooking the valuable community structures in networks. Notable approaches such as BIGCLAM [26] and MemeRep [27] incorporate community structures into the network embedding (NE) process, but they do not consider the dynamic evolution characteristics of the target network. To obtain precise network representations, it is essential to include diverse and varied external information alongside the network topology. TADW [28] addresses this issue by incorporating the text features of nodes through matrix factorization, enhancing the strength and effectiveness of network node

Table 1 List of abbreviations

Abbreviation	Description	Abbreviation	Description
OSN	Online social network	NE	Network embedding
SNE	Static network embedding	DNE	Dynamic network embedding
Bi-DNE	Bi-layer dynamic network embedding (proposed model)	LR	The task of link reconstruction
LP	The task of link prediction	C.LR	Changed link reconstruction
C.LP	Changed link prediction	W/o mic	Removing the microlevel structure of Bi-DNE in the ablation experiment
W/o mac	Removing the macrolevel structure of Bi-DNE in the ablation experiment	W/o ss	Removing the social structure of Bi-DNE in the ablation experiment
W/o cs	Removing the conjunctive structure of Bi-DNE in the ablation experiment		

representations. CANE [29] is another noteworthy method that captures contextual information by considering node neighbors, improving the quality of the resulting network representations. These approaches provide insights into the integration of community structures, external information, and contextual dependencies for effective network representation learning. Li et al. [30] proposed using neural networks to learn node embeddings that capture air transportation network features. These embeddings were utilized to predict market influences for flight frequency optimization. In addition to network topology information, incorporating node attribute information into embeddings has shown promise. Li et al. [31] modeled the influence propagation process by jointly representing users in a social network structure and a multidimensional attribute space.

DNE models

Many methods have been suggested for preserving evolutionary trends in dynamic networks. DANE [14] and DHPE [32] employ matrix perturbation theory and generalized singular value decomposition to update node vector representations in dynamic networks while preserving high-order proximity. DNE [33] enables separate learning of node embeddings, enhancing the efficiency of vector representation updates. These methods contribute to preserving evolutionary patterns in dynamic networks by incorporating different techniques for updating node vector representations. Yu et al. [34] introduced NetWalk, a clique-based DNE approach for anomaly detection. Zhou et al. [35] proposed a method that incorporates dynamic information into a vector representation. A triadic closure simulation mechanism helps their model achieve excellent results. Ma et al. [36] presented a smoothing technique to maintain continuous community dynamics. This community-aware mechanism greatly improves the model fitting degree of real community change. CTDNE [37] is a dynamic network that captures time sequence information with continuous-time series represented as random walk sequences. The influence of past neighbors on neighbor formation for node vector representations is captured by HTNE [38] using the Hawkes process. However, the driving force behind the evolution of dynamic networks, especially the vertex evolution patterns, is often overlooked by these methods.

Lu et al. [39] proposed a novel approach for representing temporal networks by incorporating both micro- and macrodynamics, enabling the modeling of temporal dependencies and capturing microlevel network evolution. Additionally, they addressed macroscopic dynamics by defining a generic dynamical equation parameterized with network embeddings. This equation captured the inherent evolutionary patterns of the network and imposed constraints on the embeddings at higher structural levels. By considering both micro- and macrodynamics, their approach acknowledged the mutual influence between these dynamics and their impact on node embedding learning in temporal networks. The innovative combination of micro- and macrodynamics in their work has served as an inspiration for our own research.

Zhou et al. [40] introduced DynamicTriad, a novel representation learning method that preserves the structural information and evolution patterns of networks. They proposed using triples, comprising three vertices, to capture the fundamental triadic closure process that drives network formation and evolution. By incorporating this process into their model, DynamicTriad effectively captures network dynamics and learns vertex representation vectors at different time steps. This innovative approach provides insights into the temporal dynamics of networks and holds promise for network representation learning. Zhang et al. [41] designed a dynamic graph convolutional network to extract structural embeddings for DNE. Their method learns node representations via a leader-based fake labeling scheme. DynamiSE, proposed by Sun et al. [42], is a novel approach for link prediction that encodes both the dynamic and symbolic semantics of a network. The model integrates balance theory and ordinary differential equations in combination with node representation learning to build a deeper dynamic signed graph neural network. This network is designed to capture the complex symbolic semantics formed by two types of edges, resulting in a more comprehensive and accurate representation of the underlying network dynamics.

Yang et al. [17] proposed a novel network embedding approach for dynamic networks. This method considers vertex similarity and incorporates user popularity and community structure information to capture evolution patterns. By employing the ensemble idea and quantifying the probability of closing open triples, DNETC effectively captures the dynamics of network evolution. Additionally, the authors enhanced the discriminative power of low-dimensional network representations by utilizing community structure information as a higher order proximity measure for vertices. This optimization process yields improved dynamic network evolution predictions and aids in discovering macroscopic community structures. DNETC successfully captures diverse evolutionary patterns while preserving dynamic information in real networks. Du et al. [43] proposed ERM-ME to divide emotional communities based on user emotional preference. Then, the sentiment features are used to train the base classifiers, which are combined into a meta-classifier. Zhang et al. [44] introduce a novel method called DINE that learns vertex representations for dynamic networks. Their approach models users and items concurrently and integrates their representations in dynamic and static information networks.

To provide a more intuitive summary of previous studies, Tables 2 and 3 highlight the strengths and limitations of related SNE and DNE approaches, respectively. Early SNE approaches, such as DeepWalk [20], Node2Vec [22], and LINE [23], have been successful in modeling network topology. However, they do not consider the temporal nature and dynamics of real-world networks. Nevertheless, the initial DNE techniques that rely on matrix perturbation theory or factorization, such as CTDNE [37] and DANE [14], do not fully model the complex microlevel processes that underlie network formation over time. In the work of Yang et al. [17], community information and triadic closure rules are embedded, but the realistic constraint problem of dense power laws is not considered.

This paper proposes Bi-DNE, an improved model that uses an enhanced triadic closure process to simulate the formation of microlevel networks. This model takes important factors, such as node similarities and community structures. Additionally, macrolevel dynamics are governed by a power-law equation that controls the evolution scale of the target network. The fusion of both micro- and macrodynamics enables more precise characterizations of network evolution patterns. This approach can address the limitations of previous research and provide a better understanding of the network formation and evolution process.

Problem definition

In this section, we introduce the notations and definitions used throughout our work. The variable symbols and their meanings involved in this paper are listed in Table 4.

Definition 1 (*Undirected network*) A dynamic network is defined as a sequence of undirected graphs, represented as network snapshots $G^{(1)}, G^{(2)}, \ldots, G^{(T)}$, where T denotes the set of finite time steps. At each time step $a \in \{1, 2, \ldots, T\}$, the network state is denoted as $G^{(a)} = \left\{V, E^{(a)}, W^{(a)}\right\}$, where V is the set of nodes, $E^{(a)}$ is the set of edges, and each edge $e_{ij}^{(a)}$ represents the connection between nodes v_i and v_j with weight $w_{ij}^{(a)}$. The edges in $E^{(a)}$ are undirected in networks. This definition assumes that the network topology evolves over time rather than remaining static.

Definition 2 (*Dynamic network embedding*) Let $G = \{G^{(1)}, G^{(2)}, \ldots, G^{(T)}\}$ be a dynamic network, where T snapshots represent the network at different time steps. A positive integer d denotes the dimension of a low-dimensional representation space $\mathbb{R}^d$. The mapping $f^{(a)} : V^{(a)} \rightarrow \mathbb{R}^{(d)}$, for $a \in \{1, 2, \ldots, T\}$ refers to a function that embeds vertices $V^{(a)}$ in snapshot $G^{(a)}$ into d-dimensional vectors. Let $\boldsymbol{v}_i \in V$ denote vertex i and $\boldsymbol{u}_i^{(a)}$ its corresponding low-dimensional representation. The mapping function f is assumed to learn low-dimensional representations u that can express features of both network topology and dynamics.

Definition 3 (*Microevolution pattern*) Given a dynamic network, the microscopic microevolution pattern refers to the

Table 2 A detailed comparison of the SNE studies

Methods	Highlight	Limitation
LE [18], LLE [19]	1. Utilizing graph structure information without random walks efficiently to obtain context	1. Expensive eigendecomposition and hard to scale up
	2. Learned representations directly preserve key graph structure information	2. Learned representations have high symmetry, weak discriminative ability
DeepWalk [20]	1. Representing graph data by converting a continuous vector format	1. Relying on random walk process make some nodes being overlooked or poorly represented in large-scale graphs
	2. Applying a random walk to the graph structure	2. Lack of nodes' context, leading to a loss of crucial semantic information
Node2Vec [22]	1. Controllable random walk strategy to learn comprehensive node embeddings	1. Extremely high computational complexity for large-scale networks
	2. Sampling neighbor nodes with BIAS random walks for richer context	2. Insensitive to network changes, requiring retraining to adapt
	3. Adopting Skip-gram from word2vec to get better quality low-dimensional embeddings	3. The embedding space has redundancy and high symmetry, leading to weak discriminative ability
	4. Combining first-order and second-order random walk strategies to consider both local and global structures	4. Node attribute information can not be utilized directly
LINE [23]	1. Introducing proximities to model relationships between nodes	1. Needs predefined order, cannot adaptively learn network characteristics
	2. Employing contrastive loss functions for hierarchical representation learning	2. Only considers network topology, does not utilize node content information
	3. Scaling to large networks through negative sampling and approximate training	3. High complexity for second-order model in time and space
SDNE [24]	1. Minimizing network reconstruction cost with autoencoders	1. Insensitive to dynamic network changes
	2. Introducing adaptive weight hierarchical orthogonal constraints to reduce redundancy	2. Requires careful architecture design and hyperparameter tuning
GraRep [25]	1. Defining high-order transition probability matrices to model relationships between nodes at different orders	1. Lacks capability of inductive learning for unseen nodes and explicit mechanisms for handling dynamic graphs
	2. Using sparsity regularization to avoid overfitting and maintain efficiency	2. Embedding dimensionality increases linearly with number of orders
	3. Encoding higher order representations for more global structure	3. Assumes discrete node attributes which may lose information
AGA [30]	Manipulates gradients to ensure the budget limit and achieve efficient optimization dynamically	Limits the optimization possibilities in terms of network expansion or resource allocation
GDM [31]	1. Applying Gaussian propagation model to solve the social network problems	1.Lacks capability of model the temporal evolutions characteristic of real-world networks
	2. Constructing a multidimensional space model based on offset, motif and degree dimension	2. The simulation effect is easily affected by the network
TFIP [46]	Defining user sentiment power and clustering credibility for network embedding	Lack of propagation characteristics of different topics in the network

network structure's formation process through triads. An open triad $\left(v_i, v_j, v_k\right)$ evolves into a closed triad. The set $E_{+}^{(a)}=\left\{e_{ij} \mid e_{ij} \notin E^{(a)} \cap e_{ij} \in E^{(a+1)}\right\}$ is defined to represent the new edges generated during this process. Triadic closure between time steps is assumed to be one of the primary microlevel drivers of network evolution.

Definition 4 (*Macroevolution pattern*) Given a dynamic network, the macroevolution pattern at the macroscopic level represents the evolution process in which the network scale follows a certain distribution rule. The notation $A=\left\{n_{\mathrm{e}}^{(1)}, n_{\mathrm{e}}^{(2)}, \ldots, n_{\mathrm{e}}{}^{(T)}\right\}$ is defined to represent the set of edge numbers for T time steps of the dynamic network, where $n_{\mathrm{e}}^{(a)}$ represents the total number of edges in the network up to time point a. The evolution of network edge numbers is assumed to follow certain statistical distribution laws, reflecting macrolevel network evolution characteristics.

Table 3 A detailed comparison of the model proposed in this paper and other DNE studies

Method	Highlight	Limitation
DANE [14]	1. Capturing the co-evolution of network structure and node attributes 2. Developing LSTM-based sequence model to fuse temporal network snapshots 3. Combining optimizes structural and attribute representations	1. Applicable only to networks with discrete temporal snapshots 2. Suffers from error accumulation problem in long sequences 3. Lacks capability to handle unseen nodes in continuous-time dynamic networks
DHPE [32]	1. Utilizing high-order proximity in topological structure changes 2. Enforcing smooth embedding changes between consecutive time steps	1. High computational complexity in calculating high-order proximities 2. Constrained flexibility in modeling diverse topological structure changes
DNE [33]	Extending random walk process in skip-gram model to adapt to dynamic networks	Limited capability in handling more complex network dynamics like sudden changes
NetWalk [34]	1. Enhancing the quality of embeddings by improved negative sampling 2. Proposing a flexible random walk strategy to adaptively explore dynamic networks	1. Applicable only to discrete snapshot-based dynamic network data 2. Lacks theoretical analysis on the relationship between embedding space and anomalies
CTDNE [37]	1. Modeling continuous-time network dynamics directly without relying on discrete snapshots 2. Proposing a residual network model based on Hawkes processes to capture node interaction dynamics	1. Sensitive to hyperparameters like decay rates in Hawkes processes 2. Limited capability in handling more complex node interaction mechanisms
HTNE [38]	1. Considering the neighborhood formation process for network dynamics 2. Proposing maximum likelihood-based objective for representation learning	1. Relies heavily on attention distributions being accurate 2. Sensitive to perturbations and propagation errors
DynamicTriad [40]	1. Introducing DNE framework based on triadic closure process innovatively 2. Maximizing probabilities of triadic paths to learn features of network closure formation	1. Only models triadic closures, inadequate to capture complex network dynamics 2. Applicable only to discrete snapshot-based dynamic network data
DGCN [41]	Using a graph convolutional network for extracting and embedding node features	Node features have the problem of convergence and lack of discrimination
DynamiSE [42]	Encoding the dynamics and sign semantics of dynamic signed networks simultaneously	Requires a lot of pretraining and parameter tuning
ERM-ME [43]	Detecting emotion roles in social networks by fusing the information contained in different features	Be susceptible to hyperparameters and has poor scalability
DINE [44]	Modeling users and items simultaneously for fusing into dynamic and static information networks	Be vulnerable to hyperparameters and not scale well
DNETC [17]	Combining the popularity and proximity of vertices with community structures to enhance DNE structures	The constraint rules of triadic closure are not considered

Table 4 List of variables

Variable	Description	Variable	Description
$G(t)$	Network snapshot at time step t	V	Set of nodes in the network
$E(t)$	Set of edges in snapshot $G(t)$	u_i	Embedding vector for node i
Δn_e	Number of changed edges	n_e	Total number of edges
C	Community divided by the network	$\emptyset$	A collection with no elements

Definition 5 (*Community structure*) Communities in dynamic networks consist of nodes that are densely interconnected within each community while having sparse connections with nodes outside the community. Let $C^{(\alpha)} = \left\{C_1^{(a)}, C_2^{(a)}, \ldots, C_m^{(a)}\right\}$ represent the division of the network at time a into m communities. Then, we have $V = C_1^{(a)} \cup C_2^{(a)} \cup \ldots \cup C_m^{(a)}$, and for any pair of nodes (i, j) with $i \neq j$, it holds that $C_i^{(a)} \cap C_j^{(a)} = \emptyset$. The network is assumed to exhibit community structure characteristics, where nodes have dense connections within communities but sparse connections between communities.

Framework of Bi-DNE model

This section comprehensively introduces the proposed dynamic network representation learning model (Bi-DNE) and its details. Bi-DNE includes three components: multigranularity attribute embedding (the sections "Preservation of dynamic embedding through microscopic evolution", "Preservation of dynamic embedding through macroscopic evolution and "Microscopic and macroscopic representations of dynamic evolution"), attention-based structure embedding (the section "Preservation of dynamic embedding through community structure"), and an iterative training process (the section "Conjunctive model"). Among them, the multigranularity attribute embedding method comprises two essential processes: the micro-level enhanced triadic closure process, which models network structure formation (the section "Preservation of dynamic embedding through microscopic evolution"), and the macrolevel constraints that enforce a dense power-law evolution pattern through dynamic equations (the section "Preservation of dynamic embedding through macroscopic evolution"). Figure 3 illustrates the framework of Bi-DNE, providing informative details about its structure.

Preservation of dynamic embedding through microscopic evolution

In a social network, when an open triple arises at a certain time, certain users actively introduce their friends, thus transforming it into a closed triple. Conversely, other users opt to maintain the original social status of their friends, thereby displaying a decision distinction based on individual considerations and variations in social strategies. User considerations encompass factors, such as user popularity, similarity, and the community structure to which they belong. As the network evolves, its nodes continually form new connections through the triadic closure mechanism, taking place at various points in time. This evolutionary process is influenced by factors, such as node popularity, similarity between nodes, and the community in which the nodes reside.

Considering the aforementioned factors, we utilize a d-dimension vector $\boldsymbol{D}_{ijk}^{(a)}$ to assess the decision of user v_k

$$\boldsymbol{D}_{ijk}^{(a)} = \left[\frac{w_{ik}^{(a)}}{deg_k^{(a)}} * \left(\boldsymbol{u}_k^{(a)} - \boldsymbol{u}_i^{(a)}\right) + \frac{w_{jk}^{(a)}}{deg_k^{(a)}} * \left(\boldsymbol{u}_k^{(a)} - \boldsymbol{u}_j^{(a)}\right)\right] * \xi_{ij}^{(a)}, \tag{1}$$

where the edge weight $w_{ik}^{(a)}$ denotes the tie strength of v_i and v_k, $deg_k^{(a)}$ is the degree of v_k, and $\boldsymbol{u}_k^{(a)}$ denotes the embedding vector of v_k. $\xi_{ij}^{(a)}$ is defined as a function of community influence, that is

$$\xi_{ij}^{(a)} = \begin{cases} \alpha, & \text{if } (v_i^{(a)} \in C_p^{(a)}) \bigwedge (v_j^{(a)} \in C_q^{(a)}) \bigwedge (p \neq q), \\ 1 - \alpha, & \text{otherwise.} \end{cases} \tag{2}$$

Furthermore, to calculate the probability of an edge forming between users v_i and v_j under the influence of user v_k at time step $a + 1$, use the following formula:

$$P_{k \to i,j}^{(a)} = \frac{1}{1 + \exp\left(-\left\langle \boldsymbol{\theta}, \boldsymbol{D}_{ijk}^{(a)} \right\rangle\right)}, \tag{3}$$

where the vector $\boldsymbol{\theta}$ with d-dimensions represents the social strategy information extracted from latent vertices.

Consequently, the probability of creating a new connection follows the formula below:

$$P_{e_{ij_+}}^{(a)} = \sum_{\boldsymbol{\mu}_{i,j}^{(a)} \neq 0} \prod_{k \in CN_{i,j}^{(a)}} \left(P_{k \to i,j}^{(a)}\right)^{\mu_{k \to i,j}^{(a)}} * \left(1 - P_{k \to i,j}^{(a)}\right)^{\left(1 - \mu_{k \to i,j}^{(a)}\right)}. \tag{4}$$

The vector $\boldsymbol{\mu} i, j^{(a)} = (\mu k \to i, j^{(a)}) k \in \mathrm{CN} i, j^{(a)}$ indicates the state of an open triad involving nodes v_i and v_j at time step a. Specifically, $\mu_{k \to i,j}^{(a)} = 1$ if nodes v_i and v_j will form an edge at time step $a + 1$ under the influence of their common neighbor v_k. The set $\mathrm{CN} i, j^{(a)}$ contains all common neighbors of nodes v_i and v_j at time step a. The vector $\boldsymbol{\mu} i, j^{(a)}$ encodes which open triads will close due to triadic closure in the next time step. The probability of not generating edge e_{ij} at time $a + 1$ follows the formula below:

$$P_{e_{ij_-}}^{(a)} = \prod_{k \in CN_{i,j}^{(a)}} \left(1 - P_{k \to (i,j)}^{(a)}\right). \tag{5}$$

The loss function for the triadic closure process combines two user edge evolution patterns with the objective of mini-

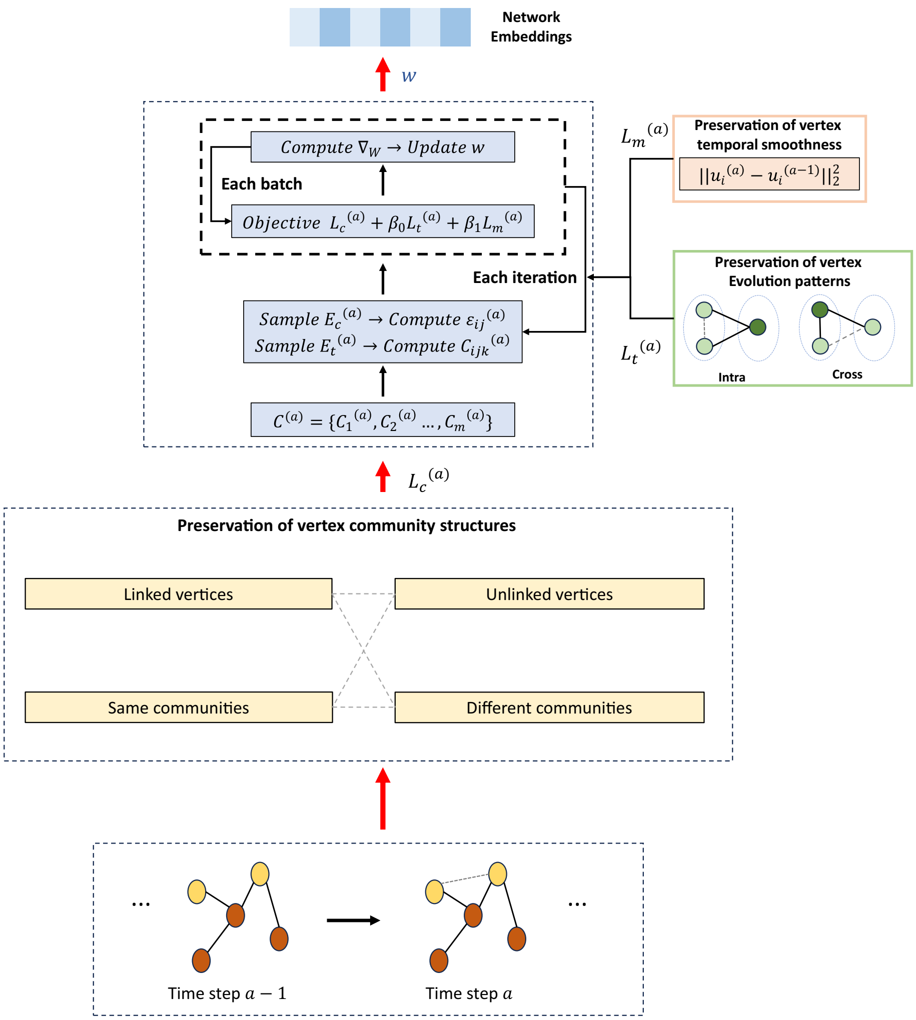

Network Embeddings
w
$Compute\ \nabla_W \rightarrow Update\ w$
Each batch
$Objective\ \ L_c{}^{(a)} + \beta_0 L_t{}^{(a)} + \beta_1 L_m{}^{(a)}$
Each iteration
$Sample\ E_c{}^{(a)} \rightarrow Compute\ \varepsilon_{ij}{}^{(a)}$
$Sample\ E_t{}^{(a)} \rightarrow Compute\ C_{ijk}{}^{(a)}$
$C^{(a)} = \{C_1{}^{(a)}, C_2{}^{(a)} \ldots, C_m{}^{(a)}\}$
$L_m{}^{(a)}$
Preservation of vertex temporal smoothness
$||u_i{}^{(a)} - u_i{}^{(a-1)}||_2^2$
Preservation of vertex Evolution patterns
$L_t{}^{(a)}$
Intra
Cross
$L_c{}^{(a)}$
Preservation of vertex community structures
Linked vertices
Unlinked vertices
Same communities
Different communities
...
Time step $a-1$
Time step a
...

Fig. 3 Schematic diagram of the Bi-DNE model. Bi-DNE retains the evolution pattern and community structure of the vertices as the microscopic expression during the embedding process and uses the dynamic equation to constrain the network structure to obey the densified power-law evolution pattern as the macroscopic expression

mizing negative log-likelihood, which is below [17]

$$L_t^{(a)} = -\sum_{(v_i, v_j)\in S_+^{(a)}} \log P_{e_{ij+}}^{(a)} - \sum_{(v_i, v_j)\in S_-^{(a)}} \log P_{e_{ij-}}^{(a)}, \tag{6}$$

where $S_+^{(a)} = (v_i, v_j) \mid e_{ij} \notin E^{(a)} \cap e_{ij} \in E^{(a+1)}$ represents the set of open triads composed of nodes v_i and v_j that become closed at time step $a+1$. In other words, $S_+^{(a)}$ contains node pairs (v_i, v_j) that do not have an edge e_{ij} in snapshot $E^{(a)}$, but gain an edge e_{ij} in snapshot $E^{(a+1)}$. $S_-^{(a)} = (v_i, v_j) \mid e_{ij} \notin E^{(a)} \cap e_{ij} \notin E^{(a+1)}$ indicates open triads that remain open, i.e., node pairs (v_i, v_j) that do not have an edge e_{ij} in snapshot $E^{(a)}$ and continue not to have an edge e_{ij} in snapshot $E^{(a+1)}$. The sets $S_+^{(a)}$ and $S_-^{(a)}$ categorize open triads based on whether they close or remain open between time steps a and $a+1$.

Preservation of dynamic embedding through macroscopic evolution

The ternary closure evolution process at the microlevel continuously drives network node topology changes, which are manifested in the continuous formation of new edges. At the macrolevel, the formation process is constrained by the network size, which determines the number of edges to be generated at a given time. Macrolevel dynamics encompass the evolution pattern of the network size, which typically follows a distinct distribution. This distribution allows the network size to be described by a dynamic equation. Incorporating this high-level structure into the network embedding process greatly enhances its effectiveness. Therefore, establishing a dynamic equation with a network embedding vector as a parameter establishes a connection between the network dynamics analysis and DNE tasks.

In a dynamic network, we use the notation $G^{(a)} = \left\{V, E^{(a)}, W^{(a)}\right\}$ at time a. Here, V represents the set of nodes, $E^{(a)}$ represents the set of edges, and $W^{(a)}$ represents the weights associated with the edges. The total number of nodes in the network is represented by $n^{(a)}$, while the total number of edges is represented by $n_{\mathrm{e}}^{(a)}$. The dense power-law model describes the network evolution process, stating that as the graph becomes denser over time. Based on this model, we define the average number of accessible neighbors for each node when attempting to establish a new edge with another node in a dynamic network

$$n_{ac} = \varphi \left(n^{(a)} - 1\right)^{\omega}, \tag{7}$$

where φ represents the linear sparse index and ω represents the dense power index. Typically, ω falls between 1 and 2. A value of 1 for the index a indicates a constant average number of degrees over a specified timeframe, while a value of 2 signifies a highly dense configuration.

The link rate $r(a)$ is crucial in developing network size, but its importance diminishes as the network progresses due to its temporal dynamics. This is because the majority of edges are formed in the early stages, and as the network becomes denser, the edge generation rate naturally decreases. The link rate $r(a)$ is also influenced by the underlying network topology. Since the network structure is not static but evolves with new edges, considering this aspect in network embedding is essential. Aiming at incorporating both the temporal and structural information, we propose a parameterized formulation for the network's link rate using the network dynamic finalization term a^{ϵ} and the node embedding $\mathcal{U}$, which can be expressed as follows:

$$r(a) = \frac{\frac{1}{n_{\mathrm{e}}^{(a)}} \sum_{e_{ij}^{(a)} \in E^{(a)}} \sigma \left(-\left\| \boldsymbol{u}_i^{(a)} - \boldsymbol{u}_j^{(a)} \right\|_2^2\right)}{a^{\epsilon}}, \tag{8}$$

where ϵ is network dynamic end item, $\sigma x = \exp(x) / (1 + \exp(x))$ is the sigmoid function. When network evolution is complete, the current network consists of $n^{(a)}$ interconnected nodes. At time a, each node v_i within the network endeavors to form new connections, specifically with other accessible neighbor nodes n_{ac}, utilizing a connection rate denoted as $r(a)$. Henceforth, we define the macrolevel network dynamics at time a as the quantity of newly established edges as follows:

$$\Delta n_{\mathrm{e}}^{\prime(a)} = n^{(a)} * r(a) * n_{ac}. \tag{9}$$

The integration of network structure into learned embeddings is crucial for capturing the inherent characteristics of a network. In Eq. (8), the numerator represents the maximum link rate of the network, which takes into account the temporal decay of node embeddings. This motivates the combination of network representation learning and the exploration of macroscopic evolutionary patterns in dynamic networks.

As the dynamic network evolves, $A = \left\{n_{\mathrm{e}}^{(1)},\ n_{\mathrm{e}}^{(2)},\ \ldots,\ n_{\mathrm{e}}{}^{(T)}\right\}$ represents the number of edges in the dynamic network for T time steps, where $n_{\mathrm{e}}^{(a)}$ is the total number of edges in the network at time up to a. Then, the true number of edges changed in the network at time a is

$$\Delta n_{\mathrm{e}}^{(a)} = n_{\mathrm{e}}^{(a+1)} - n_{\mathrm{e}}^{(a)}. \tag{10}$$

To capture the macrolevel evolution dynamics of the network, we aim to construct a loss function for the embedding process that preserves these dynamics. At time a, the loss function is determined by minimizing the sum of squared errors between the actual count of changed edges in the network, denoted as $\Delta n_{\mathrm{e}}^{(a)}$, and the predicted count of changed edges, denoted as $\Delta n_{\mathrm{e}}^{\prime(a)}$, which is based on the macroevolution pattern. The formulation is as follows:

$$L_m^{(a)} = \left(\Delta n_{\mathrm{e}}^{(a)} - \Delta n_{\mathrm{e}}^{\prime(a)}\right)^2. \tag{11}$$

Microscopic and macroscopic representations of dynamic evolution

This paper presents the Bi-DNE model, which integrates micro- and macroevolution dynamics of dynamic networks into DNE. Figure 4 shows a dynamic network diagram that reflects the model's idea, described as follows:

- Microlevel: From time step a to $a + 2$, new edges gradually evolve due to the triadic closure effect of nodes that are not connected but have common friends. Each change is described in detail, reflecting the dynamic characteristics of the network at the microlevel. For example, the triadic closure process of these three adjacent time steps generated a total of four new edges.
- Macrolevel: The number of edges does not change randomly but rather exhibits significant distribution characteristics. This macroevolution dynamic reveals the potential evolution patterns of dynamic networks, which can further constrain the learning of network node representations at a higher level. That is, networks gradually evolve through fine-grained changes and ultimately form a network with a certain size. For example, from time step a to $a+1$, the network generates three new edges through microevolution, while from time step $a + 1$ to $a + 2$, the network generates only one new edge through microevolution. Observing the parabolic trend in the graph shows that the network generates more edges in the early stages through triadic closures, but the growth rate gradually declines as time passes.

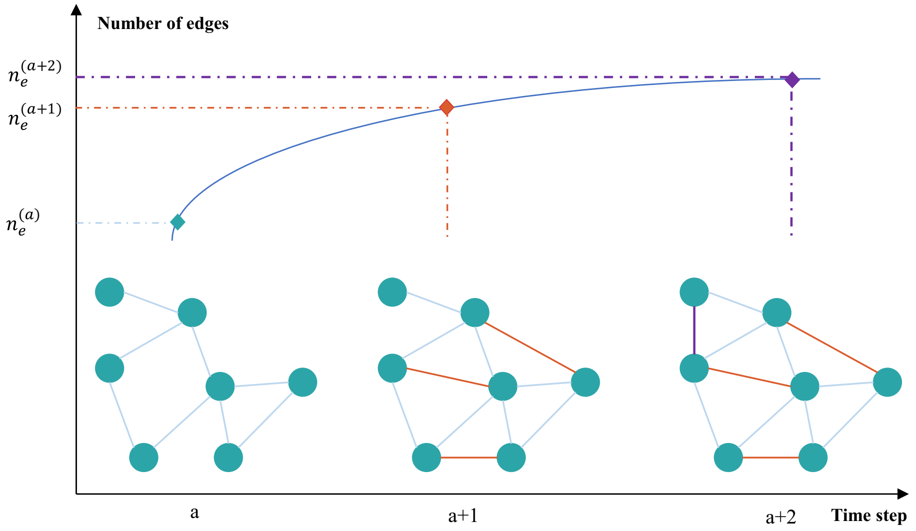


Fig. 4 An illustration of a dynamic network exhibits both micro- and macrolevel evolution patterns

- Micro- and macrolevels combined: The micro- and macrolevels serve as guiding principles for the network evolution and node embedding tasks. The triadic closure process that occurs among the nodes at the microlevel drives the continuous generation of edges. However, this effect is limited by the distribution characteristics at the macrolevel, which jointly determine how many new edges are generated at each time step.

Preservation of dynamic embedding through community structure

In dynamic networks, community structures are important characteristics. Social network groups are represented as data points in an embedding space. Even if they are not directly connected, groups tend to be closer together than individuals from different groups. Measuring the distances between communities helps keep the network structure intact.

Let $C^{(a)} = \left\{C_1^{(a)}, C_2^{(a)}, \ldots, C_m^{(a)}\right\}$ be the set of m community structures in dynamic network G at time step a. For any vertex pair $v_i^{(a)} \in C_p^{(a)}$ and $v_j^{(a)} \in C_q^{(a)}$ $(i \neq j)$, there are four possible edge subsets based on whether nodes have an edge e_{ij} and whether they belong to the same community $p = q$ or different communities $p \neq q$ [17]

- $S_1^{(a)} = \left\{(v_i, v_j) \mid e_{ij} \in E^{(a)} \bigwedge p = q\right\}$: Linked vertex pair in the same community. [17]
- $S_2^{(a)} = \left\{(v_i, v_j) \mid e_{ij} \in E^{(a)} \bigwedge p \neq q\right\}$: Linked vertex pair in different communities. [17]
- $S_3^{(a)} = \left\{(v_i, v_j) \mid e_{ij} \notin E^{(a)} \bigwedge p = q\right\}$: Unlinked vertex pair in the same community. [17]
- $S_4^{(a)} = \left\{(v_i, v_j) \mid e_{ij} \notin E^{(a)} \bigwedge p \neq q\right\}$: Unlinked vertex pair in different communities. [17]

The proximity between vertex pairs is measured separately for each of the four subsets. The loss function to preserve community structure is formally defined as

$$L_c^{(a)} = \sum_{(i,j)\in S^{(a)}} \left[\xi_{ij}^{(a)} * g^{(a)}(v_i, v_j) \left\| \boldsymbol{u}_i^{(a)} - \boldsymbol{u}_j^{(a)} \right\|_2^2 + \gamma \right]_+ . \tag{12}$$

Let set $S^{(a)} = S_1^{(a)} \cup S_2^{(a)} \cup S_3^{(a)} \cup S_4^{(a)}$ represent all vertex pairs in network G at time a. The Euclidean distance between embeddings of vertices v_i and v_j is denoted as $\left\| \boldsymbol{u}_i^{(a)} - \boldsymbol{u} j^{(a)} \right\| 2^2$. The notation $[x]+$ represents $\max(0, x)$ for a real number x. $\gamma \in \mathbb{R}^+$ is a margin hyperparameter. $\xi i j^{(a)}$ is defined in Eq. (2). The function $g^{(a)}(v_i, v_j)$ indicates whether vertices v_i and v_j are linked at time a

$$g^{(a)}(v_i, v_j) = \begin{cases} 1, & \text{if } (v_i, v_j) \in S_1^{(a)} \cup S_2^{(a)} \\ -1, & \text{otherwise,} \end{cases} \tag{13}$$

where it outputs 1 if v_i and v_j are linked, and -1 otherwise.

Conjunctive model

Dynamic evolution of networks occurs at both micro- and macrolevels and plays a crucial role in their development. We construct an overall objective function that incorporates three loss functions: $L_c^{(a)}$ (Eq. (12)) for community structure preservation, $L_t^{(a)}$ for microevolution preservation (Eq. (6)), and $L_m^{(a)}$ for macroevolution preservation (Eq. (11)). Minimizing these loss functions allows us to capture the intricate dynamics of dynamic networks and obtain meaningful representations for nodes in the network. The formulation is as follows:

$$L_{loss} = \sum_{a \in T} L_c^{(a)} + \beta_0 L_t^{(a)} + \beta_1 L_m^{(a)}, \tag{14}$$

where $\beta_0 \in [0, 1]$ is the weight of the microlevel network evolution pattern on the DNE constraint, and $\beta_1 \in [0, 1]$ is the weight of the macrolevel network evolution pattern on the DNE constraint.

In this study, the entire training procedure of the Bi-DNE approach discussed above is outlined as Algorithm 1.

Algorithm 1 Overall framework of Bi-DNE

```
Input: $G = \{G^{(1)}, G^{(2)}, \ldots, G^{(T)}\}$: Sequence of dynamic network snapshots. $\alpha, \eta, \gamma$ and $d$: all hyperparameter
Output: $\boldsymbol{U} = \{\boldsymbol{U}^{(1)}, \boldsymbol{U}^{(2)}, \ldots, \boldsymbol{U}^{(T)}\}$: low-dimensional representations in a dynamic network.
Initialize model parameters $W = \left\{\boldsymbol{\theta} \in \mathbb{R}^{1\times d}, \boldsymbol{U} \in \mathbb{R}^{|V|\times d}, \varphi, \omega, \varepsilon\right\}$ randomly.
For each time step $a \in T$, count the number of nodes $n^{(a)}$ and edges ${n_\mathrm{e}}^{(a)}$ in the network snapshot.
for each iteration do
    Sample $E_c^{(a)}$ for preserving community structure.
    Compute the community influence $\xi_{ij}^{(a)}$ for each instance of $E_c^{(a)}$.
    Sample $E_t^{(a)}$ for preserving evolution pattern.
    Compute each $C_{ijk}^{(a)}$ based on $C_{ijk}^{(a)} = 1 - \frac{P_{k\to i,j}^{(a)}}{1-\prod_{k*\in CN_{i,j}^{(a)}}\left(1-P_{k\to i,j}^{(a)}\right)}$.
    for each batch of training set do
        Compute loss $L_{\mathrm{loss}}$ based on Eq. (14).
        Compute gradients based on $\nabla_{\boldsymbol{W}} = \frac{\partial L^{(a)}}{\partial \boldsymbol{W}^{(n)}}$.
        Update parameters by the gradient based on $W^{(n+1)} \leftarrow W^{(n)}$
    end for
end for
```

Experiment settings

All methods were implemented using Python 3[1] The hyperparameters involved in this experiment and their meanings are enumerated in Table 5.

Dataset

In this study, the proposed model Bi-DNE is evaluated using four real-world dynamic networks, which are described as follows:

- **The fb-messages network** [45]: This dataset represents an online community of 1,899 students from the University of California, Irvine. It includes 59,835 messages sent between them from April to October 2004.
- **The ia-facebook network** [45]: This dataset shows the connections between 44,686 users over 24 months. A total of 735,439 links that were recorded between 2007 and 2008. Earlier data were not included due to lack of information.
- **The ia-contacts-Dublin network** [45]: This network consists of daily dynamic contacts collected during the Infectious Social Patterns event at the Science Gallery in Dublin, Ireland, during the ArtScience Exhibition. The data were collected at 20-s intervals. The network spans ten time steps, with each time step representing a 3-week period.
- **The ia-retweet-pol network** [45]: This dataset includes 470 Twitter users as vertices and 61,157 retweet relations as edges. It spans six time steps, with each step representing a three-week period.

Figure 5 presents more detailed statistics concerning the experimental data.

Baseline

To evaluate the effectiveness of the suggested model, we have chosen five baseline methods. These methods include three SNE techniques (DeepWalk, Node2vec, and LINE) and two DNE methods (TNE and DynamicTriad). To ensure fair comparisons in our experiments, we have carefully chosen the best settings for each of the baseline algorithms by tuning their parameters.

To evaluate the effectiveness of the suggested model, we choose five baseline methods for comparison purposes. These methods include three SNE techniques (DeepWalk, Node2Vec, and LINE) and nine DNE methods (TNE, DynamicTriad, AGA, GDM, TFIP, DGCN, DynamiSE, ERM-ME, and DINE). To ensure fair comparisons in our experiments, we carefully select the best settings for each of the baseline algorithms by tuning their parameters.

- **DeepWalk** (2014) [20] is a widely used approach to embed static networks. It leverages truncated random walks to capture vertex structural information and generates low-dimensional vector representations for each vertex. Close vertices in the random walk sequence are assumed to be similar. Our DeepWalk experiments used a step size of 8, a window size of 5, and 100 steps per vertex.
- **Node2vec** (2016) [22] is a model that improves neighbor selection during random walks by introducing hyperparameters p and q. This results in better node represen-

[1] The complete experiment data and codes are available at https://github.com/gooSAMA/Bi-DNE, and the experiments were performed on a Windows OS with Intel(R) Core(TM) i9-10900K CPU @ 3.70 GHz. and the GPU is a Nvidia 2080ti 12 G. Details of our software and hardware environments were as follows: Windows 11, Python ver. 3.6.6, NumPy ver. 1.19.2, NetworkX ver. 2.1, Gensim ver. 3.8.3, Pandas ver. 0.24.2, Matplotlib ver. 2.2.3.

Table 5 List of hyperparameters

Parameter	Description	Parameter	Description
α	Weight parameter for community structure	β_0, β_1	Weights for micro and macro dynamics
d	Dimension of embedding vectors	γ	Margin hyperparameter
θ	Model parameter		

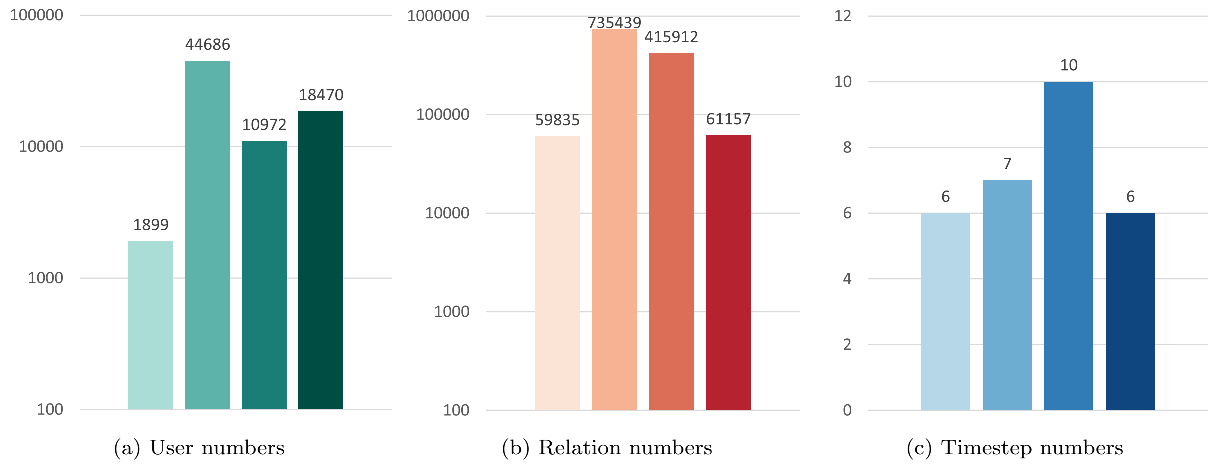


Fig. 5 The statistics of four datasets. From left to right are: fb-messages, ia-facebook, ia-contacts, and ia-retweet

tation. It also uses a skip-gram graph model and has similar step and window size settings as DeepWalk. We set $p = 0.15$ and $q = 4$ in our experiments.

- **LINE** (2015) [23] is a method that captures the first-order similarity between directly connected nodes, preserving local information in the network. It also introduces second-order similarity for nodes with familiar neighbors, capturing global network information. By integrating these two similarity measures, LINE models the relationships between nodes and obtains node representations as embedding vectors.
- **TNE** (2016) [12] is a matrix factorization algorithm for DNE that captures temporal changes. We used its default parameters in our experiments.
- **DynamicTriad** (2018) [35] studies changing networks by analyzing triads. The settings in our experiments are the same as Yang *et. al* [17].
- **AGA** (2021) [30] introduced a budget-constrained optimization module, and achieved superior accuracy on many paths. The hyperparameter β is set to 1.
- **GDM** (2021) [31] is a static network embedding method combined with a Gaussian propagation model. We used its default parameters in our experiments.
- **TFIP** (2022) [46] uses sentiment analyzing methods to embedding static network. The active probability in TFIP is set as 0.1.
- **DGCN** (2022) [41] is a dynamic network embedding model applying GCN modules. The number of hidden layers is set to 2, the hidden layer size is set to 128, and the number of convolutional layers is set to 2.
- **DynamiSE** (2023) [42] is a model that simultaneously encodes the dynamics and symbolic semantics of the network for link prediction. The search range is set to 0.001, 0.01, 0.1, 0, 1, 10, 100.
- **ERM-ME** (2022) [43] is a dynamic network embedding method to capture the attributes and emotions of different roles in social networks. The parameters that control the iteration are the same as Du et al. [43].
- **DINE** (2023) [44] is a network node dynamic embedding model for social recommendation. The batch size is set as 512 and the dimension of embedding d is set as 24. The learning rate and the regularization coefficient are searched in [0.0001, 0.0005, 0.001, 0.0057] and [0.0001, 0.001, 0.01, 0.1], respectively.

Parameter setting

Prior to learning the network representation, a comprehensive examination of different parameter combinations is conducted to identify the optimal control parameters. The main parameters considered in this study are $\eta \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$, $\alpha \in \{0.3, 0.4, 0.5, 0.6, 0.7, 0.8\}$, and $\gamma \in \{0, 1, 2, 3, 4\}$, which are tested with values from predefined sets.

The Bi-DNE model and the baselines are used to obtain embedding vectors for nodes via network representation learning. Subsequently, the positive and negative samples acquired at various time steps are collected, and a logistic regression model is utilized as the classifier. The collected samples are subjected to fivefold cross-validation, and the experiments are repeated ten times to assess the performance of the tested methods. Experimental comparisons are then conducted based on the averaged results.

To determine the most suitable dimension for the latent vector space, different values of $d = \{24, 48, 72, 96, 120\}$ are tested. For performance comparison, we chose embedding vector dimension $d = 48$ after weighing computational complexity and performance.

Evaluate tasks and benchmarks

Bi-DNE learns low-dimensional vertex representations, which are applied to graph-related tasks, including link reconstruction, prediction, and modification. In this study, the embedding vectors of vertices are used to create feature representations for network edges. The feature representation for each edge is formulated as $e_{ij} = \left|u_i - u_j\right|$.

- **Link prediction:** The objective of link prediction in dynamic networks is to predict the presence of future edges within the network. In DNE, low-dimensional representations $\boldsymbol{u}_i^{(a)}$ and $\boldsymbol{u}\,j^{(a)}$ of vertices v_i and v_j at time a are used to predict the existence of an edge eij between them at time $a + 1$. Link prediction is done by using vertex embeddings to estimate connection likelihoods. The embedding vectors of nodes can express the likelihood of connections between them.
- **Link reconstruction:** Link reconstruction is a method for testing the effectiveness of NEs by analyzing the spatial positions of the embedding vectors assigned to two vertices and determining their presence or absence. The relative positioning of two nodes in the embedding space reflects whether they share a connection.
- **Dynamic link prediction and reconstruction:** This work studies how dynamic networks predict and reconstruct changes in their edges, which greatly affect their overall development over time. By analyzing these changes, we can better understand the network's dynamic qualities at different points in time. The network connections are assumed to be dynamic and evolving, with new links forming based on mechanisms like triadic closure.

Link prediction and reconstruction are fundamental components that assess if embeddings effectively encode the presence or absence of edges. Dynamic link prediction and reconstruction, on the other hand, specifically evaluate how well embeddings handle network evolution over time. The problem suite assesses whether Bi-DNE achieves this objective. To assess the Bi-DNE's effectiveness, we consider precision, recall, and F1-score as performance metrics.

Results

This section includes experiments conducted on four datasets using our proposed model Bi-DNE and several baseline models. We evaluate the performance of the learned embedding vectors through four downstream tasks: link reconstruction, link prediction, and joint link reconstruction and prediction. The statistical values obtained from these experiments are recorded in Table 6 and Figs. 6, 7, 8, 9, with the best results highlighted in bold.

Our research shows that the Bi-DNE algorithm performs better than the other baseline algorithms in most cases. The comparison among the F1 scores achieved by Bi-DNE and the other baselines across the four downstream tasks conducted on the fb-messages, ia-facebook, ia-contacts, and ia-retweet-pol datasets is shown in Figs. 6, 7, 8, and 9. It is noteworthy that Bi-DNE outperforms the other algorithms by achieving the best statistical F1 scores in 12 out of 16 tests. These results serve as compelling evidence for the effectiveness and superiority of our algorithm.

The experimental results lead to several key conclusions regarding the six network representation learning methods used in this study, which are categorized into static and dynamic models. The following findings are observed:

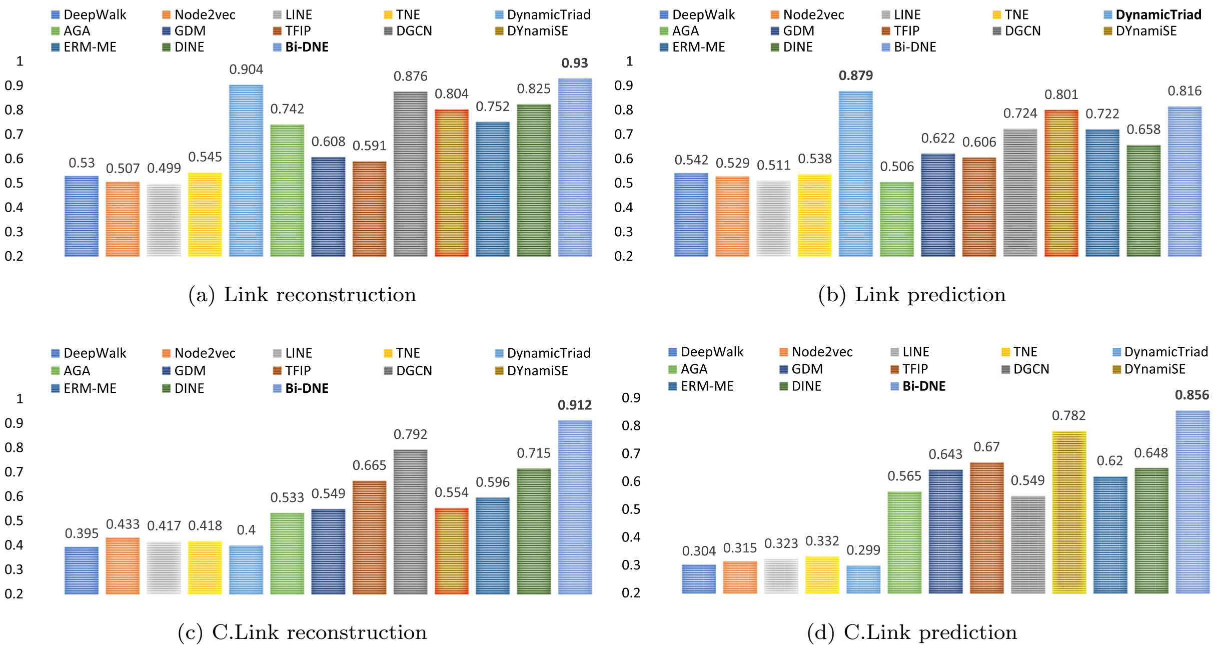
These findings demonstrate the superior performance of Bi-DNE, which effectively combines both micro- and macrolevel network characteristics. The model outperforms other approaches in terms of various tasks and datasets, showcasing its strength as a comprehensive and robust network representation learning method.

(1) Our proposed models, Bi-DNE, DynamicTriad, and TNE, along with other DNE methods, are found to outperform SNE methods such as DeepWalk, LINE, and Node2Vec. This suggests that accounting for the dynamic characteristics of a network along with its topological structure properties during the embedding process can yield an improved network representation learning effect.

(2) Out of the three methods for DNE, Bi-DNE and DynamicTriad have shown to perform the best. These models are able to capture the microlevel evolution trend and fine-grained structural properties of the network by modeling the triadic closure process. They also create representation vectors for each vertex step, which shows how effective it is to incorporate the ternary closure principle in the DNE process. However, TNE has suboptimal performance with sparse network adjacency matrices due to imbalanced samples.

Table 6 The F1-score of Bi-DNE and the baselines exhibited on four datasets

Method	fb-messages				ia-facebook				ia-contacts-Dublin				ia-retweet-pol			
	LR	LP	C.LR	C.LP	LR	LP	C.LR	C.LP	LR	LP	C.LR	C.LP	LR	LP	C.LR	C.LP
DeepWalk	0.530	0.542	0.395	0.304	0.536	0.526	0.610	0.616	0.384	0.370	0.393	0.542	0.533	0.505	0.591	0.474
Node2vec	0.507	0.529	0.433	0.315	0.532	0.481	0.546	0.607	0.371	0.336	0.396	0.453	0.536	0.508	0.594	0.463
LINE	0.499	0.511	0.417	0.323	0.529	0.489	0.548	0.609	0.465	0.450	0.418	0.548	0.539	0.538	0.541	0.467
TNE	0.545	0.538	0.418	0.332	0.524	0.528	0.577	0.573	0.502	0.502	0.564	0.577	0.525	0.530	0.578	0.546
DynamicTriad	0.904	**0.879**	0.400	0.299	0.969	**0.953**	0.582	0.734	0.987	**0.988**	0.578	0.648	0.983	**0.974**	0.613	0.647
AGA	0.742	0.506	0.533	0.565	0.625	0.780	0.549	0.671	0.640	0.578	0.558	0.526	0.598	0.572	0.637	0.543
GDM	0.608	0.622	0.549	0.643	0.748	0.473	0.596	0.577	0.671	0.608	0.535	0.558	0.507	0.652	0.558	0.596
TFIP	0.591	0.606	0.665	0.670	0.595	0.596	0.663	0.620	0.620	0.712	0.517	0.777	0.558	0.516	0.592	0.559
DGCN	0.876	0.724	0.792	0.549	0.953	0.725	0.829	0.787	0.865	0.762	0.759	0.631	0.846	0.789	0.764	0.789
DYnamiSE	0.804	0.801	0.554	0.782	0.64	0.642	0.757	0.643	0.725	0.559	0.609	0.518	0.632	0.764	0.746	0.722
ERM-ME	0.752	0.722	0.596	0.620	0.588	0.701	0.682	0.715	0.646	0.547	0.420	0.702	0.616	0.520	0.703	0.714
DINE	0.825	0.658	0.715	0.648	0.723	0.519	0.772	0.782	0.762	0.459	0.547	0.575	0.697	0.903	0.732	0.699
Bi-DNE	**0.930**	0.816	**0.912**	**0.856**	**0.998**	0.780	**0.970**	**0.973**	**0.998**	0.708	**0.997**	**0.998**	**0.992**	0.857	**0.941**	**0.921**
Improvement	2.60%	−6.30%	51.20%	55.80%	2.90%	−17.30%	38.80%	23.90%	1.10%	−28.00%	41.90%	35.00%	0.90%	−11.70%	32.80%	27.40%



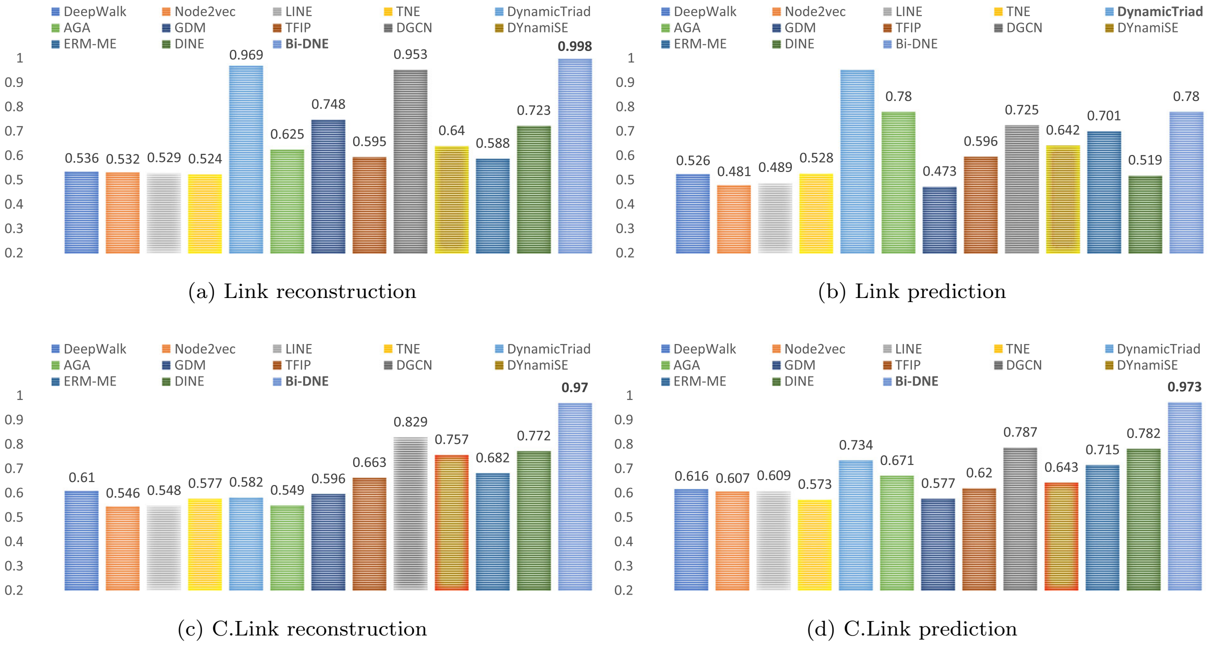
(a) Link reconstruction

(b) Link prediction

(c) C.Link reconstruction

(d) C.Link prediction

Fig. 6 Comparison of F1-scores for all algorithms across fb-messages dataset and downstream tasks



(a) Link reconstruction

(b) Link prediction

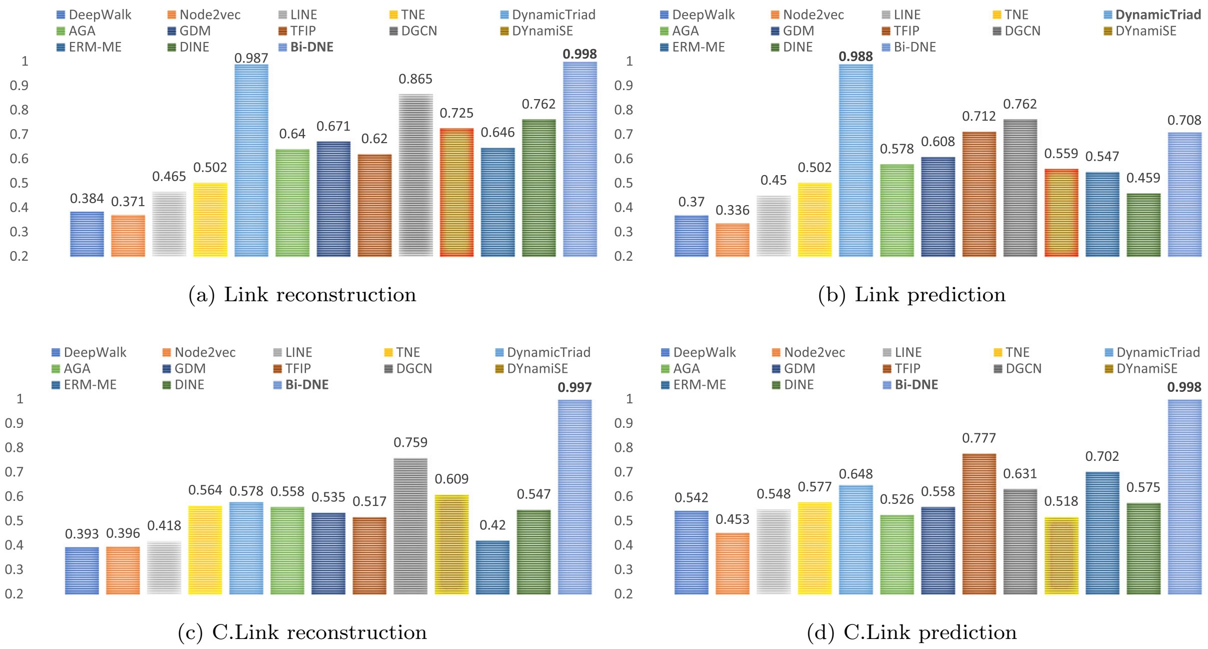
(c) C.Link reconstruction

(d) C.Link prediction

Fig. 7 Comparison of F1-scores for all algorithms across ia-facebook dataset and downstream tasks

(3) Building upon the modeling of the dynamic network evolution process at the microlevel, Bi-DNE further incorporates the macrolevel evolution pattern to encode high-level structures in the embedding space. This additional feature enhances the performance of the model, making Bi-DNE stand out from the other approaches. Our model achieves the best results in three tasks (all except for link prediction) across various datasets.

In addition, the four downstream tasks can be classified into generic tasks and dynamic tasks. The comparisons drawn from our analysis reveal the following key findings: In tasks that require constant change, Bi-DNE has proven to show a considerable improvement in performance. When compared to the baseline model that obtained the highest F1-score on the four datasets for the altered link reconstruction task, Bi-DNE has shown a significant gain of 51.2%,



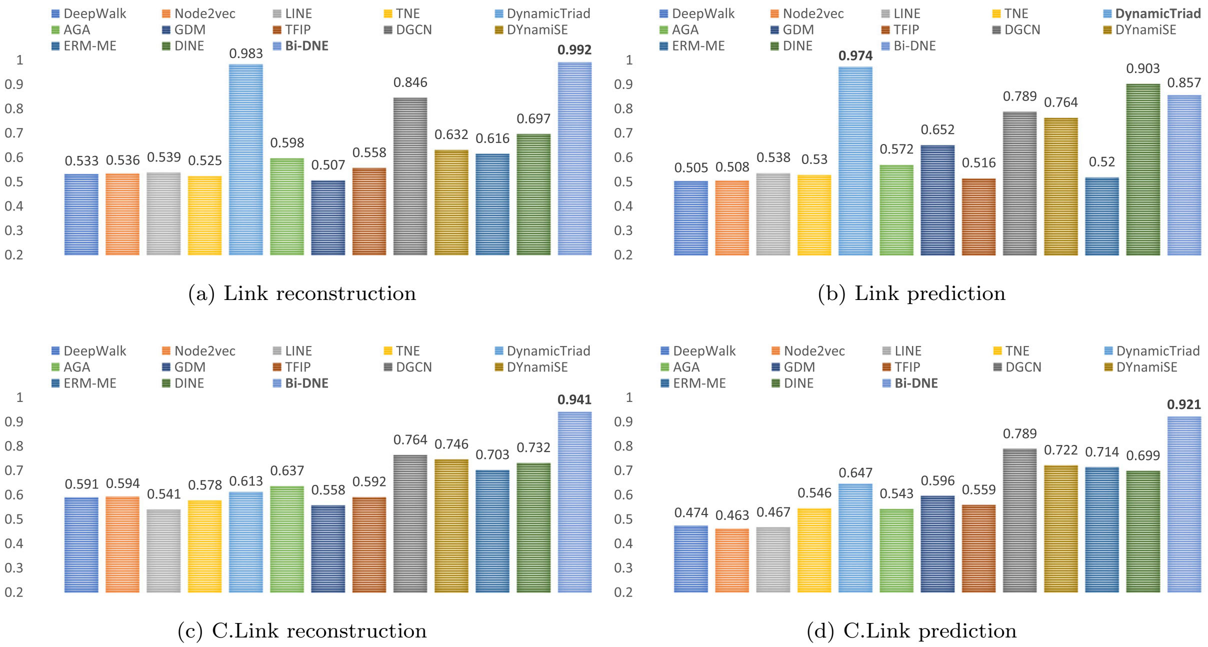
(a) Link reconstruction

(b) Link prediction

(c) C.Link reconstruction

(d) C.Link prediction

Fig. 8 Comparison of F1-scores for all algorithms across ia-contacts-Dublin dataset and downstream tasks



(a) Link reconstruction

(b) Link prediction

(c) C.Link reconstruction

(d) C.Link prediction

Fig. 9 Comparison of F1-scores for all algorithms across ia-retweet-pol dataset and downstream tasks

38.8%, 41.9%, and 32.8%, respectively. Similarly, the Bi-DNE model outperformed the baseline in predicting changed links, with F1 scores and improvements up to 55.8%. It adapts to dynamic networks by capturing changing trends at micro- and macrolevels, successfully preserving evolutionary patterns. Our proposed model is a suitable solution for addressing the needs of dynamic networks. Finally, we analyzed the datasets of four real networks used in the experiment, and the trend of edge changes is shown in Fig. 10:

By carefully observing the network dynamics over time, several noteworthy insights can be gleaned:

- First, the fb-messages dataset exhibits a distinct downward trend in edge changes, with the disappearance of edges outweighing the generation of new edges. This pat-

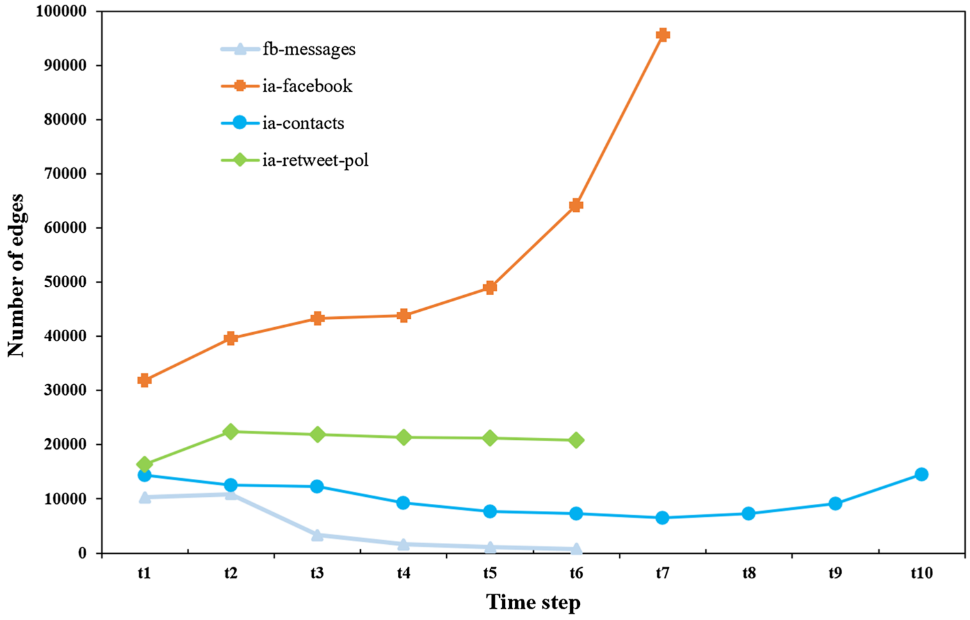


Fig. 10 The trend of edge changes in the four datasets

tern indicates a substantial loss of connections within the network.

- Second, ia-facebook emerges as a prototypical social network, characterized by a high frequency of evolutionary trends commonly observed in real-life social networks. This dataset, therefore, serves as a valuable reference for experimental analysis, given its relevance to the dynamics of social networks.
- Third, the ia-contacts dataset displays a relatively small range of edge changes. This dataset represents a unique scenario where the generation and disappearance of edges resulting from triadic closure processes are roughly balanced.
- Finally, in the ia-retweet-pol dataset, edges demonstrate a gradual and steady growth rate during the initial two moments, followed by a stable phase with minimal changes. Leveraging the principles of ternary closure theory and dense power law, our research successfully captures the dynamic process of edge generation and the overall network growth scale. Consequently, our approach achieves optimal results in networks where the rate of edge growth intensifies, such as ia-facebook, which exemplifies the typical dynamics of social networks.

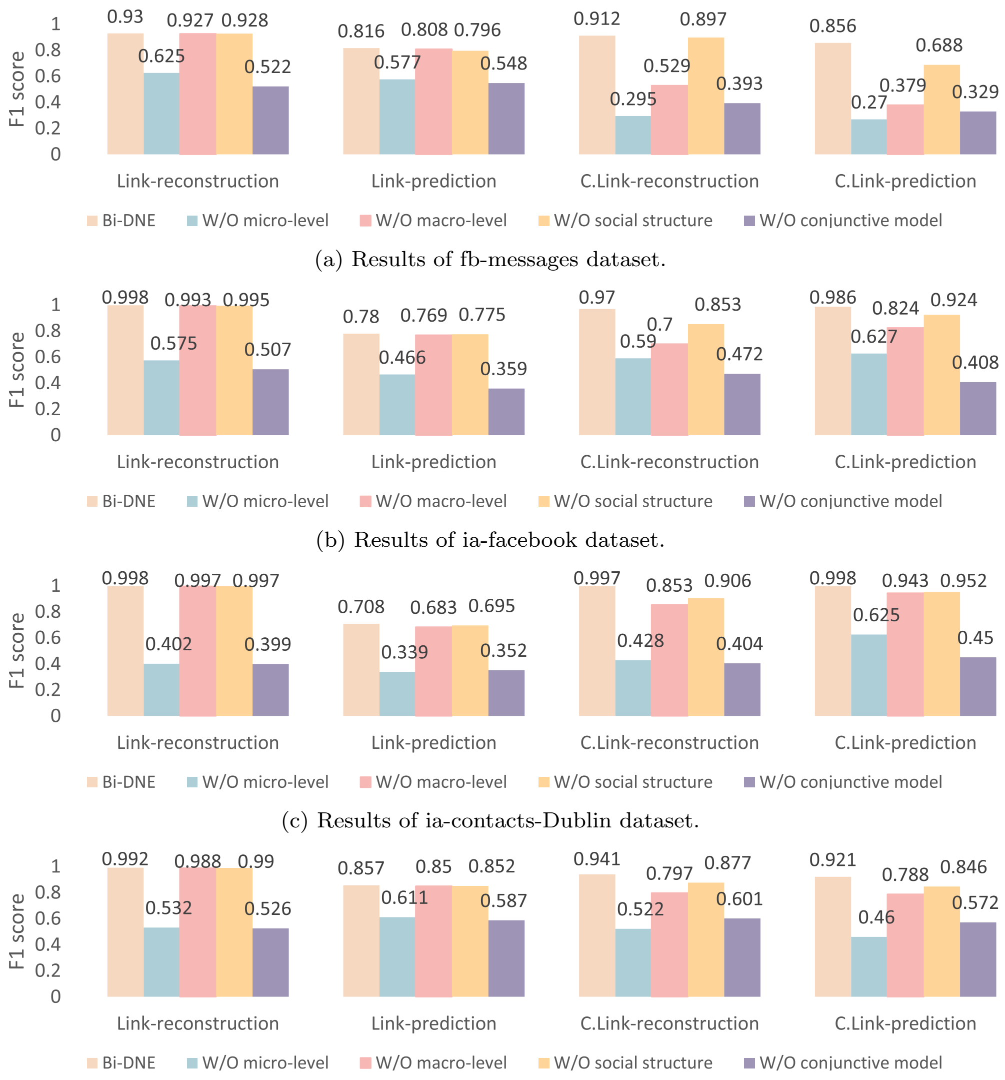
We performed ablation experiments on four datasets to investigate the effect of various modules of this paper on the model's final performance. Table 7 and Fig. 11 present the experimental results, which indicate varying degrees of influence on Bi-DNE's performance from the microlevel structure, macrolevel structure, and community structure. The research shows that embeddings with community structure combined with the removal of macrolevel dense power laws have the greatest influence on Bi-DNE and can enhance the quality of low-dimensional vectors. The modules that combine both macro- and microlevels play a critical role in this process. When comparing the ablation experiment of micro- and macrostructure, we found that the performance of microdense power law decreases more when the macroternary closure rule is removed. This indicates that the macrolevel is the most important in the module that combines macro and micro. From another perspective, both the microlevel and the community structure effectively improve the performance of the triadic closure model. However, the results obtained by only using community structure are not satisfactory, indicating that relying solely on community structure is insufficient for understanding the evolution of social networks. In the experiments conducted on the four data sets, the effect of the changing link task was better, indicating that our method is better at dealing with dynamically changing network characteristics compared to a static network.

Conducting a sensitivity analysis on the utilized parameters is a crucial aspect of research, as it aids in comprehending the extent of influence that parameters have on study outcomes and assessing the soundness of parameter selection. The Bi-DNE algorithm encompasses five primary hyperparameters: (1) α, responsible for controlling community influence, (2) γ, determining the boundary value of the loss function for community structure information, (3) d, representing the vector dimension, (4) β_0, denoting the weight of microlevel network evolution pattern on DNE constraint, and (5) β_1, denoting the weight of macrolevel network evolution pattern on DNE constraint. The range for the parameter α

Table 7 The F1 values of the Bi-DNE model ablation experiments on the four datasets

Method	fb-messages				ia-facebook			
	LR	LP	C.LR	C.LP	LR	LP	C.LR	C.LP
Bi-DNE	**0.93**	**0.816**	**0.912**	**0.856**	**0.998**	**0.78**	**0.97**	**0.986**
W/o mic	0.625 (−32.8%)	0.577 (−29.29%)	0.295 (−67.65%)	0.27 (−68.46%)	0.575 (−42.38%)	0.466 (−40.26%)	0.59 (−39.18%)	0.627 (−36.41%)
W/o mac	0.927 (−0.32%)	0.808 (−0.98%)	0.529 (−42%)	0.379 (−55.72%)	0.933 (−6.51%)	0.769 (−1.41%)	0.7 (−27.84%)	0.824 (−16.43%)
W/o ss	0.928 (−0.22%)	0.796 (−2.4%)	0.879 (−3.6%)	0.688 (−19.6%)	0.995 (−0.3%)	0.775 (−0.64%)	0.853 (−12.06%)	0.924 (−6.29%)
W/o cs	0.522 (−43.87%)	0.548 (−32.84%)	0.393 (−56.91%)	0.329 (−61.57%)	0.507 (−49.2%)	0.359 (−53.97%)	0.472 (−51.34%)	0.408 (−58.62%)
Method	ia-contacts				ia-rewteet-pol			
	LR	LP	C.LR	C.LP	LR	LP	C.LR	C.LP
Bi-DNE	**0.998**	**0.708**	**0.997**	**0.998**	**0.992**	**0.857**	**0.941**	**0.921**
W/o mic	0.402 (−59.72%)	0.339 (−52.12%)	0.428 (−57.07%)	0.625 (−37.37%)	0.532 (−46.37%)	0.611 (−28.7%)	0.522 (−44.53%)	0.46 (−50.05%)
W/o mac	0.997 (−0.1%)	0.683 (−3.53%)	0.853 (−14.44%)	0.943 (−5.51%)	0.988 (−0.4%)	0.85 (−0.82%)	0.797 (−15.3%)	0.788 (−14.44%)
W/o ss	0.997 (−0.1%)	0.695 (−1.84%)	0.906 (−9.13%)	0.952 (−4.61%)	0.99 (−0.2%)	0.852 (−0.58%)	0.877 (−6.8%)	0.846 (−8.14%)
W/o cs	0.399 (−60.02%)	0.352 (−50.28%)	0.404 (−59.48%)	0.45 (−54.91%)	0.526 (−46.98%)	0.587 (−31.51%)	0.601 (−36.13%)	0.572 (−37.89%)

Details of the abbreviation can be found in Table 1



(a) Results of fb-messages dataset.

(b) Results of ia-facebook dataset.

(c) Results of ia-contacts-Dublin dataset.

(d) Results of ia-reteewt-pol dataset.

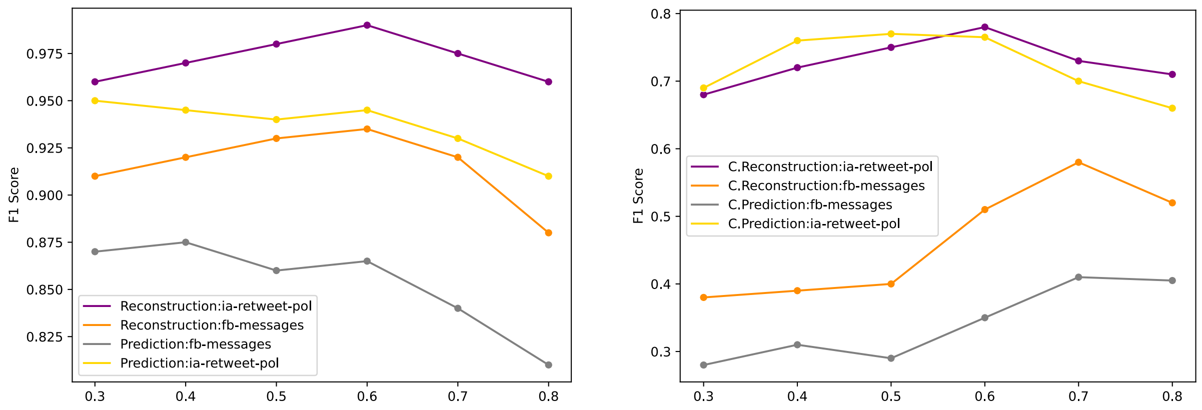
Fig. 11 Ablation experimental results of Bi-DNE on four datasets

was set to $\{0.3, 0.4, 0.5, 0.6, 0.7, 0.8\}$. The parameter γ was tested with values $\{1, 2, 3, 4\}$. The parameter d with values $\{24, 48, 72, 96, 120\}$, β_0 with $\{0.01, 0.1, 0.2, 0.3, 0.4, 0.5\}$, β_1 with $\{0.01, 0.1, 0.2, 0.3, 0.4, 0.5\}$. These experiments were conducted using the fb-message and ia-retweet-pol datasets.

Figure 12 illustrates the results obtained when analyzing the α parameter. It is evident that the performance of Bi-DNE improves across all four tasks as α increases up to a certain threshold. Notably, when α is set to 0.5, our model fails to differentiate whether network vertices belong to the same community. However, as α continues to increase, specifically reaching 0.6, Bi-DNE takes the community structure information into account. This consideration results in a significant increase in the F1 value, indicating that incorporating community structure information proves advantageous for enhancing the performance of the network embedding process.

The experimental findings depicted in Fig. 13 showcase the varying performance of Bi-DNE across different values of γ. The outcomes illustrate a gradual enhancement in the F1 score for all four tasks as γ increases from 0 to 1. Nevertheless, as γ continues to escalate, the model's performance stabilizes, suggesting its robustness to this parameter.

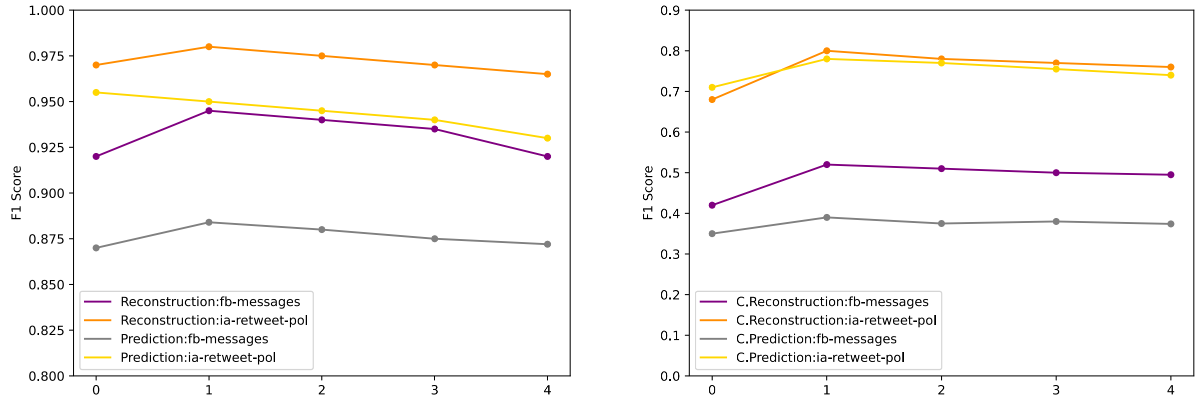
The experimental results captured in Fig. 14 portray the evolving performance of Bi-DNE in terms of the F1 scores with respect to the embedding dimension d. The findings reveal a progressive enhancement in Bi-DNE's performance across the four tasks as d increases from 24 to 96. However, the improvements become marginal upon further increase in the d value, as a substantial portion of the essential information has already been captured within the embedding vectors.



(a) Variations of parameter alpha

(b) Variations of parameter alpha

Fig. 12 Statistical performance of Bi-DNE with different alpha value



(a) Variations of parameter gamma

(b) Variations of parameter gamma

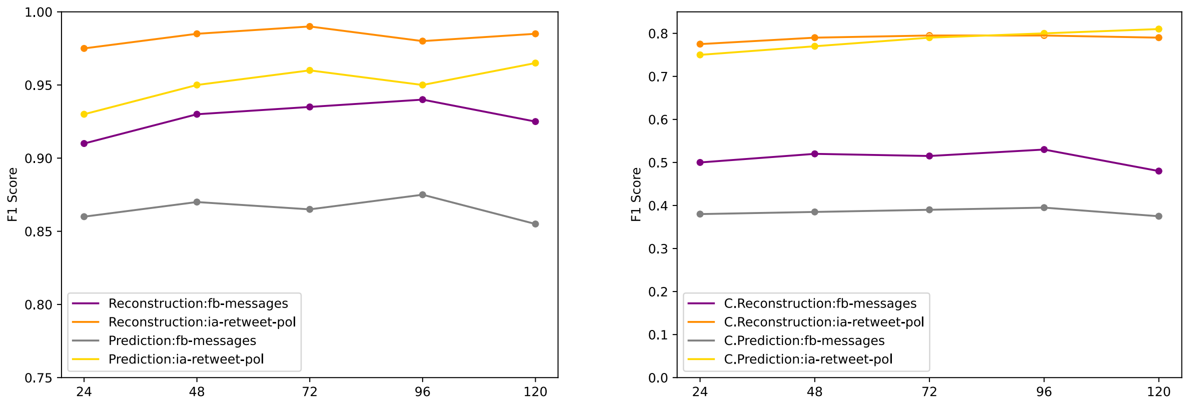
Fig. 13 Statistical performance of Bi-DNE with different gamma value

To strike a balance between computational resources and Bi-DNE's performance, this study sets the number of dimensions to 48.

The analysis conducted on the β_0 parameter, with a fixed value of 0.01, is depicted in Fig. 15. Notably, the model's F1 index scores exhibit variations ranging from 0.003 to 0.009 for the fb-facebook dataset, and 0.002 to 0.013 for the ia-retweet-pol dataset. In Fig. 16, the analysis focuses on the β_1 parameter, while maintaining a fixed value of β_0=0.1. The model's F1 index scores display variations ranging from 0.012 to 0.029 for the fb-facebook dataset, and 0.024 to 0.088 for the ia-retweet-pol dataset. These findings clearly indicate that the proposed model demonstrates consistent performance even when the β_0 and β_1 parameters are adjusted. Hence, it can be concluded that the Bi-DNE model exhibits robustness in relation to these two primary model parameters.

Complexity and overhead analysis

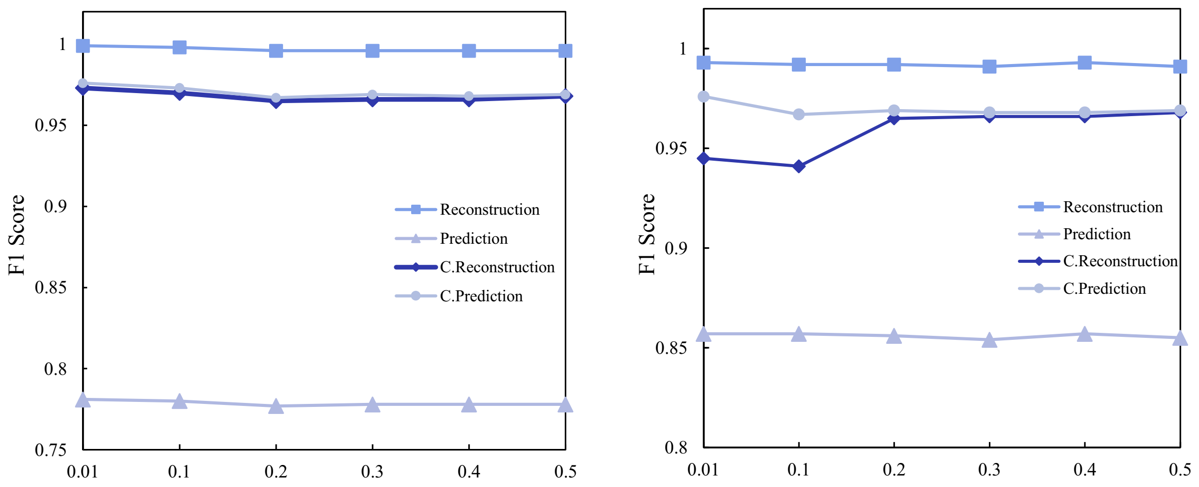
Based on the analysis of the Bi-DNE algorithm in the section "Framework of Bi-DNE model", we can conclude that the algorithm automatically updates coupling coefficients through multiple iterations to obtain part-whole relationships. The algorithmic complexity of Bi-DNE can be seen as the summation of embedding complexities from the microlevel module, macrolevel module, and community structure module. In this process, we assume the number of nodes as n, the embedding dimension as d, and the number of edges as l. The required training time per round for our proposed model is 740 seconds, while Deepwalk is 95 seconds and DynamicTraid is 702 seconds. We compared Bi-DNE with DynamicTriad, another triadic closure-based model, and classic representation learning algorithms Deep-



(a) Variations of parameter d

(b) Variations of parameter d

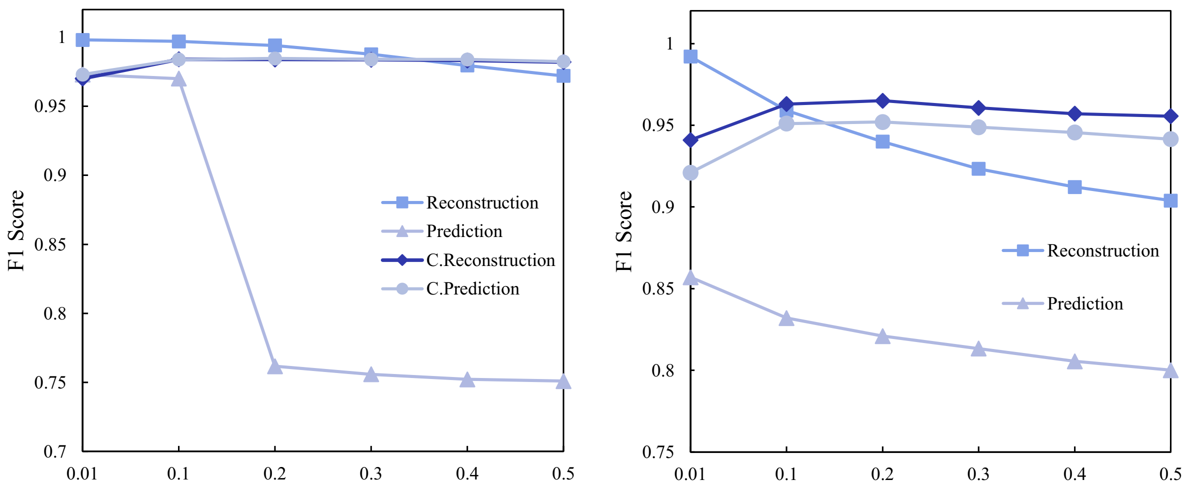
Fig. 14 Statistical performance comparison of Bi-DNE with different values of d



(a) ia-facebook-dataset d

(b) ia-retweet-pol dataset d

Fig. 15 Statistical performance of Bi-DNE with different β_0



(a) ia-facebook-dataset d

(b) ia-retweet-pol dataset d

Fig. 16 Statistical performance of Bi-DNE with different β_0

Walk and Node2Vec. With a microlevel complexity of $O(n^3)$ utilizing triadic closure, macrolevel complexity of $O(nd)$, and community structure complexity of $O(nd)$, the overall complexity of Bi-DNE is $O(n^3+2nd)$, which is still $O(n^3)$. DynamicTriad performs well on link prediction tasks and has an $O(n^3)$ worst-case complexity. DeepWalk and Node2Vec require additional consideration of walk times, thus having an overall split complexity of $O(nd+nl)$ for walks and embedding. Finally, based on the dataset attributes in this paper, the complexity of Bi-DNE and DynamicTriad is $O(n^3)$. At the same time, the classic network embedding algorithms like DeepWalk and Node2Vec have a complexity of $O(n^2)$. The running time of the three algorithms is in line with the complexity of their respective algorithms. While both Bi-DNE and DynamicTraid have an $O(n^3)$ complexity, Bi-DNE also takes into account the community structure and dense power-law overhead, resulting in a slightly longer running time compared to the DynamicTraid method.

Real-world feasibility analysis

The paper uses real-world datasets, which allows the proposed model to demonstrate its excellent performance in practical applications. Although Bi-DNE may be more computationally complex than traditional static network embedding models, it is still feasible with modern computing resources. The advantage of Bi-DNE over traditional models is its ability to effectively capture the evolutionary process of a network over time, which is invaluable in many practical applications. As a dynamic network embedding model with superior performance, Bi-DNE is suitable for real-time or near-real-time analysis tasks, such as social media trend analysis. To successfully implement the Bi-DNE model in real-world scenarios, it is first necessary to collect high-quality dynamic network datasets and perform necessary preprocessing work, such as denoising and normalization, to ensure the quality of the input data. Additionally, appropriate model parameters, such as the community influence coefficient and learning rate, should be selected for performing accurate tuning according to actual business needs. Cloud computing platforms can also be used to deploy model services to ensure high availability, low latency, and service scalability. Finally, continuous model monitoring, evaluation, and update mechanisms should be established to ensure the stability and continuous improvement of the model performance. After conducting pretraining with data from different fields, our proposed method can also be applied to user profiling and recommendation in social networks, abnormal transaction detection in financial networks, and traffic flow prediction.

Conclusion

In this paper, we analyze the evolution of networks on both a micro- and macrolevel. At the microlevel, nodes in the network dynamically form their structure based on the triadic closure principle. Meanwhile, the number of edges in the network on a macrolevel follows a power-law distribution that determines the scale of network evolution. Based on these insights, we introduce the integration of triadic closure and dense power-law concepts into network representation learning. To address this important problem, a novel dynamic network representation learning model called Bi-DNE is proposed. The model captures the process of network structure formation through a strengthened triadic closure mechanism at the microlevel. Additionally, a dynamic equation ensures that the network structure adheres to the dense power-law evolution pattern at the macrolevel. To further enhance the quality of network embedding, the model incorporates community structure, enabling the extraction of richer information from the network. To assess the effectiveness of the proposed model, a comprehensive evaluation is conducted on four real-world datasets, encompassing diverse tasks, such as link prediction, link reconstruction, changed link prediction, and reconstruction. The experimental findings consistently demonstrate that the Bi-DNE framework adeptly captures the dynamic evolution information, providing robust evidence of its effectiveness and practical utility.

In this paper, we analyze the evolution processes of networks at both the micro- and macrolevels. At the microlevel, the nodes in a network dynamically form their structure based on the triadic closure principle. Furthermore, the number of edges in the network at the macrolevel follows a power-law distribution that determines the scale of network evolution. Based on these insights, we integrate the triadic closure and dense power-law concepts into the network representation learning process. To address this important problem, a novel dynamic network representation learning model called Bi-DNE is proposed. The model captures the network structure formation process through a strengthened triadic closure mechanism at the microlevel. Additionally, a dynamic equation ensures that the network structure adheres to the dense power-law evolution pattern at the macrolevel. To further enhance the quality of network embeddings, the model incorporates community structure information, enabling the extraction of richer information from the network. To assess the effectiveness of the proposed model, a comprehensive evaluation is conducted on four real-world datasets, encompassing diverse tasks, such as link prediction, link reconstruction, changed link prediction, and reconstruction. The experimental findings consistently demonstrate that the Bi-DNE framework adeptly captures dynamic evolution information, providing robust evidence of its effectiveness and practical utility.

Acknowledgements This work is supported by the the Science and Technology Program of Sichuan Province (No. 2023YFS0424), the "Open bidding for selecting the best candidates" Technology Project of Chengdu (No. 2023-JB00-00020-GX), the National Natural Science Foundation (Nos. 61902324, 11426179, and 61872298), and the Foundation of Cyberspace Security Key Laboratory of Sichuan Higher Education Institutions (No. sjzz2016-73).

Data availability The data used to support the findings of this study are publicly available at https://github.com/Goosama/Bi-DNE.

Declarations

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

1. Rostami M, Oussalah M, Berahmand K, Farrahi V (2023) Community detection algorithms in healthcare applications: a systematic review. IEEE Access 11:30247–30272. https://doi.org/10.1109/ACCESS.2023.3260652
2. Souravlas S, Anastasiadou SD, Economides T, Katsavounis S (2023) Probabilistic community detection in social networks. IEEE Access 11:25629–25641. https://doi.org/10.1109/ACCESS.2023.3257021
3. Liben-Nowell D, Kleinberg J (2007) The link-prediction problem for social networks. J Am Soc Inf Sci Technol 58(7):1019–1031
4. Chen J, Zhang Q, Huang X (2016) Incorporate group information to enhance network embedding. In: Proceedings of the 25th ACM international on conference on information and knowledge management, CIKM '16. Association for Computing Machinery, New York, pp 1901–1904
5. Xiao W, Peng C, Jing W, Jian P, Yang S (2017) Community preserving network embedding. In: The 31st AAAI conference on artificial intelligence
6. Shi C, Hu B, Zhao X, Yu P (2017) Heterogeneous information network embedding for recommendation. IEEE Trans Knowl Data Eng 31:357–370
7. Laurens VDM, Hinton G (2008) Visualizing data using t-SNE. J Mach Learn Res 9(2579–2605):2579–2605
8. Chen B, Chen X (2022) MAUIL: multilevel attribute embedding for semisupervised user identity linkage. Inf Sci 593:527–545. https://doi.org/10.1016/j.ins.2022.02.023
9. Gu X, Chen X, Lu P, Lan X, Li X, Du Y (2023) SiMaLSTM-SNP: novel semantic relatedness learning model preserving both Siamese networks and membrane computing. J Supercomput. https://doi.org/10.1007/s11227-023-05592-7
10. Ding J, Chen X, Lu P, Yang Z, Li X, Du Y (2023) DialogueINAB: an interaction neural network based on attitudes and behaviors of interlocutors for dialogue emotion recognition. J Supercomput 79(18):20481–20514. https://doi.org/10.1007/s11227-023-05439-1
11. Xiao Y, Jin Y, Cheng R, Hao K (2022) Hybrid attention-based transformer block model for distant supervision relation extraction. Neurocomputing 470:29–39. https://doi.org/10.1016/j.neucom.2021.10.037
12. Zhu L, Guo D, Yin J, Steeg GV, Galstyan A (2016) Scalable temporal latent space inference for link prediction in dynamic social networks. IEEE Trans Knowl Data Eng 28(10):2765–2777
13. Gu X, Wang Z, Chen X, Lu P, Du Y, Tang M (2023) Influence maximization in social networks using role-based embedding. Netw Heterog Media 18(4):1539–1574. https://doi.org/10.3934/nhm.2023068
14. Li J, Dani H, Hu X, Tang J, Chang Y, Liu H (2017) Attributed network embedding for learning in a dynamic environment. In: Proceedings of the 2017 ACM on conference on information and knowledge management, CIKM '17. Association for Computing Machinery, New York, pp 387–396. https://doi.org/10.1145/3132847.3132919
15. Goyal P, Kamra N, He X, Liu Y (2018) Dyngem: Deep embedding method for dynamic graphs. arXiv:1805.11273
16. Coleman JS (1994) Foundations of social theory. American political science review. Harvard University Press, Cambridge
17. Yang M, Chen X, Chen B, Lu P, Du Y (2023) DNETC: dynamic network embedding preserving both triadic closure evolution and community structures. Knowl Inf Syst 65(3):1129–1157. https://doi.org/10.1007/s10115-022-01792-4
18. Belkin M, Niyogi P (2001) Laplacian eigenmaps and spectral techniques for embedding and clustering. Adv Neural Inf Process Syst 14(6):585–591
19. Roweis S, Saul L (2000) Nonlinear dimensionality reduction by locally linear embedding. Science 290(5500):2323–2326
20. Perozzi B, Al-Rfou R, Skiena S (2014) Deepwalk: online learning of social representations. In: Proceedings of the 20th ACM SIGKDD international conference on knowledge discovery and data mining. ACM, New York, pp 701–710. https://doi.org/10.1145/2623330.2623732
21. Mikolov T, Chen K, Corrado G, Dean J (2013) Efficient estimation of word representations in vector space. In: Proceedings of the 1st international conference on learning representations—workshop track. ICLR, Scottsdale
22. Grover A, Leskovec J (2016) node2vec: scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. ACM, New York, pp 855–864. https://doi.org/10.1145/2939672.2939754
23. Tang J, Qu M, Wang M, Zhang M, Yan J, Mei Q (2015) Line: large-scale information network embedding. In: Proceedings of the 24th international conference on world wide web. ACM, New York, pp 1067–1077. https://doi.org/10.1145/2736277.2741093
24. Wang D, Cui P, Zhu W (2016) Structural deep network embedding. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. ACM, pp 1225–1234. https://doi.org/10.1145/2939672.2939753
25. Cao S, Lu W, Xu Q (2015) Grarep: learning graph representations with global structural information. In: Proceedings of the 24th ACM international conference on information and knowledge

management, CIKM 2015, Melbourne, October 19–23. ACM, pp 891–900. https://doi.org/10.1145/2806416.2806512

26. Yang J, Leskovec J (2013) Overlapping community detection at scale: a nonnegative matrix factorization approach. In: Proceedings of the sixth ACM international conference on web search and data mining, WSDM '13. Association for Computing Machinery, New York, pp 587–596. https://doi.org/10.1145/2433396.2433471
27. Maoguo G, Cheng C, Yu X, Shanfeng W (2018) Community preserving network embedding based on memetic algorithm. IEEE transactions on emerging topics in computational intelligence, pp 1–11
28. Yang C, Liu Z, Zhao D, Sun M, Chang EY (2015) Network representation learning with rich text information. In: Proceedings of the 24th international joint conference on artificial intelligence. AAAI, Palo Alto, pp 2111–2117
29. Tu C, Liu H, Liu Z, Sun M (2017) Cane: context-aware network embedding for relation modeling. In: Proceedings of the 55th annual meeting of the association for computational linguistics, vol 1: long papers
30. Li D, Liu J, Jeon J, Hong S, Le T, Lee D, Park N (2021) Large-scale data-driven airline market influence maximization. Association for Computing Machinery, New York, pp 914–924
31. Li W, Li Z, Luvembe AM, Yang C (2021) Influence maximization algorithm based on gaussian propagation model. Inf Sci 568:386–402. https://doi.org/10.1016/j.ins.2021.04.061
32. Zhu D, Cui P, Zhang Z, Pei J, Zhu W (2018) High-order proximity preserved embedding for dynamic networks. IEEE Trans Knowl Data Eng 30:2134–2144. https://doi.org/10.1109/TKDE.2018.2822283
33. Lun D, Yun W, Song G, Lu Z, Wang J (2018) Dynamic network embedding: an extended approach for skip-gram based network embedding. In: Twenty-seventh international joint conference on artificial intelligence IJCAI-18
34. Yu W, Cheng W, Aggarwal CC, Zhang K, Chen H, Wang W (2018) Netwalk: a flexible deep embedding approach for anomaly detection in dynamic networks. In: Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery and data mining, KDD '18. Association for Computing Machinery, New York, pp 2672–2681. https://doi.org/10.1145/3219819.3220024
35. Zhou L, Yang Y, Ren X, Wu F, Zhuang Y (2018) Dynamic network embedding by modeling triadic closure process. In: Proceedings of the thirty-second AAAI conference on artificial intelligence (AAAI-18), the 30th innovative applications of artificial intelligence (IAAI-18), and the 8th AAAI symposium on educational advances in artificial intelligence (EAAI-18), New Orleans, February 2–7. AAAI Press, pp 571–578. https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16572
36. Ma L, Zhang Y, Li J, Lin Q, Bao Q, Wang S, Gong M (2020) Community-aware dynamic network embedding by using deep autoencoder. Inf Sci 519:22–42
37. Nguyen GH, Lee JB, Rossi RA, Ahmed NK, Koh E, Kim S (2018) Continuous-time dynamic network embeddings. In: Companion proceedings of the the web conference 2018, WWW '18. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, pp 969–976. https://doi.org/10.1145/3184558.3191526
38. Zuo Y, Liu G, Lin H, Guo J, Hu X, Wu J (2018) Embedding temporal network via neighborhood formation. In: Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery and data mining, KDD '18. Association for Computing Machinery, New York, pp 2857–2866. https://doi.org/10.1145/3219819.3220054
39. Lu Y, Wang X, Shi C, Yu PS, Ye Y (2019) Temporal network embedding with micro- and macro-dynamics. In: Proceedings of the 28th ACM international conference on information and knowledge management, CIKM 2019, Beijing, November 3–7. ACM, 2019, pp 469–478. https://doi.org/10.1145/3357384.3357943
40. Zhou L, Yang Y, Ren X, Wu F, Zhuang Y (2018) Dynamic network embedding by modeling triadic closure process. In: Proceedings of the thirty-second AAAI conference on artificial intelligence (AAAI-18), the 30th innovative applications of artificial intelligence (IAAI-18), and the 8th AAAI symposium on educational advances in artificial intelligence (EAAI-18), New Orleans, February 2–7. AAAI Press, pp 571–578. https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16572
41. Zhang C, Li W, Wei D, Liu Y, Li Z (2022) Network dynamic GCN influence maximization algorithm with leader fake labeling mechanism. IEEE Trans Comput Soc Syst. https://doi.org/10.1109/TCSS.2022.3193583
42. Sun H, Tian P, Xiong Y, Zhang Y, Xiang Y, Jia X, Wang H (2023) Dynamise: dynamic signed network embedding for link prediction. In: 10th IEEE international conference on data science and advanced analytics, DSAA 2023, Thessaloniki, October 9–13, 2023. IEEE, pp 1–2. https://doi.org/10.1109/DSAA60987.2023.10302493
43. Du Y, Wang Y, Hu J, Li X, Chen X (2022) An emotion role mining approach based on multiview ensemble learning in social networks. Inf Fusion 88:100–114. https://doi.org/10.1016/J.INFFUS.2022.07.010
44. Zhang Y, Meng D, Zhang L, Kong C (2023) Dine: dynamic information network embedding for social recommendation. In: Web information systems and applications. Springer Nature Singapore, Singapore, pp 76–87
45. Rossi RA, Ahmed NK (2015) The network data repository with interactive graph analytics and visualization. In: Proceedings of the twenty-ninth aaai conference on artificial intelligence, AAAI'15. AAAI Press, pp 4292–4293
46. Li W, Li Y, Liu W, Wang C (2022) An influence maximization method based on crowd emotion under an emotion-based attribute social network. Inf Process Manag 59(2):102818. https://doi.org/10.1016/j.ipm.2021.102818