



# 计算机科学

COMPUTER SCIENCE

## 自适应建模网络动力学的动态链路预测方法

郭翹, 侯锦霖, 王文俊, 焦鹏飞

引用本文

郭翹, 侯锦霖, 王文俊, 焦鹏飞. [自适应建模网络动力学的动态链路预测方法](#)[J]. 计算机科学, 2025, 52(6): 118-128.

GUO Xuan, HOU Jinlin, WANG Wenjun, JIAO Pengfei. [Dynamic Link Prediction Method for Adaptively Modeling Network Dynamics](#) [J]. Computer Science, 2025, 52(6): 118-128.

---

## 相似文章推荐 (请使用火狐或 IE 浏览器查看文章)

Similar articles recommended (Please use Firefox or IE to view the article)

### [基于多视图表示学习的语义感知异质图注意力网络](#)

Semantic-aware Heterogeneous Graph Attention Network Based on Multi-view Representation Learning

计算机科学, 2025, 52(6): 167-178. <https://doi.org/10.11896/jsjcx.240600032>

### [基于自适应图自编码器的离群点检测方法](#)

Outlier Detection Method Based on Adaptive Graph Autoencoder

计算机科学, 2025, 52(6): 129-138. <https://doi.org/10.11896/jsjcx.240500092>

### [基于生成对抗网络的云制造工业服务选择方法](#)

Selection Method for Cloud Manufacturing Industrial Services Based on Generative Adversarial Networks

计算机科学, 2025, 52(4): 54-63. <https://doi.org/10.11896/jsjcx.241000102>

### [NLGAE:一种基于改进网络结构及损失函数的图自编码器节点分类模型](#)

NLGAE: A Graph Autoencoder Model Based on Improved Network Structure and Loss Function for Node Classification Task

计算机科学, 2024, 51(10): 234-246. <https://doi.org/10.11896/jsjcx.230700122>

### [结构影响力及标签冲突感知的图课程学习方法](#)

Structural Influence and Label Conflict Aware Based Graph Curriculum Learning Approach

计算机科学, 2024, 51(10): 227-233. <https://doi.org/10.11896/jsjcx.230800167>

# 自适应建模网络动力学的动态链路预测方法

郭 翾<sup>1</sup> 侯锦霖<sup>1</sup> 王文俊<sup>1</sup> 焦鹏飞<sup>2</sup>

1 天津大学智能与计算学部 天津 300350

2 杭州电子科技大学网络空间安全学院 杭州 310018

(guoxuan@tju.edu.cn)

**摘要** 动态网络链路预测是理解和分析动态网络的核心问题之一。针对链路预测面临的捕获复杂网络结构和真实演化规律等困难的问题,提出了一种融合图神经网络和神经常微分方程的自适应网络动力学建模方法——双层活跃度约束神经常微分方程模型 DANOM。DANOM 融合节点的重要性和相对位置信息,增强了网络结构的表征;通过节点活跃度约束下的神经常微分方程单元强化了演化规律的学习过程;并在节点活跃度和节点表示的重构损失优化下,挖掘到网络的有效信息。DANOM 在多个真实数据集上的多种下游任务中均达到了最优效果,其中在单步链路预测任务中 AUC 与 AP 最高分别提升 14% 和 9.7%,在快照缺失情况下的链路预测任务中 AUC 与 AP 分别平均仅降低 0.43% 和 0.03%,在用户缝合实验中 AUC 与 AP 最高分别提升 20.6% 和 24.4%。

**关键词**: 图表示学习; 动态网络; 链路预测; 神经常微分方程; 网络动力学

**中图分类号** TP391

## Dynamic Link Prediction Method for Adaptively Modeling Network Dynamics

GUO Xuan<sup>1</sup>, HOU Jinlin<sup>1</sup>, WANG Wenjun<sup>1</sup> and JIAO Pengfei<sup>2</sup>

1 College of Intelligence and Computing, Tianjin University, Tianjin 300350, China

2 College of Cyberspace Security, Hangzhou Dianzi University, Hangzhou 310018, China

**Abstract** Dynamic network link prediction is one of the core issues in understanding and analyzing dynamic networks. In response to the challenges of capturing complex network structures and real evolution patterns faced by link prediction, this paper proposes a method integrating the graph neural network and neural ordinary differential equation to adaptive model various network dynamics: double-layer activity-constrained neural ordinary differential equation model (DANOM). DANOM integrates the importance and relative positional information of nodes to enhance the representation of network structures, strengthens the learning process of evolution patterns through neural ordinary differential equation units constrained by node activity, and mines effective information of the network under the reconstruction loss of node activity and node representation. DANOM achieves optimal results in various down-stream tasks on multiple real-world datasets. It achieves the highest improvements of 14% and 9.7% in terms of AUC and AP, respectively, in the single-step link prediction task. In cases of snapshot missingness, the average AUC and AP of link prediction are only reduced by 0.43% and 0.03%, respectively. In the user stitching experiments, DANOM achieves the highest improvements of 20.6% and 24.4% in terms of AUC and AP, respectively.

**Keywords** Graph representation learning, Dynamic network, Link prediction, Neural ordinary differential equation, Network dynamics

## 1 引言

动态网络作为一种现实世界复杂系统的抽象表示,被广泛应用,其不仅描述了静态链接,还揭示了随时间演变的链接模式。研究动态网络对于理解社交网络、学术合作等复杂系

统的内在运作机制至关重要,且对于预测和影响这些系统的未来发展趋势具有重要的实际意义<sup>[1]</sup>。

动态链路预测(Dynamic Link Prediction, DLP)的核心目标是利用动态网络的历史数据来预测未来可能形成的链接,这有助于揭示现实网络中的隐藏社会关系、信息流动路径或

到稿日期:2024-04-04 返修日期:2024-09-03

基金项目:海南省重点研发项目(ZDYF2024SHFZ051);国家自然科学基金(62372146);浙江省自然科学基金(LDT23F01015F01);新奥集团合作项目(2023GKF-1220)

This work was supported by the Key R&D Program of Hainan(ZDYF2024SHFZ051), National Natural Science Foundation of China(62372146), Zhejiang Provincial Natural Science Foundation(LDT23F01015F01) and ENN Group Project(2023GKF-1220).

通信作者:王文俊(wjwang@tju.edu.cn)

潜在功能链接,还对预防网络欺诈、优化社媒推荐、分析学术前沿等具有重大意义<sup>[2]</sup>。

动态网络中的链接可能随时间出现或消失,这要求 DLP 方法既要捕捉网络的当前状态,还要理解其随时间的递推演化模式,即网络动力学。现有 DLP 方法大多基于深度学习技术,这些方法分开处理静态和动态信息<sup>[3]</sup>:首先通过多层感知机(Multi-Layer Perceptron, MLP)<sup>[4-5]</sup>或图神经网络(Graph Neural Network, GNN)<sup>[6-7]</sup>等算法学习快照网络的静态结构特征,然后利用循环神经网络(Recurrent Neural Network, RNN)<sup>[8]</sup>或注意力机制<sup>[9]</sup>等技术来掌握网络的动态变化。然而,现有方法常常忽略了一些关键的结构信息,如节点重要性和相对位置等,做出预测也仅是依赖历史快照信息对未来快照计算近似表示,未能建立快照间的递推演化模式,难以应对复杂多变的现实应用环境。例如,由于收集过程中的权限、隐私等限制,现实网络数据往往不够完整(如某些时间点的信息缺失);用户可能增加活动的隐蔽性,如使用多个账号进行不同类型的活动。为适应上述复杂环境,DLP 方法需要自适应地推理缺失的数据信息并归纳个体的行为规律,其本质在于有效地提取快照中的关键信息并建模快照间的动力学模式。

因此,本文引入神经常微分方程(Neural Ordinary Differential Equation, NODE)<sup>[10]</sup>,旨在自适应地建模网络动力学,以提升模型的理解和预测能力。具体地,本文提出了双层活跃度约束的神经常微分方程模型(Double-layer Activity-Constrained Neural Ordinary Differential Equation Model, DANOM)。该模型融合了节点快照重要性、全局节点重要性、结构相对位置及时间维度相对位置 4 种关键信息,用于丰富每个动态网络快照表征的信息。模型构建了多组双层的神经常微分方程单元,每组包含学习模块和节点活跃度模块,其分别用于表示层和统计性的递推建模。相邻两组通过局部和全局图卷积网络(Graph Convolutional Network, GCN)<sup>[11]</sup>激活,确保节点活跃度对时序表示的监督和修正。

本文的主要贡献包括:

1)指出 DLP 方法需要有效地提取快照中的关键信息并建模快照间的动力学模式。现有研究在这两方面都存在不足。

2)提出用于 DLP 的框架 DANOM,通过节点活跃度约束下的 NODE 单元和多维度结构编码从表示层和统计性两方面有效建模网络动力学。

3)在多个真实网络上的实验验证了,DANOM 能有效处理多种链路预测任务,且在网络动力学、节点活跃度和用户缝合实验中也表现优异。

本文第 2 章介绍相关工作;第 3 章介绍相关定义和问题的表述;第 4 章介绍本文所提算法;第 5 章展示实验结果并对其进行分析;最后总结全文。

## 2 相关工作

动态网络表示学习是当前 DLP 的主流方法。本文根据对动态信息的不同处理方式将这些方法大致分为 3 类:基于循环神经网络的方法、基于注意力机制的方法和基于神经常微分方程的方法。

### 2.1 基于循环神经网络的方法

这类方法利用循环神经网络,如长短时记忆网络(Long

Short Term Memory Network, LSTM)和门控循环单元(Gated Recurrent Unit, GRU)等,来编码网络的时序信息。例如, DynGEM<sup>[12]</sup>引入自编码器学习每个网络快照的静态结构信息,并利用 RNN 串联不同快照,使用前一快照参数初始化当前快照,加速模型收敛并保持节点表示的稳定性。Dyn-graph2vec<sup>[8]</sup>通过结合 MLP 和 LSTM,综合考虑历史时刻对当前状态的影响,实现对时间序列中高度非线性交互的捕捉。DGCN<sup>[13]</sup>利用 LSTM 更新 GCN 的权重参数,并使用相似度信息指导邻居信息聚合。EvolveGCN<sup>[14]</sup>同样在每个时刻用 LSTM 或 GRU 更新 GCN 参数,使模型适应更近期的网络快照。ROLAND<sup>[15]</sup>通过 LSTM 或 GRU 将静态 GNN 扩展为动态 GNN,融合邻近快照的 GNN 隐表示以学习网络的演化。一些研究将生成对抗网络应用到 DLP 中,例如 GCN-GAN<sup>[16]</sup>将 GCN 学习的结构信息通过 LSTM 得到时序补充,进而通过 MLP 鉴别器进行结果修正。NetworkGAN<sup>[17]</sup>在静态信息学习上采用矩阵分解方法,随后通过 LSTM 学习时序信息,并结合 GNN 和 MLP 构建鉴别器以优化学习效果。

### 2.2 基于注意力机制的方法

基于注意力机制的方法通过注意力机制学习网络的时序表示。DySAT<sup>[18]</sup>利用拓展图注意力网络学习快照静态信息,并使用自注意力机制学习时序信息,这一组合允许模型有效地融合时间序列数据,更精确地捕捉网络中的关键时序特征和动态变化。此外,STGSN<sup>[9]</sup>使用 GNN 学习从所有历史快照中提取到的全局邻接信息,并用注意力机制将其融合,实现了图结构学习的显著提升,且在对时间依赖性的理解上,尤其是在捕捉长期依赖关系方面展现出了显著的优势。

### 2.3 基于神经常微分方程的方法

基于神经常微分方程的方法通过神经常微分方程(NODE)来挖掘网络的时序信息。NDCN<sup>[19]</sup>首次结合 NODE 和 GNN 来学习动态网络表示。该方法将 GCN 作为常微方程函数,有效地实现了对网络静态和动态特征的同时捕捉。与其他时序信息学习模型相比,NODE 能够提高深度学习模型逐步推理的能力,更好地预测复杂的、非线性的网络演变过程。然而该方法对关键结构信息的编码能力不足,且以 GCN 作为常微方程函数使其适应性受到网络拓扑的约束。本文对此进行了更进一步的探索。

## 3 基本概念

### 3.1 动态网络

本文研究的动态网络定义为有序的快照序列  $G = \{g^0, g^1, \dots, g^T\}$ ,其中  $g^t = (\mathcal{V}^t, \mathcal{E}^t)$  表示  $t$  时刻包含节点集  $\mathcal{V}^t$  和边集  $\mathcal{E}^t$  的快照。用  $\mathcal{V} = \mathcal{V}^0 \cup \mathcal{V}^1 \cup \dots \cup \mathcal{V}^T = \{v_0, v_1, \dots, v_{N-1}\}$  表示所有历史快照中全部  $N$  个节点的集合,  $\mathcal{E} = \mathcal{E}^0 \cup \mathcal{E}^1 \cup \dots \cup \mathcal{E}^T$  为所有历史快照中出现过的边,则可构建全局网络  $g^s = (\mathcal{V}, \mathcal{E})$  表示全局时间上的关系。 $g^t$  的邻接结构可用二元邻接矩阵  $\mathbf{A}^t \in \{0, 1\}^{N \times N}$  表示,  $\mathbf{A}_{ij}^t = 1$  表示边  $(v_i, v_j) \in \mathcal{E}^t$ ,  $\mathbf{A}_{ij}^t = 0$  则表示  $(v_i, v_j) \notin \mathcal{E}^t$ 。本文称一个节点  $v_i$  在  $g^t$  的节点度  $a_i^t = \sum_j \mathbf{A}_{ij}^t$  为其快照级活跃度,包含全部节点活跃度的  $\mathcal{A}^t \in \mathbb{R}^{N \times 1}$  为时刻  $t$  的节点活跃度矩阵,其中节点活跃度最大值表示为  $d_{\max}^t$ 。类似地,在全局网络  $g^s$  的邻接矩阵  $\mathbf{A}^s \in \{0, 1\}^{N \times N}$  上计算节点度  $a_i^s = \sum_j \mathbf{A}_{ij}^s$ ,可得全局级节点活跃度矩阵  $\mathcal{A}^s \in \mathbb{R}^{N \times 1}$ 。

### 3.2 动态网络表示学习

对于给定的动态网络  $G$ , 动态网络表示学习的目标是获得每个节点  $v_i$  在时刻  $t$  的低维表示  $z_i^t$ , 并将学习到的低维表示用于下游预测任务, 如链路预测等。

### 3.3 动态链路预测

传统链路预测通过历史快照  $\{g^0, g^1, \dots, g^{\tau-1}\}$  预测网络在  $\tau$  时刻的邻接关系  $A^\tau$ 。本文将这种单步链路预测扩展到多步链路预测, 即预测未来  $l$  个快照的邻接关系  $\{A^\tau, A^{\tau+1}, \dots, A^{\tau+l-1}\}$ 。此外, 通过删除一定比例的历史快照来模拟数据缺失场景, 并在数据缺失条件下进行单步链接预测任务。多步和快照缺失链路预测要求模型有更强大的动力学建模能力。

### 3.4 其他相关预测任务

1) 节点活跃度预测: 节点活跃度预测的目标是已知历史快照  $\{g^0, g^1, \dots, g^{\tau-1}\}$  的信息, 预测网络在  $\tau$  时刻的节点活跃度矩阵  $\mathcal{A}^\tau$ 。

2) 用户缝合: 用户缝合<sup>[20]</sup>的任务目标是已知历史快照  $\{g^0, g^1, \dots, g^{\tau-1}\}$  的信息, 判断给定两个节点  $v_i, v_k \in \mathcal{V}$  是否具有相同的现实意义, 如社交网络中两个账户是否对应同一自然人。通常, 具有相同现实意义的节点表示应该尽可能相似。

### 3.5 神经常微分方程

NODE 结合了深度学习的表达能力和微分方程在建模动态系统中的自然优势, 使之成为建模网络动力学的强大工具。NODE 通过连续时间模型来捕捉网络状态的微小变化, 允许对网络的动态演化进行更精细的建模。

NODE 能够模拟连续动态, 如式(1)所示, 对于一个未来状态  $z(t_i)$ , 可以通过从初始状态  $z(t_0)$  求解获得:

$$z(t_i) = z(t_0) + \int_{t_0}^{t_i} f(z(t), t, \theta) dt \quad (1)$$

其中,  $f(z(t), t, \theta)$  是一个用于拟合微分方程  $dz(t)/dt$  的神经网络, 能够通过常微分方程 (ODE) 函数控制动态系统变化。其中的  $z(t_i)$  表示动态系统在时间  $t_i$  的状态,  $t_i \in [0, T]$  表示 0 到  $T$  中的一个时刻,  $\theta$  则代表模型的参数集合。整个常微分方程的计算是通过一个黑箱的 ODE 求解器得到的, 它使用伴随灵敏度方法来计算梯度, 从而提升梯度计算的效率。为 NODE 提供动态系统的初始状态和一个需要求解的时间序列, 就可以获得该序列的递推公式, 这与 DLP 的要求契合。NODE 还具有内存效率高、训练过程自适应和可建模连续时间序列等优点, 因此本文将将其引入 DLP 方法。

## 4 双层活跃度约束神经常微分方程模型

本章将详细介绍 DANOM 模型, 模型架构如图 1 所示。首先对快照网络进行预处理, 提取快照节点重要性、全局节点重要性、结构相对位置及时序相对位置 4 种结构信息, 并将它们输入特征编码模块进行特征融合作为随后双层演化学习部分的初始输入。双层演化学习部分学习递推演化模式。该部分由多组双层演化单元构成, 每组包含一个学习模块和一个节点活跃度模块。学习模块以上一组演化单元所得表示和节点活跃度为输入, 然后将学得节点表示在节点活跃度模块进行统计性修正。通过  $M$  组演化单元得到最终表示后, 在快照重构、快照平滑、活跃度修正和活跃度重构 4 个优化目标下对模型参数进行优化, 保留最优模型进行下游预测任务。

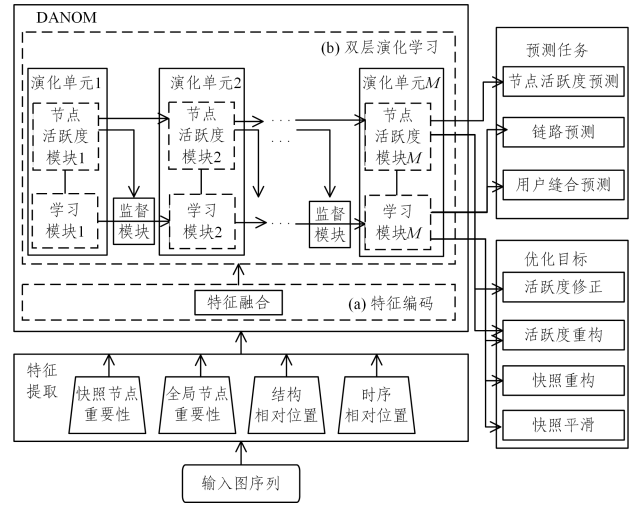


图 1 DANOM 框架图

Fig. 1 DANOM framework

### 4.1 特征提取

#### 4.1.1 快照节点重要性特征

该特征提取每个快照的一阶度、二阶度和聚集系数。快照一阶度由节点活跃度  $\mathcal{A}^t$  表示。二阶度为每个节点的二阶邻居数量, 由矩阵  $\mathcal{A}_2^t \in \mathbb{R}^{N \times 1}$  表示, 其中  $\mathcal{A}_{2,i}^t = |\{v_j \in \mathcal{V} \mid (\mathcal{A}^t)_{ij}^2 > 0\}|$ 。节点聚集系数表示节点邻居的凝聚程度, 体现节点的影响力大小。动态网络在时刻  $t$  的聚集系数矩阵  $C^t \in \mathbb{R}^{N \times 1}$  的计算方式为:

$$C_i^t = \frac{2E_i^t}{a_i^t(a_i^t - 1)} \quad (2)$$

其中,  $E_i^t$  为节点  $v_i$  在  $g^t$  上邻居间的总边数。最终, 通过拼接操作得到时刻  $t$  的快照节点重要性特征  $I^t$ :

$$I^t = \mathcal{A}^t \parallel \mathcal{A}_2^t \parallel C^t \quad (3)$$

其中,  $\parallel$  表示拼接操作。该特征衡量节点活跃性与邻居的凝聚程度, 体现节点的影响力大小。

#### 4.1.2 全局节点重要性特征

全局节点重要性从全局网络  $g^k$  上同样提取一阶度  $\mathcal{A}^k$ 、二阶度  $\mathcal{A}_2^k$  ( $\mathcal{A}_{2,i}^k = |\{v_j \in \mathcal{V} \mid (\mathcal{A}^k)_{ij}^2 > 0\}|$ ) 和聚集系数  $C^k$  ( $C_i^k = 2E_i^k/a_i^k(a_i^k - 1)$ ,  $E_i^k$  为  $v_i$  全局邻居之间的总边数), 则所有快照得到同样的全局节点重要性特征  $I^k$ :

$$I^k = \mathcal{A}^k \parallel \mathcal{A}_2^k \parallel C^k \quad (4)$$

该特征综合全部快照判断节点重要性, 帮助模型获得更大的感受野, 让节点的重要性评估更全面。

#### 4.1.3 结构相对位置性特征

该特征提取快照中每个节点的固定标记位置、社团标记位置和传递标记位置, 其中每种位置特征通过对快照中节点进行标记确定。

1) 固定标记位置。首先, 将快照中度等于 1 的节点  $v_i$  标记为  $P_{f,i}^t = 1$ , 其他节点标记为  $P_{f,i}^t = 0$ 。随后, 按下式更新标记:

$$P_{f,i}^t \leftarrow \begin{cases} 0, & P_{f,i}^t = 0 \wedge \max(P_{f,\Gamma_i^t}) = 0 \\ \max(P_{f,\Gamma_i^t}) + 1, & P_{f,i}^t = 0 \wedge \max(P_{f,\Gamma_i^t}) > 0 \\ P_{f,i}^t, & P_{f,i}^t > 0 \end{cases} \quad (5)$$

其中,  $\Gamma_i^t$  为  $v_i$  在该快照中的邻居集合,  $P_{f,S}^t$  为集合  $S$  所有节点的标记组成的向量。迭代更新标记, 直到所有节点标记不为

0,得到所有节点在快照中的固定标记位置矩阵 $\mathbf{P}_f \in \mathbb{R}^{N \times 1}$ 。

2) 社团标记位置。对快照进行社团划分,得到一组子图,分别在这些子图上采取上述固定标记位置方法,得到时刻  $t$  的社团标记位置矩阵 $\mathbf{P}_c \in \mathbb{R}^{N \times 1}$ 。

3) 传递标记位置。构建转移矩阵 $\mathbf{T}_p$ :

$$\mathbf{T}_p = \text{Diag}(\mathcal{A}')^{-\frac{1}{2}} \mathbf{A}' \text{Diag}(\mathcal{A}')^{-\frac{1}{2}} \quad (6)$$

将快照中度  $1 \leq a_i' \leq 0.1 d_{\max}'$  的节点标记为 1,得到初始传递标记位置矩阵 $\mathbf{P}_p' \in \mathbb{R}^{N \times 1}$ 。随后,通过计算 $\mathbf{P}_p \leftarrow \mathbf{T}_p' \mathbf{P}_p'$  传递标记迭代更新 $\mathbf{P}_p'$ ,直到全部节点均被标记。

最终拼接得到时刻  $t$  的结构相对位置矩阵 $\mathbf{P}_s'$ :

$$\mathbf{P}_s' = \mathbf{P}_f' \parallel \mathbf{P}_c' \parallel \mathbf{P}_p' \quad (7)$$

该特征为模型训练提供有效的相对位置信息,以更好地区分节点,避免不同节点表示过于相似。

#### 4.1.4 时序相对位置性特征

时序相对位置性特征提取网络中每个节点起始时刻映射、消失时刻映射和出现频次,所有快照共享该特征。节点在快照网络中出现和消失的时刻反映节点的活跃范围,而出现的频次则代表着节点的活跃程度。起始时刻的映射矩阵 $\mathbf{P}_b$ 通过余弦映射得到,节点 $v_i$ 的起始时刻映射为:

$$\mathbf{P}_{b,i} = \cos\left(\frac{t_{b,i}}{T}\right) \quad (8)$$

其中, $t_{b,i}$ 表示节点 $v_i$ 最初出现的时刻。该映射刻画了越晚出现的节点对网络的长期贡献越小的特性。

同理,消失时刻映射矩阵 $\mathbf{P}_e$ 通过正弦映射得到。节点 $v_i$ 的消失时刻映射为:

$$\mathbf{P}_{e,i} = \sin\left(\frac{t_{e,i}}{T}\right) \quad (9)$$

其中, $t_{e,i}$ 表示节点 $v_i$ 最后出现的时刻。该映射刻画了越晚消失的节点对网络的长期贡献越大的特性。

节点在快照网络中的出现频次被 $\mathbf{P}_n$ 记录,即 $\mathbf{P}_{n,i} = |\{t \in [0, T] | v_i \in \mathcal{V}^t\}|$ 。最终拼接得到时间相对位置矩阵 $\mathbf{P}_t$ :

$$\mathbf{P}_t = \mathbf{P}_b \parallel \mathbf{P}_e \parallel \mathbf{P}_n \quad (10)$$

#### 4.2 特征编码

特征编码模块由一个特征融合层构成,其首先将提取的 4 种特征拼接:

$$\mathbf{X}' = \mathbf{I}' \parallel \mathbf{I}^s \parallel \mathbf{P}_s' \parallel \mathbf{P}_t \quad (11)$$

通过融合函数 $f_{\text{con}}$ 对每个快照得到的特征矩阵 $\mathbf{X}'$ 进行特征融合,得到特征编码模块的输出 $\mathbf{H}_i'$ :

$$\mathbf{H}_i' = f_{\text{con}}(\mathbf{X}') = \tanh(\mathbf{X}' \mathbf{W}_c^{(0)} + \mathbf{b}_c^{(0)}) \mathbf{W}_c^{(1)} + \mathbf{b}_c^{(1)} \quad (12)$$

其通过可学习的权重矩阵 $\mathbf{W}_c^{(0)}$ 和 $\mathbf{W}_c^{(1)}$ 以及偏置项 $\mathbf{b}_c^{(0)}$ 和 $\mathbf{b}_c^{(1)}$ 进行调节。最终得到了特征表示集合 $\mathbf{H}_s = \{\mathbf{H}_s^0, \mathbf{H}_s^1, \dots, \mathbf{H}_s^{\tau-1}\}$ 。

#### 4.3 双层演化学习

##### 4.3.1 学习模块

第一组演化单元的输入是从特征编码得到的节点特征表示 $\mathbf{H}_s = \{\mathbf{H}_s^0, \mathbf{H}_s^1, \dots, \mathbf{H}_s^{\tau-1}\}$ ,后续演化单元的输入为前一组节点表示经过监督模块的修正节点表示,即第  $m$  组的输入为 $\mathbf{H}_{o(m-1)}$ 。

为了通过 NODE 计算所需快照的时序表示,需要将离散时间表示 $\mathbf{H}_s$ 转化为连续时间表示。采用插值投影,可得到时

刻  $t$  的连续时间表示 $\mathbf{H}_{ip}^t$ :

$$\mathbf{H}_{ip(m-1)}^t = \frac{\lceil t \rceil_e - t}{\lceil t \rceil_e - \lfloor t \rfloor_e} \mathbf{H}_{ip(m-1)}^{\lfloor t \rfloor_e} + \frac{t - \lfloor t \rfloor_e}{\lceil t \rceil_e - \lfloor t \rfloor_e} \mathbf{H}_{ip(m-1)}^{\lceil t \rceil_e} \quad (13)$$

其中, $\lfloor t \rfloor_e$ 适应网络快照缺失的情况,自动向小于  $t$  的非负整数进行检索,直到对应离散时刻的表示存在时停止。同理, $\lceil t \rceil_e$ 自动向大于  $t$  的非负整数进行检索。当  $t \leq 0$  时,映射为 0 时刻的表示,即 $\mathbf{H}_{ip(m-1)}^t = \mathbf{H}_{ip(m-1)}^0$ ;而当  $t \geq \tau - 1$  时,映射为  $\tau - 1$  时刻的表示,即 $\mathbf{H}_{ip(m-1)}^t = \mathbf{H}_{ip(m-1)}^{\tau-1}$ 。

随后,使用注意力机制构建 NODE:

$$f_{\text{ode}}(\mathbf{H}_{(m)}, t, \boldsymbol{\theta}) = \text{softmax}\left(\frac{\mathbf{Q}_s \mathbf{K}_s^T}{\sqrt{d_k}}\right) \mathbf{V}_s^T \quad (14)$$

$$\mathbf{Q}_s = \mathbf{H}_{ip(m-1)} \mathbf{W}_q, \mathbf{K}_s^T = \mathbf{H}_{ip(m-1)}^T \mathbf{W}_k, \mathbf{V}_s^T = \mathbf{H}_{(m)} \quad (15)$$

其中,NODE 函数旨在拟合时刻  $t$  的表示 $\mathbf{H}_{(m)}^t$ ,即动态系统在  $t$  时刻的状态。 $\top$ 表示矩阵转置, $d_k$ 是表示的维度, $\boldsymbol{\theta}$ 是包含了权重矩阵 $\mathbf{W}_q$ 和 $\mathbf{W}_k$ 可学习的参数集合。该注意力机制的本质是基于上组学习模块所得表示 $\mathbf{H}_{ip(m-1)}^t$ 求得各节点间的相似度,即注意力系数,再对本组学习模块所得表示 $\mathbf{H}_{(m)}^t$ 进行加权求和,从而令具有相似结构特征的节点间信息进一步融合。

然后通过单层 MLP 获得整个系统的初始状态,即通过 $f_{\text{mlp}}(\mathbf{H}_{ip(m-1)}^0)$ 得到了在 $t_0$ 时刻的输入,最终 NODE 构建为:

$$\mathbf{H}_{(m)}^t = f_{\text{mlp}}(\mathbf{H}_{ip(m-1)}^0) + \int_0^{t_i} f_{\text{ode}}(\mathbf{H}_{(m)}, t, \boldsymbol{\theta}) dt \quad (16)$$

定义 $\mathcal{T}$ 为所求表示对应的离散时刻集合, $\mathcal{T} = \{0, 1, \dots, \tau + l - 1\}$ , $t_i \in \mathcal{T}$ 。通过式(16)最终可以计算得到特定时刻的时序表示。本文采用五阶龙格-库塔法<sup>[21]</sup>作为常微分方程的求解器。学习模块最终得到了第  $m$  组的节点时序特征表示 $\mathbf{H}_{(m)} = \{\mathbf{H}_{(m)}^0, \mathbf{H}_{(m)}^1, \dots, \mathbf{H}_{(m)}^{\tau-1}\}$ 。

##### 4.3.2 节点活跃度模块

节点活跃度模块输入为前一组节点活跃度模块得到的活跃度表示 $\mathcal{A}_{(m-1)}$ 和同组学习模块得到的节点时序特征表示 $\mathbf{H}_{(m)}$ 。模块通过式(17)将输入的节点活跃度 $\mathcal{A}_{(m-1)}$ 也映射到连续时间:

$$\mathcal{A}_{ip(m-1)}^t = \frac{\lceil t \rceil_e - t}{\lceil t \rceil_e - \lfloor t \rfloor_e} \mathcal{A}_{ip(m-1)}^{\lfloor t \rfloor_e} + \frac{t - \lfloor t \rfloor_e}{\lceil t \rceil_e - \lfloor t \rfloor_e} \mathcal{A}_{ip(m-1)}^{\lceil t \rceil_e} \quad (17)$$

随后使用连续空间的节点活跃度 $\mathcal{A}_{ip(m-1)}^t$ 和特征表示 $\mathbf{H}_{ip(m)}$ 构建注意力机制,得到节点活跃度的 NODE:

$$f_a(\mathcal{A}_{(m)}^t, t, \boldsymbol{\theta}) = \text{softmax}\left(\frac{\mathbf{Q}_a \mathbf{K}_a^T}{\sqrt{d_k}}\right) \mathbf{V}_a^T \quad (18)$$

$$\mathbf{Q}_a = \mathbf{H}_{ip(m)} \mathbf{W}_q, \mathbf{K}_a^T = \mathcal{A}_{(m)}^t, \mathbf{V}_a^T = \mathcal{A}_{ip(m-1)}^t \mathbf{W}_v \quad (19)$$

其中, $\mathbf{H}_{ip(m)}$ 由式(13)得到, $\mathcal{A}_{(m)}^t$ 为常微分方程拟合的节点活跃度表示。该注意力机制本质上构建了同组单元学得表示 $\mathbf{H}_{ip(m)}$ 与活跃度 $\mathcal{A}_{(m)}^t$ 的关联,以及不同组活跃度间的映射。随后,通过 MLP 获得节点活跃度表示的初始状态,即通过 $f_{\text{mlp}}(\mathcal{A}_{(m-1)}^0)$ 得到了节点活跃度在 $t_0$ 时刻的输入,最终节点活跃度 NODE 构建为:

$$\mathcal{A}_{(m)}^t = f_{\text{mlp}}(\mathcal{A}_{(m-1)}^0) + \int_0^{t_i} f_a(\mathcal{A}_{(m)}, t, \boldsymbol{\theta}) dt \quad (20)$$

节点活跃度模块旨在从节点特征中学习节点活跃度的表示,并在后续优化过程中要求表示尽可能地拟合原始活跃度,使得通过式(20)的 NODE 可以递推各时刻的活跃度。这是因为节点历史邻居信息可能随时间积累,导致未来快照表示

空间中大量节点聚集,难以基于节点表示相似性对边精准推断。DANOM 通过节点活跃度的监督,约束未来快照表示空间中节点附近相似节点的数量,提升了表示整体的均匀性,从而有利于边的推断。

#### 4.3.3 监督模块

DANOM 通过节点活跃度监督模型时序信息学习并修正节点表示,引入了监督模块。监督模块分别对每个快照表示进行监督,先将节点活跃度表示  $\mathcal{A}_{(m)}^t$  和节点表示  $\mathbf{H}_{(m)}^t$  融合:

$$\mathbf{H}_{\text{con}(m)}^t = \mathcal{A}_{(m)}^t \cup \mathbf{H}_{(m)}^t \quad (21)$$

其中,  $\cup$  表示将 1 维节点活跃度表示加在节点表示的每一维上。然后分别通过两层全局图卷积和两层的局部图卷积得到辅助矩阵  $\mathbf{H}_{a(m)}^g$  和  $\mathbf{H}_{a(m)}^l$ :

$$\mathbf{H}_{a(m)}^g = \text{GCN}^{(1)}(\mathbf{A}^g, \text{GCN}^{(0)}(\mathbf{A}^g, \mathbf{H}_{\text{con}(m)}^t)) \quad (22)$$

$$\mathbf{H}_{a(m)}^l = \text{GCN}^{(1)}(\mathbf{A}^l, \text{GCN}^{(0)}(\mathbf{A}^l, \mathbf{H}_{\text{con}(m)}^t)) \quad (23)$$

其中,  $\mathbf{A}^g$  为全局网络的邻接关系。之后计算得到监督矩阵  $\mathbf{O}_{(m)}^t$ :

$$\mathbf{O}_{(m)}^t = \sigma(\mathbf{H}_{a(m)}^g \oplus \mathbf{H}_{a(m)}^l) \quad (24)$$

其中,  $\oplus$  表示矩阵加法,  $\sigma$  表示 Sigmoid 激活函数。通过监督矩阵加权节点表示和节点活跃度表示,实现节点活跃度表示对节点表示的修正:

$$\mathbf{H}_{o(m)}^t = \mathbf{O}_{(m)}^t \otimes \mathcal{A}_{(m)}^t \oplus (1 - \mathbf{O}_{(m)}^t) \otimes \mathbf{H}_{(m)}^t \quad (25)$$

其中,  $\otimes$  为哈达玛积,  $\mathbf{H}_{o(m)}^t$  为第  $m$  组演化单元修正后时刻  $t$  的节点表示。

最终将第  $M$  组的节点表示  $\mathbf{H}_{(M)}$  和节点活跃度表示  $\mathcal{A}_{(M)}$  作为 DANOM 最终的学习结果进行优化,即节点表示  $\mathbf{Z} = \mathbf{H}_{(M)}$ , 节点活跃度  $\mathbf{Z}_a = \mathcal{A}_{(M)}$ 。

#### 4.4 模型优化

整个模型优化包括对节点表示和节点活跃度表示进行优化。节点表示优化包含快照重构和快照平滑,节点活跃度表示优化包含活跃度修正和活跃度重构。

为了使节点表示能够保留其邻域节点的信息,受 DySAT<sup>[17]</sup> 启发,在每个时刻采用二元交叉熵函数得到快照重构损失:

$$\mathcal{L}_r = \sum_{t=0}^{T-1} \sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{P}_i^t} (-\log \sigma(\langle \mathbf{z}_i^t, \mathbf{z}_j^t \rangle)) - \alpha \sum_{j' \in \mathcal{N}_i^t} \log(1 - \sigma(\langle \mathbf{z}_i^t, \mathbf{z}_{j'}^t \rangle)) \quad (26)$$

其中,  $\mathbf{z}_i^t = \mathbf{Z}_{i,t}^t$ ,  $\langle \mathbf{a}, \mathbf{b} \rangle$  表示  $\mathbf{a}$  和  $\mathbf{b}$  的内积。  $\mathcal{P}_i^t$  表示在时刻  $t$  通过随机游走获得的节点  $v_i$  的正样本集合。这种方法相较于直接将邻居  $\Gamma_i^t$  作为正样本,对数据中的噪声有更好的抗性,因为在均匀噪声分布下,  $\mathcal{P}_i^t$  相较于  $\Gamma_i^t$  的相对变化更小。负样本  $\mathcal{N}_i^t$  使用度分布进行采样。超参数  $\alpha$  调节正负采样之间的贡献值。

保持两个相邻快照表示的平滑是本模型的另一个优化目标,该目标可以惩罚网络中突变的节点表示,保证网络时序信息学习的稳定性。使用 L1 范数实现快照平滑损失:

$$\mathcal{L}_s = \sum_{t=1}^{T-1} \sum_{i \in \mathcal{V}} \|\mathbf{z}_i^t - \mathbf{z}_i^{t-1}\|_1 \quad (27)$$

活跃度修正期望学习到的活跃度能尽可能拟合原始活跃度分布。DANOM 通过 MSE 损失函数实现活跃度修正损失:

$$\mathcal{L}_{ac} = \sum_{t=1}^{T-1} \sum_{i \in \mathcal{V}} (\mathbf{a}_i^t - \mathbf{z}_{a,i}^t)^2 \quad (28)$$

活跃度重构旨在对节点表示进行约束。这种约束会最小化两者之间的差距,确保节点表示与节点活跃度表示学习到相同的现实意义。DANOM 通过 MSE 损失函数实现活跃度重构损失:

$$\mathcal{L}_{ar} = \sum_{t=1}^{T-1} \sum_{i \in \mathcal{V}} \sum_j (\sum_j (\mathbf{Z}^t \mathbf{Z}^{t^T})_{ij} - \mathbf{z}_{a,i}^t)^2 \quad (29)$$

其中,  $\mathbf{Z}^t \mathbf{Z}^{t^T}$  的第  $i$  行和代表通过节点表示计算的  $v_i$  节点活跃度。

最后融合 4 部分损失,通过超参数  $\beta, \gamma$  和  $\delta$  来调整各部分损失的贡献度:

$$\mathcal{L} = \mathcal{L}_r + \beta \mathcal{L}_s + \gamma \mathcal{L}_{ac} + \delta \mathcal{L}_{ar} \quad (30)$$

#### 4.5 模型扩展

属性网络和权重网络也是常见的动态网络类型。本文将 DANOM 模型通过以下扩展方式,以利用这些网络的丰富信息。

##### 4.5.1 属性网络

对于属性网络,在式(11)中可将节点属性  $\mathbf{F}$  与提取的结构特征拼接,即:

$$\mathbf{X}^t = \mathbf{F} \parallel \mathbf{I}^t \parallel \mathbf{I}^g \parallel \mathbf{P}_s^t \parallel \mathbf{P}_t \quad (31)$$

随后通过式(12)定义的融合函数  $f_{\text{con}}$  对属性和特征进行融合。

##### 4.5.2 权重网络

权重网络上 DANOM 涉及边权的处理过程包括部分特征提取、监督模块和模型优化。其中,相关特征的提取和监督模块的 GCN 都有权重网络版本。因此,DANOM 扩展到权重网络的难点在于优化时损失函数的设计。  $\mathbf{A}^t$  中的 0 值不仅是权重,还代表了边不存在,是网络稀疏特性的表现。此时链路预测既是分类任务,也是回归任务。因此,在式(30)中添加基于采样计算的边权 MSE 损失  $\mathcal{L}_w$ :

$$\mathcal{L}_w = \sum_{t=0}^{T-1} \sum_{i \in \mathcal{V}} \sum_{j \in \Gamma_i^t} (\sum_{j \in \Gamma_i^t} (\mathbf{A}_{i,j}^t - f_w(\mathbf{z}_i^t \parallel \mathbf{z}_j^t)))^2 - \alpha \sum_{j' \in \mathcal{N}_i^t} (0 - f_w(\mathbf{z}_i^t \parallel \mathbf{z}_{j'}^t))^2 \quad (32)$$

其中,  $f_w$  为一个多层感知机构成的回归器。边权预测综合分类和回归结果,即  $f_w(\mathbf{z}_i^t \parallel \mathbf{z}_{j'}^t) \cdot \langle \mathbf{z}_i^t, \mathbf{z}_{j'}^t \rangle$ 。实际处理时,先将网络中的权重矩阵  $\mathbf{A}^t \in \mathbb{R}^{N \times N}$  线性映射到绝对值为  $[0, 1]$  的范围内,避免由于过大/小的元素阻碍模型训练,之后将预测值逆映射到原取值空间。可额外选取一个较小阈值  $\epsilon$ ,令处于范围  $[-\epsilon, \epsilon]$  的预测结果为 0,以保证网络的稀疏性。

## 5 实验分析

### 5.1 实验设置

#### 5.1.1 数据集

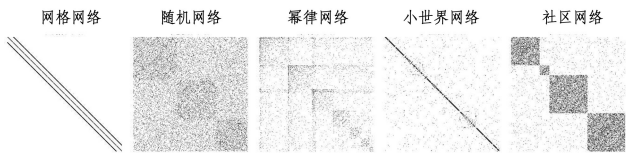
本文在 7 个真实数据集上进行实验,包括 Email-dnc, UCI, Bitcoin-alpha, Reality-call, Wikipedia, Reddit 和 Math(下文对部分名称使用缩写)。数据集的统计信息如表 1 所列。构建了 5 个生成网络以进行网络动力学实验<sup>[22]</sup>,其邻接关系如图 2 所示。网络网络中每个节点与 8 个相近节点相连;随机网络由 Erdős-Rényi 模型生成;幂律网络由 Albert-Barabási

模型生成;小世界网络由 Watts-Strogatz 模型生成;社区网络由随机块模型生成。每个网络包含 400 个节点。

表 1 真实数据集统计信息

Table 1 Statistics of real-world datasets

数据集	节点数	边数	快照中 最小边数	快照中 最大边数	快照 平均边数	快照数
Email	1 891	5 870	102	2 861	978	6
UCI	1 899	15 675	291	9 015	2 239	7
Bitcoin	3 783	14 380	945	3 113	1 798	8
Reality	6 809	16 623	607	1 351	924	18
Wiki	8 227	32 856	1 045	2 744	2 169	14
Reddit	10 985	292 222	9 015	11 869	10 489	27
Math	24 740	230 956	4 899	13 162	8 883	26



注:行/列元素为节点,黑色代表对应节点间存在边。

图 2 生成网络邻接关系

Fig. 2 Adjacencies in synthetic networks

### 5.1.2 对比方法

本文对 7 种动态网络链接预测算法进行了比较分析,包括 Dyngraph2vec(最优版本 DynAERNN)、EvolveGCN(最优版本 EvolveGCN-H)、ROLAND(最优版本为 ROLAND-GRU)、DySAT、STGSN、NDCN 和 GRU-GCN<sup>[23]</sup> 算法。这些算法覆盖了 3 类动态链路预测方法。

### 5.1.3 评价指标

采用 AUC(Area Under the Curve)和 AP(Average Precision)作为链路预测任务和用户缝合的评价指标。网络动力学实验指标采用 L1 范数。节点活跃度实验的评价指标为平均绝对误差(MAE)和均方误差(MSE)。

### 5.1.4 参数设置

除网络动力学实验的节点表示大小设置为 32 维外,其他节点表示的大小均设置为 256 维。DANOM 使用 Adam 优化器,学习率设置为 0.01,权重衰减为 0.0005,丢弃率为 0.05,默认使用 2 组双层演化学习单元。快照重构损失中的随机游走长度为 20,每个正边的负样本数量为 10 个。调整损失超参数  $\beta$ ,以保持训练过程中各部分损失处于同一数量级。具体地,对于数据集 Email,UCI,Bitcoin,Reality,Wiki,Reddit 和 Math,超参数  $\beta$  分别为设为 10,10,10,10,1,0.0001 和 0.00001;超参数  $\gamma$  和  $\delta$  设为 0.1。所有实验记录在 10 个随机种子上的实验结果,并求平均结果。

### 5.2 单步链路预测

单步链路预测实验在全部数据集上进行。所有模型在快照集  $\{g^0, g^1, \dots, g^{T-1}\}$  上训练,其中基于网络动力学的模型 DANOM 和 NDCN 因为具备演化预测能力,直接使用  $g^T$  生成的节点表示进行链路预测,而其他模型使用  $g^{T-1}$  的节点表示进行链路预测。按照 1:1:2 的比例将  $g^T$  全部链接以节点对的形式划分为训练集、验证机和测试集,并随机抽取同等数量的负样本(无链接的两个节点)。随后训练一个逻辑回归器,将节点表示的哈达玛积作为边的表示输入逻辑回归器进行预测。预测结果以平均值  $\pm$  标准差的形式在表 2 中列出,其中粗体表示最优结果,下划线表示次优结果。DANOM 整体上优于全部基线方法,高于基线方法 1%~9%。在较大规模数据集(Reality,Wiki,Reddit 和 Math)上,大部分方法均有不错的效果,在规模较小的数据集上对比方法表现差异较大,这是由于部分方法依赖更多的训练样本和历史信息。而 DANOM 均保持最优性能,说明模型通过神经常微分方程能基于不同环境学习到真实的演化模式。同时,DANOM 在稀疏图 Email 和 Reality 上优势显著,因为其多元的特征提取方法,能保证在稀疏图上也能捕捉到充分的结构信息。

表 2 单步链路预测结果

Table 2 Prediction results of single-step link

(%)

方法	指标	Email	UCI	Bitcoin	Reality	Wiki	Reddit	Math
Dyngraph2vec	AUC	97.6 $\pm$ 0.9	87.0 $\pm$ 1.1	89.6 $\pm$ 0.7	96.4 $\pm$ 0.4	93.4 $\pm$ 0.2	<u>96.7 <math>\pm</math> 0.1</u>	91.9 $\pm$ 1.5
	AP	97.9 $\pm$ 1.5	85.7 $\pm$ 1.6	89.5 $\pm$ 0.8	96.0 $\pm$ 0.2	92.5 $\pm$ 0.8	<u>95.1 <math>\pm</math> 0.6</u>	91.4 $\pm$ 1.2
EvolveGCN	AUC	96.9 $\pm$ 1.2	85.9 $\pm$ 2.1	92.2 $\pm$ 0.4	92.9 $\pm$ 0.9	88.7 $\pm$ 0.8	91.9 $\pm$ 0.7	90.4 $\pm$ 0.5
	AP	97.1 $\pm$ 1.5	87.7 $\pm$ 1.9	<u>93.3 <math>\pm</math> 0.6</u>	94.8 $\pm$ 0.9	92.0 $\pm$ 1.3	91.7 $\pm$ 0.7	90.2 $\pm$ 0.6
DySAT	AUC	95.9 $\pm$ 1.3	87.3 $\pm$ 1.8	87.8 $\pm$ 1.7	96.5 $\pm$ 3.6	89.3 $\pm$ 0.2	96.5 $\pm$ 0.3	93.3 $\pm$ 0.9
	AP	96.7 $\pm$ 1.8	84.0 $\pm$ 0.9	87.5 $\pm$ 3.6	95.5 $\pm$ 3.6	88.9 $\pm$ 0.9	95.1 $\pm$ 0.4	93.0 $\pm$ 0.2
NDCN	AUC	95.4 $\pm$ 1.2	80.4 $\pm$ 3.3	85.3 $\pm$ 1.4	96.7 $\pm$ 1.3	88.2 $\pm$ 1.5	95.3 $\pm$ 2.7	92.3 $\pm$ 0.8
	AP	96.3 $\pm$ 2.3	82.2 $\pm$ 1.8	87.4 $\pm$ 1.6	95.7 $\pm$ 2.7	89.5 $\pm$ 3.6	94.4 $\pm$ 2.0	92.4 $\pm$ 0.7
STGSN	AUC	<u>97.9 <math>\pm</math> 0.2</u>	<u>89.3 <math>\pm</math> 2.7</u>	82.9 $\pm$ 2.5	93.0 $\pm$ 1.2	<u>94.1 <math>\pm</math> 2.5</u>	94.3 $\pm$ 1.3	<u>95.6 <math>\pm</math> 0.4</u>
	AP	97.3 $\pm$ 0.4	88.0 $\pm$ 1.8	89.4 $\pm$ 4.4	90.1 $\pm$ 1.9	<u>94.1 <math>\pm</math> 0.9</u>	93.9 $\pm$ 1.1	<u>95.1 <math>\pm</math> 0.2</u>
GRU-GCN	AUC	95.0 $\pm$ 2.5	83.6 $\pm$ 1.6	91.0 $\pm$ 0.8	96.4 $\pm$ 0.6	88.7 $\pm$ 1.7	92.9 $\pm$ 1.1	91.1 $\pm$ 1.1
	AP	<u>98.0 <math>\pm</math> 1.1</u>	86.5 $\pm$ 2.6	93.1 $\pm$ 0.3	<u>96.1 <math>\pm</math> 0.8</u>	91.1 $\pm$ 0.9	91.2 $\pm$ 0.5	90.2 $\pm$ 0.7
ROLAND	AUC	96.5 $\pm$ 1.8	88.5 $\pm$ 0.9	<u>94.7 <math>\pm</math> 1.6</u>	93.1 $\pm$ 0.4	92.7 $\pm$ 1.4	95.0 $\pm$ 0.9	91.6 $\pm$ 1.3
	AP	96.8 $\pm$ 1.3	<u>88.1 <math>\pm</math> 2.2</u>	92.4 $\pm$ 2.4	95.1 $\pm$ 1.6	91.0 $\pm$ 1.4	94.2 $\pm$ 2.9	91.8 $\pm$ 1.8
DANOM	AUC	<b>99.2 <math>\pm</math> 0.4</b>	<b>90.1 <math>\pm</math> 0.9</b>	<b>96.9 <math>\pm</math> 0.4</b>	<b>98.2 <math>\pm</math> 0.1</b>	<b>95.4 <math>\pm</math> 0.3</b>	<b>97.0 <math>\pm</math> 0.3</b>	<b>96.0 <math>\pm</math> 0.4</b>
	AP	<b>98.9 <math>\pm</math> 0.5</b>	<b>89.3 <math>\pm</math> 0.4</b>	<b>94.8 <math>\pm</math> 0.7</b>	<b>97.9 <math>\pm</math> 0.9</b>	<b>94.4 <math>\pm</math> 0.9</b>	<b>96.1 <math>\pm</math> 0.6</b>	<b>95.8 <math>\pm</math> 0.4</b>

注:AUC/AP 值越大,预测越准确。

为验证 DANOM 模型的鲁棒性,分别构建含噪声和复杂结构的动态网络数据进行单步链路预测。在 Reality 和 Wiki 数据集上,对历史快照随机删除或添加 5% 的边,得到含噪声

网络;采用 Albert-Barabási 模型和动态随机块模型分别得到具有幂律特性和社团结构的网络。对比方法选择在表 2 中保持稳定较好结果的 Dyngraph2vec 和 STGSN 算法。实验结果

如表 3 和表 4 所列。

表 3 含噪声网络上单步链路的预测结果

Table 3 Prediction results of single-step link on noisy networks

(%)			
方法	指标	Reality	Wiki
Dyngraph2vec	AUC	<u>95.5±0.9</u>	92.6±1.2
	AP	<u>95.0±0.6</u>	91.4±1.3
STGSN	AUC	91.3±1.5	<u>93.4±1.3</u>
	AP	89.2±1.6	<u>92.9±1.4</u>
DANOM	AUC	<b>98.2±0.5</b>	<b>94.1±0.8</b>
	AP	<b>97.6±0.9</b>	<b>93.4±1.3</b>

注: 值越大越准确; 加粗为最优结果; 下划线为次优结果; 边扰动率为 5%。

表 4 复杂结构网络上单步链路的预测结果

Table 4 Prediction results of single-step link on complex structured networks

(%)			
方法	指标	动态幂律网络	动态社团网络
Dyngraph2vec	AUC	97.5±1.4	83.1±2.1
	AP	96.1±1.8	<u>78.1±0.9</u>
STGSN	AUC	97.0±1.5	76.1±1.4
	AP	96.9±1.6	70.1±2.3
DANOM	AUC	<b>98.4±0.4</b>	<b>86.4±0.7</b>
	AP	<b>97.3±2.1</b>	<b>79.3±0.3</b>

注: 值越大越准确; 加粗为最优结果; 下划线为次优结果。

在含噪声网络中, DANOM 模型的效果损失最少, 这得益

于基于随机游走的正样本采样; 在复杂结构网络中, DANOM 也均保持最优效果, 这说明本文采取的特征学习和双层演化单元有效应对了复杂结构的挑战。

### 5.3 多步链路预测

本节通过多步链路预测实验进一步验证模型学习真实演化规律的能力。实验在快照数量大于 10 的 Reality, Wiki, Reddit 和 Math 数据集上进行。实验在快照集合  $\{g^0, g^1, \dots, g^{r-1}\}$  上训练模型, 然后基于网络动力学模型 DANOM 和 NDCN 直接输出  $\{g^r, g^{r+1}, \dots, g^{r+4}\}$  的节点表示进行链路预测, 其他方法则使用  $g^{r-1}$  的节点表示来预测这 5 个快照中的链接。对于每个快照, 使用 4.2 节中提到的相同的评估方式, 实验结果如图 3 所示, 其中快照标签代表预测步数。DANOM 在全部数据集上显示出优越的结果, 通过演化推理得到的预测结果在多个时刻均保持高水平, 其预测性能随预测快照时刻的增加下降幅度最小。同时, NDCN 相对较低的结果说明动态推理需要全面的静态结构信息学习作为支撑。在 Reality 数据集上, DANOM 对第 4 个目标快照预测效果相较于第 3 个目标快照有所提高, 且部分对比算法对该快照的预测也均有提高, 说明该快照与第 1 个目标快照存在相似的邻接信息; 而 DANOM 对第 5 个目标快照的预测效果相较于第 1 个目标快照下降幅度最小, 说明其通过神经微分方程的演化推理, 能进一步细化各时刻间的依赖关系, 从而得到了更好的预测结果。

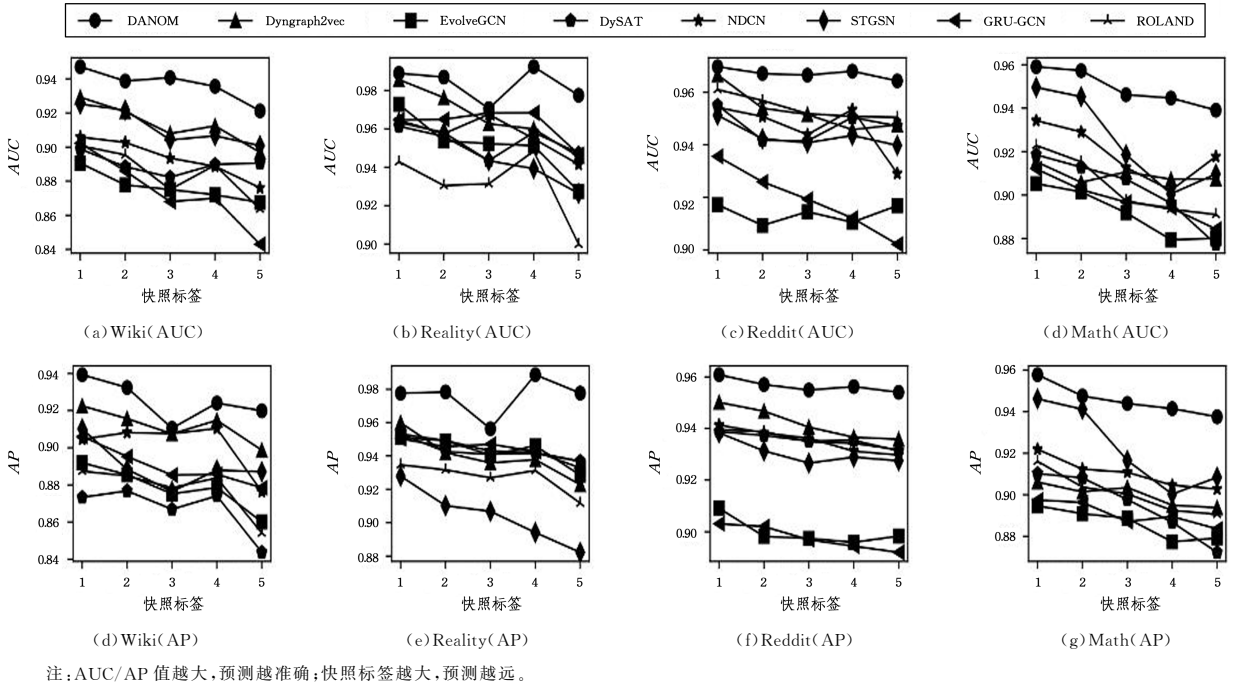


图 3 多步链路预测结果

Fig. 3 Prediction results of multi-step link

### 5.4 快照缺失链路预测

为了验证 DANOM 模型能否摆脱对最近历史快照的强依赖, 学习到真实的网络演化规律, 对 DANOM 和基线模型进行了快照缺失链路预测实验。实验在 7 个数据集上进行。通过移除每个数据集 20% 的快照 (向上取整) 来模拟快照缺失, DANOM 直接在缺少快照的情况下进行训练和预测。对于基线方法, 使用邻近未移除快照的加权组

合填补空缺, 使基线方法可以正常训练与预测。填补快照的具体构建操作为: 保留邻近两个未移除快照的公共边, 并根据它们与填补快照时间距离, 按比例抽样非公共边。快照缺失链路预测的实验结果如表 5 所列, 最后一列显示了每个算法在全部数据集上的平均下降程度。可以看出基线方法平均下降值均高于 DANOM, 表明本文提出的模型能有效处理数据不完整的情况, 学习到了远期网络的

演化规律。此外,DANOM在部分数据集上的预测效果相对于快照完整时还有略微提升,这表明后续演化规律并不完全

由近期快照的信息决定。合理利用整个历史快照的长期信息对链路预测十分重要。

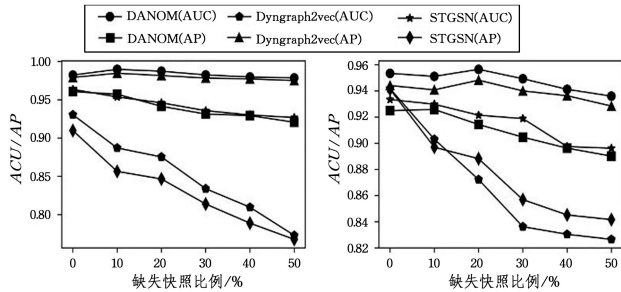
表5 快照缺失链路预测结果

Table 5 Prediction results of snapshot-scarce link

方法	指标	Email	UCI	Bitcoin	Reality	Wiki	Reddit	Math	Avg. Drop
Dyngraph2vec	AUC	97.3	82.1	87.9	94.6	92.2	95.3	88.3	2.12 ↓
	AP	97.7	82.6	87.7	94.2	91.4	93.0	87.8	1.96 ↓
EvolveGCN	AUC	96.0	84.3	75.7	91.6	88.6	90.3	88.0	3.49 ↓
	AP	96.1	84.1	78.8	93.9	90.9	90.1	88.0	3.56 ↓
DySAT	AUC	88.2	82.3	87.3	96.2	85.6	95.0	91.3	2.96 ↓
	AP	91.2	77.0	86.9	95.1	86.4	93.6	91.3	2.74 ↓
NDCN	AUC	94.4	79.2	81.5	88.9	82.8	94.2	91.2	3.06 ↓
	AP	95.1	80.8	82.8	89.0	82.1	93.4	90.6	3.44 ↓
STGSN	AUC	68.1	88.5	75.5	87.5	87.2	91.1	93.4	7.97 ↓
	AP	76.1	87.0	84.7	84.7	88.8	90.2	93.1	6.19 ↓
GRU-GCN	AUC	95.1	78.2	81.4	95.5	86.5	90.9	86.5	3.51 ↓
	AP	97.1	82.9	91.6	95.0	88.7	89.7	85.4	2.26 ↓
ROLAND	AUC	96.2	86.5	91.9	92.9	86.8	94.3	88.1	2.20 ↓
	AP	96.3	86.3	92.3	92.3	86.9	92.3	88.5	1.81 ↓
DANOM	AUC	<b>99.2</b>	<b>88.5</b>	<b>95.2</b>	<b>98.7</b>	<b>95.7</b>	<b>96.8</b>	<b>95.7</b>	<b>0.43 ↓</b>
	AP	<b>98.8</b>	<b>88.9</b>	<b>94.6</b>	<b>98.2</b>	<b>94.8</b>	<b>95.9</b>	<b>95.8</b>	<b>0.03 ↓</b>

注:值越大越准确;加粗为最优结果;下划线为次优结果;20%历史快照被删除。

为进一步验证 DANOM 模型在快照缺失情况下的稳健性,本节在不同缺失快照比例(0%~50%)的数据上进行实验验证,结果如图4所示。



注:AUC/AP 值越大预测越准确。

图4 不同比例缺失快照链路的预测结果

Fig. 4 Prediction results with different ratios of snapshot-scarce link

随着数据缺失比例的增大,其他方法的性能出现了明显的下降,尤其是 STGSN 算法在数据严重缺失的情况下几乎失效。而 DANOM 模型在数据严重缺失下依然保持高效,这

得益于基于注意力机制的神经常微分方程具有更好地模拟真实演化规律并自适应地推测缺失数据的能力。

## 5.5 网络动力学实验

为了验证模型能否捕获真实网络的演化规律,参考 NCDN<sup>[18]</sup>,在生成网络上设置了网络动力学实验。研究模型拟合热扩散、物种互利共生和基因调节动力学 3 种网络动力学的的能力。

在 5.1.1 小节中介绍的 5 个生成网络上通过模型计算 100 个时刻的系统状态,即 100 个时刻的最终节点表示矩阵  $\{Z^0, Z^1, \dots, Z^{99}\}$ ,在前 80 时刻进行训练,最后 20 个时刻做预测。100 个时刻的真实系统状态,由热扩散、互利共生和基因调节的动力学方程构建的 NODE 函数得到的预测值代替。通过预测与真实的节点表示之间的 L1 范数值进行评估,结果如表 6 所列,加粗为最优结果。每个值为最后 20 个快照的 L1 范数值之和,该值越小,拟合越准确。本实验对比了同样基于网络动力学的模型 NDCN,可以看出其挖掘动力学规律的能力不如 DANOM。

表6 网络动力学实验结果

Table 6 Experiment results of network dynamics

方程	方法	网格网络	随机网络	幂律网络	小世界网络	社区网络
热扩散	NDCN	1.07 ± 0.08	1.01 ± 0.15	12.45 ± 6.38	0.25 ± 0.03	4.96 ± 1.61
	DANOM	<b>0.81 ± 0.12</b>	<b>9.21 × 10<sup>-5</sup> ± 4.5 × 10<sup>-6</sup></b>	<b>2.33 ± 1.04</b>	<b>0.02 ± 0.02</b>	<b>3.76 × 10<sup>-4</sup> ± 1.0 × 10<sup>-4</sup></b>
互利共生	NDCN	5.86 ± 0.39	6.58 ± 2.27	18.36 ± 2.39	6.48 ± 1.88	2.23 ± 1.62
	DANOM	<b>2.08 ± 0.01</b>	<b>1.54 ± 1.42</b>	<b>4.92 ± 0.88</b>	<b>3.26 ± 0.17</b>	<b>1.07 ± 0.40</b>
基因调节	NDCN	1.97 ± 0.16	18.18 ± 3.66	11.33 ± 2.35	1.53 ± 0.09	16.79 ± 7.16
	DANOM	<b>1.68 ± 0.44</b>	<b>13.91 ± 1.77</b>	<b>2.62 ± 0.62</b>	<b>1.42 ± 0.15</b>	<b>10.38 ± 3.93</b>

## 5.6 节点活跃度实验

现实系统中个体的活跃度是有限且有规律的。为了验证学习到的节点活跃度表示是否有效,本节在 7 个数据集上进行节点活跃度实验。所有模型在快照集  $\{g^0, g^1, \dots, g^{t-1}\}$  上训练,得到最优的  $g^t$  节点表示对  $g^t$  的节点活跃度预测。DANOM 直接预测节点活跃度  $\mathcal{A}^t$ ,其他模型先通过预测得到的邻接关

系计算节点活跃度表示进行实验。评估过程首先将  $g^t$  的全部节点按照 1:1:2 的比例划分为训练集、验证集和测试集,随后训练一个逻辑回归器,以真实节点活跃度作为标签进行训练和预测,结果如表 7 所列,加粗为最优结果,下划线为次优结果。MSE/MAE 值越小,预测越准确。DANOM 在全部数据集上都保持最优的效果。在 Reddit 数据集上,不同快照中的

节点活跃度差异较大,导致全部基线方法预测较差,而 DAN-OM 在此条件下依旧能保持较好的预测效果。

表 7 节点活跃度预测结果  
Table7 Prediction results of node activity

方法	指标	Email	UCI	Bitcoin	Reality	Wiki	Reddit	Math
Dyngraph2vec	MSE	1.3942	4.3890	7.5754	2.4568	4.2313	74.1130	6.5226
	MAE	0.1766	0.8796	0.6612	0.2914	0.7084	1.7500	0.7769
EvolveGCN	MSE	1.4015	3.3035	6.8834	2.6477	4.2390	16.6760	6.7527
	MAE	0.1818	0.6379	0.6183	0.3029	0.7292	1.4490	0.7817
DySAT	MSE	2.5731	3.6712	14.6142	2.4305	3.2314	16.1950	5.1828
	MAE	0.1854	0.5965	0.7427	0.2796	0.6953	1.2560	0.6214
NDCN	MSE	1.5241	3.7251	6.5852	2.4565	3.3629	66.7230	5.8639
	MAE	0.1895	0.6845	0.6957	0.2885	0.62850	1.5280	0.6957
STGSN	MSE	1.2435	3.8787	9.8808	2.9735	3.8923	58.3810	5.3810
	MAE	0.1610	0.6307	0.6329	0.2887	0.6320	1.5090	0.6757
GRU-GCN	MSE	1.0640	3.7351	9.8384	2.9070	3.4703	68.6638	5.8726
	MAE	0.1443	0.6003	0.6511	0.3201	0.6172	1.6372	0.7006
ROLAND	MSE	2.9823	5.3970	10.2180	2.4723	3.2918	90.0580	4.9649
	MAE	0.1976	0.6103	0.6882	0.2707	0.6247	2.0104	0.5964
DANOM	MSE	<b>0.3362</b>	<b>3.1036</b>	<b>6.2850</b>	<b>2.3327</b>	<b>3.0795</b>	<b>7.2848</b>	<b>4.3851</b>
	MAE	<b>0.1200</b>	<b>0.5601</b>	<b>0.6055</b>	<b>0.2641</b>	<b>0.6052</b>	<b>0.9986</b>	<b>0.6083</b>

### 5.7 用户缝合实验

为了验证模型是否能处理用户的复杂行为,本节设计了用户缝合实验。首先处理数据集:随机选取 10% 的节点,为这些节点各设置一个影子节点,节点与其影子节点代表同样的现实意义(如表示同一用户的不同账号),其以节点对的形式存储,供后续评估使用。在每个快照中,让原节点的每条边与原节点相连、与影子节点相连和与两节点同时相连的概率为 6:3:1。处理后在快照集  $\{g^0, g^1, \dots, g^{r-1}\}$  上训练模型,并通过模型在  $g^r$  上的节点表示进行用户缝合预测。评估过程首

先将全部原节点和影子节点的节点对按照 1:1:2 的比例划分为训练集、验证集和测试集,并随机抽取同等数量的负样本(不同现实意义的两个节点构成的节点对)。随后训练一个逻辑回归器,将用户缝合转化为二分类任务。预测结果如表 8 所列,可以看出,DANOM 相比其他基线方法在用户缝合问题上更具优势。ROLAND 通过将静态图神经网络拓展到动态的方法能充分学习快照的静态信息,在用户缝合上也保持着不错的效果。NDCN 在此实验中处于中上水平,证明了网络动力学结构能有效处理复杂用户行为。

表 8 用户缝合实验结果  
Table 8 Experiment results of user stitching

方法	指标	Email	UCI	Bitcoin	Reality	Wiki	Reddit	Math
Dyngraph2vec	AUC	71.3±9.8	79.6±8.2	84.0±9.8	84.1±1.4	82.8±7.2	85.6±2.7	82.4±2.8
	AP	66.7±8.7	74.5±9.0	79.5±8.5	78.2±2.0	76.8±8.3	84.3±2.1	82.4±2.8
EvolveGCN	AUC	79.2±1.6	80.9±4.6	78.8±2.9	80.8±1.5	82.2±1.1	83.4±1.7	80.0±1.7
	AP	85.9±1.0	86.7±3.3	85.4±2.6	82.9±1.7	87.5±0.6	82.9±1.9	80.0±1.5
DySAT	AUC	82.4±4.2	76.6±6.0	84.2±4.9	90.9±1.7	89.4±2.7	85.8±2.2	88.2±2.5
	AP	75.9±6.8	68.5±9.2	88.7±8.7	90.5±0.8	85.4±4.6	78.0±3.4	87.2±1.4
NDCN	AUC	90.8±0.7	86.6±1.9	90.1±0.5	91.1±1.6	88.8±1.6	91.6±1.3	89.7±2.4
	AP	89.2±1.8	86.0±1.8	87.9±0.8	91.1±1.8	88.7±1.2	90.4±1.8	88.6±1.9
STGSN	AUC	82.3±5.0	87.0±3.2	86.2±1.2	86.2±1.0	88.6±0.8	92.1±2.1	83.6±2.2
	AP	80.5±5.3	86.1±3.0	84.3±2.0	82.6±1.9	86.2±1.3	91.7±2.9	81.7±1.8
GRU-GCN	AUC	77.3±3.9	71.5±5.0	77.9±4.2	79.2±1.1	71.0±2.7	82.3±2.7	81.8±1.2
	AP	75.9±4.8	68.2±6.6	72.9±5.8	74.3±1.9	63.4±3.6	81.3±1.9	80.7±2.6
ROLAND	AUC	87.2±4.1	81.0±2.2	90.2±0.8	91.1±1.8	84.4±1.5	93.2±1.4	92.0±2.6
	AP	89.4±3.2	80.6±2.6	90.3±0.8	90.8±1.1	84.1±1.1	92.0±1.9	90.2±1.2
DANOM	AUC	<b>91.9±1.8</b>	<b>87.4±3.4</b>	<b>91.6±1.6</b>	<b>92.3±1.6</b>	<b>89.6±1.6</b>	<b>94.3±1.5</b>	<b>93.4±1.4</b>
	AP	<b>91.1±2.7</b>	<b>86.6±6.1</b>	<b>89.0±1.6</b>	<b>91.4±2.1</b>	<b>89.9±1.4</b>	<b>93.6±1.1</b>	<b>92.3±2.8</b>

### 5.8 消融实验

为验证 DANOM 中不同模块对模型的影响,本节通过单步链路预测在 Email, UCI, Bitcoin 和 Wiki 这 4 个数据集上进行了消融实验。实验为模型构建了 8 种变体,分别验证监督模块、节点活跃度模块、快照节点重要性矩阵、全局节点重要性矩阵、结构相对位置矩阵、时序相对位置矩阵、活跃度修正损失和活跃度重构损失的作用。

消融实验的结果如表 9 所列。可以看出,不使用监督模

块会明显降低模型能力,说明多组演化单元间不通过监督模块的监督会使得模型过拟合,不利于模型学习。验证节点活跃度模块的有效性时,先移除节点活跃度模块,然后在每个监督模块使用原节点活跃度进行监督,此条件下实验效果会下降,因为原节点活跃度和节点表示不处于同一分布,使用处于同分布的节点活跃度表示进行监督效果更好。随后验证了 4 种特征信息的作用,可以看出,减少任何一种特征信息都会影响模型结果。缺少活跃度修正损失  $\mathcal{L}_{ac}$  和活跃度重构损失  $\mathcal{L}_{nr}$

同样会明显降低模型效果,说明保持节点活跃度和节点表示符合真实的节点度分布是必不可少的。

表9 DANOM 模型消融实验结果

Table 9 Ablation experimental results of DANOM model

数据集	指标	原型	消融实验结果 (%)							
			无监督模块	无活跃度模块	无 $I'$	无 $I^a$	无 $P'_i$	无 $P_t$	无 $\mathcal{L}_{ac}$	无 $\mathcal{L}_{ac}$
Email	AUC	<b>99.2</b>	95.1	98.1	98.9	98.5	98.8	98.8	98.7	98.3
	AP	<b>98.9</b>	95.5	97.9	98.7	98.7	98.7	98.7	98.4	98.2
UCI	AUC	<b>90.1</b>	86.9	86.1	86.9	89.0	87.8	86.3	87.5	86.6
	AP	<b>89.3</b>	84.6	84.4	86.9	87.5	86.2	85.6	86.6	85.4
Bitcoin	AUC	<b>96.9</b>	93.7	93.6	94.0	95.7	93.8	94.6	94.1	94.3
	AP	<b>94.8</b>	91.9	92.2	93.0	94.0	93.2	94.4	92.6	93.5
Wiki	AUC	<b>95.4</b>	89.2	89.6	94.0	94.7	94.0	94.3	94.4	93.9
	AP	<b>94.4</b>	89.6	87.1	93.4	94.1	94.0	94.1	93.7	92.6

### 5.9 参数敏感性

参数敏感性实验通过单步链路预测任务分别验证 DANOM 演化单元组数  $M$ 、表示维度和 3 个损失超参数的敏感性。实验固定其他参数,调整一个参数的取值并观测结果。

不同演化单元组数的实验结果如图 5(a)所示。模型在 Reality 和 Wiki 数据集上均呈现实验效果随组数先升高再下降的现象,在 2 到 3 组演化单元时具有相对较好的效果。

不同表示维度的实验结果如图 5(b)所示。节点在 256

维时,在 Wiki 数据集上获得最佳效果,在 Reality 数据集上获得次优结果,说明模型学习 256 维节点表示时能保持良好稳定的效果。

超参数  $\beta$ 、 $\gamma$  和  $\delta$  的敏感性实验结果分别如图 5(c)、图 5(d)和图 5(e)所示。随着超参数  $\beta$ 、 $\gamma$  和  $\delta$  的增大,对应调节的快照平滑  $\mathcal{L}_s$ 、活跃度修正损失  $\mathcal{L}_{ac}$  和活跃度重构损失  $\mathcal{L}_{ar}$  也增大。从实验结果中可以看出,模型效果均出现先提升后下降的趋势,当 4 种损失处于同一数量级时模型效果最佳。

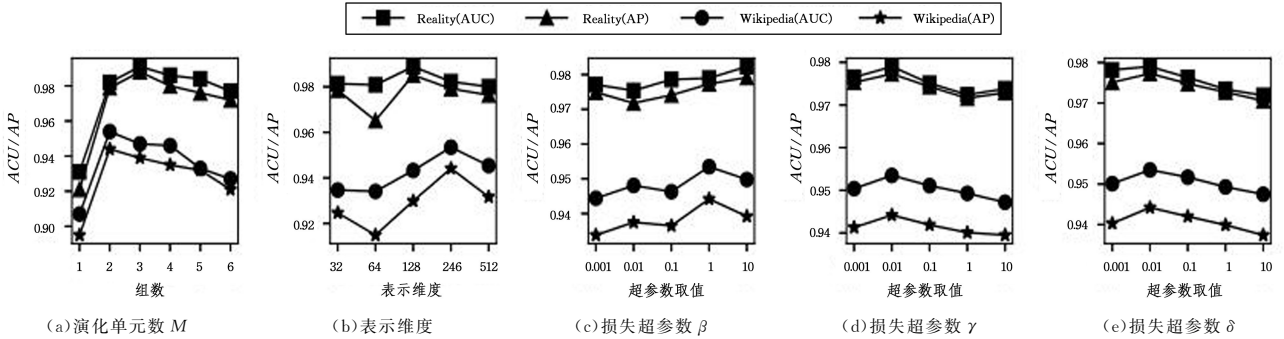


图5 DANOM 模型参数敏感性实验结果

Fig. 5 Experiment results of parameter sensitivity for DANOM model

**结束语** 本文提出了一种动态链路预测模型——双层活跃度约束神经常微分方程模型(DANOM)。DANOM 通过使用双层演化单元有效学习了动态网络的时序演化规律;此外,通过节点重要性和相对位置信息提取了更全面的结构特征,为模型学习节点表示提供了有效信息。DANOM 在多种链路预测任务上都保持了最优的效果,表明模型能学习到符合网络演化规律的节点表示。此外,在网络动力学、节点活跃度和用户缝合实验上的优异结果也验证了模型的有效性。DANOM 为动态网络的时序演化提供了一个改进的方案,并能有效处理数据不完整和用户行为复杂的情况。但 DANOM 仍存在改进空间,目前节点活跃度约束虽然有效,但约束方式相对单一,缺少对节点表示的多维度监督。未来研究可以考虑融合结构、属性和时序信息,从多方面监督节点表示的演化,使模型训练结果更符合现实网络的动态性。

### 参考文献

[1] CAO Z, FAN Z, WANG Q, et al. Link Prediction Algorithm Based on Denoising Autoencoder in Complex Networks[J].

Journal of Chinese Computer Systems, 2023, 44(3): 665-672.  
 [2] CAO Y, DONG Y H, WU S Q, et al. Advances in Dynamic Network Representation Learning Research [J]. Acta Electronica Sinica, 2020, 48(10): 2047-2059.  
 [3] QIN M, YEUNG D Y. Temporal Link Prediction: A Unified Framework, Taxonomy, and Review [J]. ACM Computing Surveys, 2023, 56(4): 1-40.  
 [4] ZHANG W, LAI X, WANG J. Social Link Inference via Multiview Matching Network from Spatio-Temporal Trajectories [J]. IEEE Transactions on Neural Networks and Learning Systems, 2020, 34(4): 1720-1731.  
 [5] PARK N, LIU F, MEHTA P, et al. Evokg: Jointly Modeling Event Time and Network Structure for Reasoning over Temporal Knowledge Graphs [C]// Proceedings of the 15th ACM International Conference on Web Search and Data Mining, 2022: 794-803.  
 [6] KING I J, HUANG H H. Euler: Detecting Network Lateral Movement via Scalable Temporal Link Prediction [J]. ACM Transactions on Privacy and Security, 2023, 26(3): 1-36.

- [7] QIN M, ZHANG C, BAI B, et al. High-Quality Temporal Link Prediction for Weighted Dynamic Graphs via Inductive Embedding Aggregation [J]. *IEEE Transactions on Knowledge and Data Engineering*, 2023, 35(9): 9378-9393.
- [8] GOYAL P, CHHETRI S R, CANEDO A. Dyngraph2vec: Capturing Network Dynamics Using Dynamic Graph Representation Learning [J]. *Knowledge-Based Systems*, 2020, 187: 104816.
- [9] MIN S, GAO Z, PENG J, et al. STGSN-A Spatial-Temporal Graph Neural Network Framework for Time-Evolving Social Networks [J]. *Knowledge-Based Systems*, 2021, 214: 106746.
- [10] CHEN R T, RUBANOVA Y, BETTENCOURT J, et al. Neural Ordinary Differential Equations [C] // *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, 2018: 6572-6583.
- [11] THOMAS N K, MAX W. Semi-Supervised Classification with Graph Convolutional Networks [C] // *Proceedings of the 5th International Conference on Learning Representations*, 2017: 1-10.
- [12] GOYAL P, KAMRA N, HE X, et al. DynGEM: Deep Embedding Method for Dynamic Graphs [C] // *IJCAI International Workshop on Representation Learning for Graphs*, 2018: 1-8.
- [13] GAO C, ZHU J, ZHANG F, et al. A Novel Representation Learning for Dynamic Graphs Based on Graph Convolutional Networks [J]. *IEEE Transactions on Cybernetics*, 2022, 53(6): 3599-3612.
- [14] PAREJA A, DOMENICONI G, CHEN J, et al. Evolvegen: Evolving Graph Convolutional Networks for Dynamic Graphs [C] // *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, 2020: 5363-5370.
- [15] YOU J, DU T, LESKOVEC J, ROLAND; Graph Learning Framework for Dynamic Graphs [C] // *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022: 2358-2366.
- [16] LEI K, QIN M, BAI B, et al. GCN-GAN: A Non-Linear Temporal Link Prediction Model for Weighted Dynamic Networks [C] // *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, 2019: 388-396.
- [17] YANG M, LIU J, CHEN L, et al. An Advanced Deep Generative Framework for Temporal Link Prediction in Dynamic Networks [J]. *IEEE Transactions on Cybernetics*, 2019, 50(12): 4946-4957.
- [18] SANKAR A, WU Y, GOU L, et al. Dysat: Deep Neural Representation Learning on Dynamic Graphs via Self-Attention Networks [C] // *Proceedings of the 13th International Conference on Web Search and Data Mining*, 2020: 519-527.
- [19] ZANG C, WANG F. Neural Dynamics on Complex Networks [C] // *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2020: 892-902.
- [20] JIN D, HEIMANN M, ROSSI R A, et al. Node2bits: Compact Time- and Attribute-Aware Node Representations for User Stitching [C] // *Proceedings of the 29th Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 2019: 483-506.
- [21] CALVO M, MONTIJANO J I, RANDEZ L. A Fifth-Order Interpolant for the Dormand and Prince Runge-Kutta Method [J]. *Journal of Computational and Applied Mathematics*, 1990, 29(1): 91-100.
- [22] PÓSFAI M, BARABASI A L. *Network Science* [M]. Cambridge, UK: Cambridge University Press, 2016: 1-57.
- [23] GAO J, RIBEIRO B. On the Equivalence between Temporal and Static Equivariant Graph Representations [C] // *Proceedings of the 39th International Conference on Machine Learning*. PMLR, 2022: 7052-7076.



**GUO Xuan**, born in 1996, postgraduate, is a member of CCF(No. U5970M). His main research interests include graph machine learning and complex network analysis.



**WANG Wenjun**, born in 1970, Ph.D, professor, Ph.D supervisor, is a member of CCF(No. 13864S). His main research interests include computational social sciences, big data mining and complex network analysis.

(责任编辑:何杨)