



17010226

## 上海交通大学博士学位论文

# 基于图深度学习的异常检测及动态关系建模

博士研究生：汪旭鸿

学 号：017032910027

导 师：杨煜普 教授

申 请 学 位：工学博士

学 科：控制科学与工程

所 在 单 位：自动化系

答 辩 日 期：2022 年 8 月 22 日

授予学位单位：上海交通大学



Dissertation Submitted to Shanghai Jiao Tong University  
for the Degree of Doctor

**DEEP GRAPH LEARNING BASED  
ANOMALY DETECTION AND DYNAMIC  
RELATION MODELING**

**Candidate:** Xuhong Wang  
**Student ID:** 017032910027  
**Supervisor:** Prof. Yupu Yang  
**Academic Degree Applied for:** Doctor of Engineering  
**Speciality:** Automation  
**Affiliation:** Department of Automation  
**Date of Defence:** August 22, 2022  
**Degree-Conferring-Institution:** Shanghai Jiao Tong University



17010226

# 基于图深度学习的异常检测及动态关系建模

## 摘要

为了满足复杂系统的建模需求，越来越多的复杂场景使用图（Graph）的数据形式描述大规模系统中实体之间的关系。近年来的研究热点——图深度学习技术——为深度挖掘图系统中蕴含的大量信息提供了行之有效的方法论。另一方面，随着现实中的通信网、金融交易网、社交网等复杂网络系统规模越来越大，系统中发生异常的情况也越来越频繁，精准地发现并管控异常是十分重要的。而由于系统的极度复杂化再加上异常的实体可以借由图系统内的实体关联影响其他正常的实体，无论是检测异常还是建模预测异常对其他实体的影响都显得尤为困难，而这恰恰急需图深度学习技术所带来的关系型建模能力。

然而，目前的图深度学习方法在动态异常检测和分析场景仍然存在以下的四个问题。1) 类别不平衡情况下的有偏估计问题。绝大部分图深度学习方法遵循着纯粹的有监督、无监督的学习范式，但图深度学习方法在数据不平衡情况下容易收敛到次优解。2) 时空协同性不足问题。由于时空联合建模产生的模型可行解空间过于庞大，绝大部分致力于推理节点之间关系的动态图方法仅仅只是分别考虑时间关系预测和空间关系预测，丧失了两者之间的协同性。3) 多模态适应能力不足问题。过去的工作使用一个单独的随机过程模型，仅仅从纯全局的视角试图概括整个图上的演化模式，忽略了图上不同区域之间可能存在不同的演化模式。4) 低实时性问题。图方法不光需要存储实体属性还要额外存储实体的关系，这导致了图算法对于数据的规模相较于欧氏算法更加敏感。本文分别从四个方面对这些问题进行了改进。

1) 针对图异常检测场景种数据不平衡导致的有偏估计问题，我们提出了一种新颖但简单的基于超球面学习的半监督图神经网络框架，将正常与异常的数据通过图网络映射用特征嵌入空间的超球面加以区分。既弥补了欧氏异常检测算法在提取图关系特征的不足，也克服了有监督图深度学习算法的有偏估计问题。此外，该框架可以兼容几乎所有的图神经网络计算模型，其使用范围可以从静态图泛化到动态图。并且该框架在带来了性能提升的同时，还比最先进的图异常检测方法计算复杂度更低、更易于使用。

2) 针对时空协同性不足问题，我们认为模型需要通过引入时空约束来压缩模型的可行解空间。而为了引入时空约束，模型必须要在同一个向量空间中联合地



17010226

表达时空关系，其难点在于节点在特征嵌入空间的位置信息不但要反映空间上的远近，还要反映时间上的交互关系频率。因此，我们提出了包含两种时空关系约束的新型时空建模方法以解决这个难点。首先是时空三角闭合约束，要求一个动态图事件“源节点-时间-目标节点”三元组在特征嵌入空间中要满足源节点向量 + 时间向量  $\approx$  目标节点向量。其次是时间向量范数单调性约束，该约束假设某个事件的时间距离当前时刻越远，该事件的时间在特征嵌入空间对应的时间向量的范数也要越大。这两种约束的引入可以保证节点在特征嵌入空间的位置关系可以同时包含时间和空间的双元语义信息。而在推理阶段，给定源节点，利用与其他节点之间的相对位置关系即可确认目标节点及其向量表示，再利用三角关系求得时间向量，最后再对时间向量作岭回归预测出具体的时间戳。本章动态图模型在动态关系预测任务中取得了更加优异的性能表现。

3) 针对关系演化预测模型在图上的应对不同演化模式时存在的多模态适应能力不足问题，我们采取“分群治之”的策略，将整图上的关系演化预测任务拆分为对各个社群分别进行演化预测，再使用一个额外的随机模型建模社群之间的交互。我们把社群内部的实体看成一个整体进行演化预测，将不同的社群看成具有局部性的不同个体，保证了对图上的全局和局部信息的建模能力。为了简化概率建模的复杂度，我们还提出了一种基于图神经网络的层次化随机时序点过程模型，我们针对图上的交互事件“源节点-时间-目标节点”建立一个层次化级联的贝叶斯条件概率链，先估计事件发生的时间，再分别估计事件的源节点和目标节点。该模型不光可以显式地根据不同的图社群采用不同的随机过程参数模型来预测社群未来的演化，而且该模型通过在编码器部分使用纯的注意力结构替代了传统随机过程模型中的循环神经网络，实现了大规模图数据中的高效率随机并行训练。

4) 针对图神经网络算法难以实时性部署的问题，我们经过对模型时间复杂度的系统性分析，认为图模型性能瓶颈在图数据库查询和图计算部分，而不在于模型的推理部分。因此，我们提出了异步信息传播注意力网络。其令图查询、计算阶段和模型推理阶段分别在异步离线链路和同步在线链路上解耦计算。这样繁重的图查询操作不会影响模型推理的速度，这进一步帮助模型获得更高的整体的稳定性和可扩展性。该异步计算机制可以适用于绝大多数的图深度学习任务和模型，而不仅仅是针对某一种情况进行加速。该框架可以有效地实现在线分布式图数据库中的超大规模图推理，从而推进了图深度学习在推荐系统、金融系统、社交网络等大规模图系统中的应用。

**关键词：**图神经网络，图深度学习，动态图，异常检测，关系推理，社群演化



17010226

# DEEP GRAPH LEARNING BASED ANOMALY DETECTION AND DYNAMIC RELATION MODELING

## ABSTRACT

To meet the modeling needs of complex systems, more and more complex scenarios start to use the graph based data form to describe the relationship between entities in these large-scale systems. The research hotspot in recent years - deep learning technology of graphs - provides an effective methodology for deep mining of a large amount of information contained in graph systems. On the other hand, as the scale of complex network systems such as communication networks, financial trading networks, and social networks in reality becomes larger and larger, and anomalies occur in the systems more and more frequently, it is very important to accurately detect and control anomalies. Due to the extreme complexity of the system and the fact that abnormal entities can affect other normal entities through entity associations in the graph system, it is particularly difficult to detect anomalies or model and predict the impact of anomalies on other entities, which is in urgent need of the relational modeling capabilities brought by deep learning technology.

However, current graph deep learning methods still suffer from the following four tough problems. 1) Biased probability estimation in case of incomplete labels. The majority of graph deep learning methods follow a purely supervised or unsupervised learning paradigm, but graph deep learning methods are prone to cause models to converge to suboptimal solutions in the case of semi-supervised anomaly detection with such incomplete labels. 2) Insufficient spatio-temporal synergy. The vast majority of dynamic graph methods only consider the temporal and spatial relationship prediction



17010226

separately, which results in losing the synergy between them. 3) Insufficient multimodal adaptability for graph evolutionary prediction tasks. Past work used a single stochastic process model to capture the evolutionary patterns on the whole graph, and ignore the fact that different evolutionary patterns may exist between different regions on the graph. 4) Low real-time ability. The graph method not only needs to store entity attributes but also additionally stores the relationships of entities, which leads to the graph algorithm being more sensitive to the scale of data compared to the Euclidean algorithm. The above-mentioned four drawbacks hinder the further application of graph deep learning on real, complex, dynamic and large-scale graph systems, and this thesis provides certain effective solutions from four aspects respectively.

1) To address the problem of biased estimation due to data imbalance in graph anomaly detection scenarios, we propose a novel but simple semi-supervised graph neural network framework based on hypersphere learning, which distinguishes normal and anomalous data by graph network mapping with hyperspheres in the feature embedding space. It both compensates the deficiency of Euclidean anomaly detection algorithm in extracting graph relational features and overcomes the problem of biased estimation of supervised graph deep learning algorithms. In addition, the framework is compatible with almost all graph neural network computational models, and its usage can be generalized from static to dynamic graphs. And the framework brings performance improvements while being less computationally complex and easier to use than the state-of-the-art graph anomaly detection methods.

2) To address the problem of insufficient spatio-temporal cooperativity, we believe that the model needs to compress the feasible solution space of the model by introducing spatio-temporal constraints. And in order to introduce spatio-temporal constraints, the model must represent spatio-temporal rela-



17010226

tionships jointly in the same vector space, and the difficulty lies in the fact that the position information of nodes in the feature embedding space should reflect not only the spatial proximity but also the frequency of interaction relationships in time. Therefore, we propose a novel spatio-temporal modeling method containing two spatio-temporal relationship constraints to solve this difficulty. The first is the spatio-temporal triangular closure constraint, which requires a dynamic graph event "source-time-target node" triplet to satisfy the source node vector + time vector  $\approx$  target node vector in the feature embedding space. The second constraint is the monotonicity constraint of the time vector paradigm, which assumes that the further the time of an event is from the current moment, the larger the paradigm of the time vector corresponding to that event in the feature embedding space. The introduction of these two constraints can ensure that the position relations of nodes in the feature embedding space can contain both binary semantic information in time and space. In the inference stage, given the source node, the target node and its vector representation can be identified using the relative position relationship with other nodes, and then the time vector can be obtained using the triangular relationship, and finally the specific timestamp can be predicted by ridge regression on the time vector. The dynamic graph model in this chapter achieves the best performance in the dynamic relationship prediction task.

3) To address the problem of multimodal adaptation of the relational evolution prediction model to different evolutionary patterns on the graph, we adopt a "divide and conquer" strategy by splitting the relational evolution prediction task on the whole graph into separate evolution prediction for each community, and then using an additional stochastic model to model the interactions between communities. We consider the entities within a community as a whole for evolutionary prediction, and the different communities as different individuals with localization, ensuring the ability to model both



17010226

global and local information on the graph. To simplify the complexity of probabilistic modeling, we also propose a hierarchical stochastic time-series point process model based on graph neural networks. Node and target node respectively. The model not only can explicitly predict the future evolution of communities by using different stochastic process parameter models for different graph communities, but also achieves efficient stochastic parallel training in large-scale graph data by using a pure attentional structure in the encoder part instead of the recurrent neural network in the traditional stochastic process model.

4) To address the problem that graph neural network algorithms are difficult to deploy in real time systems, we have concluded that the graph model performance bottleneck is in the graph database query after a systematic analysis of the model time complexity. Therefore, we propose an asynchronous propagation attention network. It makes the graph query computation phase and the model inference phase decoupled and computed on asynchronous offline links and synchronous online links, respectively. The heavy graph query operations do not affect the speed of model inference, which further helps the model to obtain higher overall stability and scalability. The asynchronous computation mechanism can be applied to most of the graph deep learning methods, and is not limited to a particular algorithm. The framework can effectively implement ultra-large-scale graph inference in online distributed graph databases, thus advancing the application of graph deep learning in large-scale graph systems such as recommender systems, financial systems, and social networks.

**KEY WORDS:** Graph Neural Network, Deep Graph Learning, Temporal Graph, Anomaly Detection, Relational Reasoning, Community Evolution



17010226

## 目 录

<b>第一章 绪论 .....</b>	<b>1</b>
1.1 研究背景与意义.....	1
1.2 主要研究内容 .....	5
1.3 行文安排与整体架构 .....	8
1.4 关键技术与主要创新点 .....	12
1.5 本文使用的公开数据集 .....	14
<b>第二章 图学习理论基础与相关工作.....</b>	<b>16</b>
2.1 图论基础 .....	16
2.1.1 图的分类学 .....	16
2.1.2 图的基本概念 .....	20
2.1.3 图与矩阵 .....	23
2.2 图深度学习的数学形式 .....	25
2.2.1 基于谱域的图深度学习 .....	25
2.2.2 基于空间域的图深度学习.....	26
2.3 动态图学习任务定义 .....	26
2.3.1 动态图上的节点异常检测.....	27
2.3.2 动态图上的关系推理 .....	29
2.3.3 动态图上的关系演化预测.....	30
2.3.4 动态图任务之间的数学关系 .....	31
2.4 相关工作综述 .....	32
2.4.1 图表示学习 .....	32
2.4.2 图与异常检测 .....	34
2.4.3 动态图学习与链路预测 .....	37
2.4.4 动态图关系演化与随机点过程 .....	39
2.4.5 图深度学习算法的加速 .....	42
2.5 本章小结 .....	43
<b>第三章 基于超球面学习的图异常检测 .....</b>	<b>44</b>
3.1 引言 .....	44
3.2 基于超球面学习的图异常检测模型 .....	47
3.2.1 问题定义 .....	47



17010226

---

3.2.2	超球面学习 .....	48
3.2.3	训练目标 .....	49
3.2.4	OCGNN 的各种范式 .....	51
3.2.5	模型优化过程 .....	53
3.3	对比实验与分析 .....	54
3.3.1	数据集 .....	54
3.3.2	基线方法 .....	55
3.3.3	实验设置 .....	57
3.3.4	结果分析 .....	57
3.3.5	可视化分析 .....	60
3.3.6	参数敏感性 .....	60
3.3.7	运行效率 .....	63
3.4	本章小节 .....	64
<b>第四章</b>	<b>基于时空关系约束映射的动态关系推理 .....</b>	<b>65</b>
4.1	引言 .....	65
4.2	基于时空关系映射的动态关系推理模型 .....	68
4.2.1	多头注意力机制 .....	69
4.2.2	动态图注意力编码 .....	70
4.2.3	带有时空约束关系的动态图嵌入 .....	73
4.2.4	模型分层推理机制 .....	76
4.3	对比实验与结果分析 .....	77
4.3.1	基线方法 .....	79
4.3.2	评估指标 .....	81
4.3.3	实验细节 .....	81
4.3.4	结果分析 .....	82
4.3.5	参数敏感性分析 .....	83
4.3.6	并行计算实验 .....	85
4.3.7	异常用户行为序列可视化分析 .....	86
4.4	本章小结 .....	87
<b>第五章</b>	<b>基于层次化随机点过程的关系演化预测 .....</b>	<b>89</b>
5.1	引言 .....	89
5.2	层次化点过程图神经网络模型 .....	94
5.2.1	时间核方法 .....	95



17010226

---

5.2.2	基于动态图神经网络的特征编码器	97
5.2.3	基于层次化概率链的点过程事件预测模块	98
5.2.4	基于信息传播的自回归模块	100
5.2.5	损失函数	101
5.3	对比实验与分析	101
5.3.1	数据集	101
5.3.2	基线方法	102
5.3.3	实验细节	105
5.3.4	评估指标	105
5.3.5	结果分析	107
5.3.6	异常社群的可视化分析	108
5.3.7	消融实验	109
5.3.8	并行训练能力	112
5.4	本章小结	112
<b>第六章</b>	<b>基于异步信息传播的实时动态图推理策略</b>	<b>114</b>
6.1	引言	114
6.2	部署图深度学习模型的瓶颈分析	117
6.2.1	时间复杂度分析	117
6.3	基于异步信息传播的实时动态图模型	119
6.3.1	基于注意力机制的编码器	120
6.3.2	MLP 解码器	122
6.3.3	异步邮件传播模块	123
6.3.4	异步动态 GNN 框架	124
6.4	对比实验与分析	125
6.4.1	数据集	126
6.4.2	下游任务	127
6.4.3	基线模型	127
6.4.4	模型配置	128
6.4.5	结果分析	128
6.4.6	运行效率	130
6.4.7	参数敏感性	132
6.5	本章小结	135
<b>第七章</b>	<b>全文总结</b>	<b>137</b>



17010226

上海交通大学博士学位论文

---

7.1 主要研究结论 .....	137
7.2 研究展望 .....	139
参考文献 .....	141
致 谢 .....	158
学术论文和科研成果目录 .....	159



17010226

## 符号对照表

$\mathcal{G}$	图
$\mathcal{V}$	节点集（又名节点集）
$\mathcal{E}$	边集（又名链接集）
$ \cdot $	向量的模长
$\langle a, b \rangle$	向量的内积（点乘）
$ \mathcal{V} $	节点集中的节点数目
$ \mathcal{E} $	边集中边的数目
$\mathcal{V}_{t-}$	由 $t$ 时刻之前所出现的节点组成的节点集
$\mathcal{E}_{t-}$	由所有时间戳小于 $t$ 的边组成的边集
$\mathcal{G}_{t-}$	由所有时间戳小于 $t$ 的边所激发的子图
$v_i$	编号为 $i$ 的一个节点（节点）
$v^s$	某条边上的源节点
$v^d$	某条边上的目标节点
$e_{ij}$	节点 $i$ 和节点 $j$ 之间存在的一条边
$\mathcal{N}^{(k)}(v_i)$	节点 $v_i$ 的 $k$ 阶邻居集合
$\mathcal{N}^{(k)}(v_i; t)$	节点 $v_i$ 在 $t$ 时刻的 $k$ 阶时态邻居集合
$\mathcal{G}^{(k)}(v_i)$	节点 $v_i$ 的 $k$ 阶子图
$\mathcal{G}^{(k)}(v_i; t)$	节点 $v_i$ 在 $t$ 时刻的 $k$ 阶时态子图
$\deg(v_i)$	节点 $v_i$ 的度
$d(v_i, v_j)$	节点 $v_i$ 和 $v_j$ 之间的距离（最短路径）
<b>A</b>	图的邻接矩阵
<b>D</b>	图的度矩阵
$\mathbf{z}_{v_i; t}$	节点 $v_i$ 在 $t$ 时刻的特征嵌入



17010226

## 第一章 绪论

### 1.1 研究背景与意义

在过去几年中，GPU 的现代计算机不断增长的计算能力以及大型训练数据集（“大数据”）的可用性让深度学习方法<sup>[1]</sup>在不同的领域获得了巨大的成功。深度学习方法在语音识别<sup>[2]</sup>、机器翻译<sup>[3]</sup>和计算机视觉<sup>[4]</sup>的各种任务上取得了质的突破并远远超过了非深度学习方法。这点燃了深度学习的复兴并促进了其在学术界中的理论发展以及在工商业中的广泛应用。

深度学习之所以能取得如此大的成就，一个重要的原因在于，构建深度学习特征提取网络时，人们利用了在自然图像、视频和声音等欧氏数据中发现的先验知识，比如平移不变性（Stationarity）、局部联通性（Locality）和可合成性（Compositionality）<sup>[5]</sup>。例如在图像分析领域中大放异彩的卷积神经网络（Convolutional Neural Networks, CNNs）中，正是 CNN 的某些良好性质保证了其在处理数据时的平移不变性、局部联通性以及可合成性。首先，平移不变性指的是 CNN 具有对经过仿射变换后的图像仍然具有鲁棒性，该性质是由 CNN 在训练过程中的数据增强策略以及网络中的最大池化层保证的。其次，卷积核的局部联通计算性质则保证了 CNN 对于局部信息的提取能力。最后，可合成性则归因于卷积网络的多分辨率结构，该结构由不断累加的卷积层和池化层所组成。

尽管深度神经网络被证明是解决计算机视觉、自然语言处理和音频分析等欧氏数据的强大工具，但是近年来，越来越多的领域必须处理非欧氏几何空间上的数据。这是由于低维的非欧空间通常可以嵌入在高维的欧氏空间中<sup>[6]</sup>，使用非欧空间表达的数据往往可以具有更多更复杂的信息量。最典型的非欧数据就是图（Graph）数据，图又叫网络拓扑（Network Topology），由节点（Nodes）和边（Edges）组成。一条边的发起者叫源节点，接收者叫目标节点。而如果是动态图，则边上同时会附带时间戳，这时，动态图上的一条边（或称一个交互事件）就可以用“源节点  $v^s$ -目标节点  $v^d$ -时间戳  $t$ ”这个三元组来表示。在数学理论中，图是图论<sup>[7]</sup>的主要研究对象；在计算机科学领域中，图是一种被经常使用的基础数据结构；在数据科学领域中，图提供了一种数据的通用表达形式，来自各个领域的复杂关系型数据都可以直接表示为图，图的引入为复杂系统的建模提供了良好的工具。

在图1-1<sup>①</sup>中，我们展示了几个图数据的实际应用场景。a) 社交网络是一种以用户为节点、用户之间的关注关系作为边的图系统，在微博、微信等社交媒体中，

① 部分图片来自文献<sup>[8]</sup>



17010226

其规模如可达高达数十亿用户。人们可以从该图系统中获取历史交互信息并给用户推荐好友或商品；b) 在电子商务中，用户和商品之间组成购买关系图，用户和商品是节点、购买关系是边，基于图的学习系统可以利用用户、产品之间的交互来提出准确的商品建议；c) 在生物化学分析任务中，分子被建模为图，原子是节点、化学键是边，人们需要通过分子图结构鉴定其生物活性以进行药物、药理发现，从而指导新材料、新药物的研究任务；d) 如果将地铁交通网络中的各个站点作为节点，轨道作为边构成一张图，人们可以更直观地处理路径规划相关问题，或者去预测未来交通流量的变化情况；e) 智能工业设备所组成的工业物联网也可以用图数据来表示，图可以表示产业链或工厂内部上下游设备之间的关系，一旦链条某个关键节点出现异常，我们可以根据该节点与其他节点的关系预估其影响的大小；f) 在金融交易网络中，用户、银行、商户之间互相转账组成金融关系图，人们可以在图上进行交易关联分析以开展反欺诈、反盗用、反赌博、反洗钱等关乎金融安全的任务。

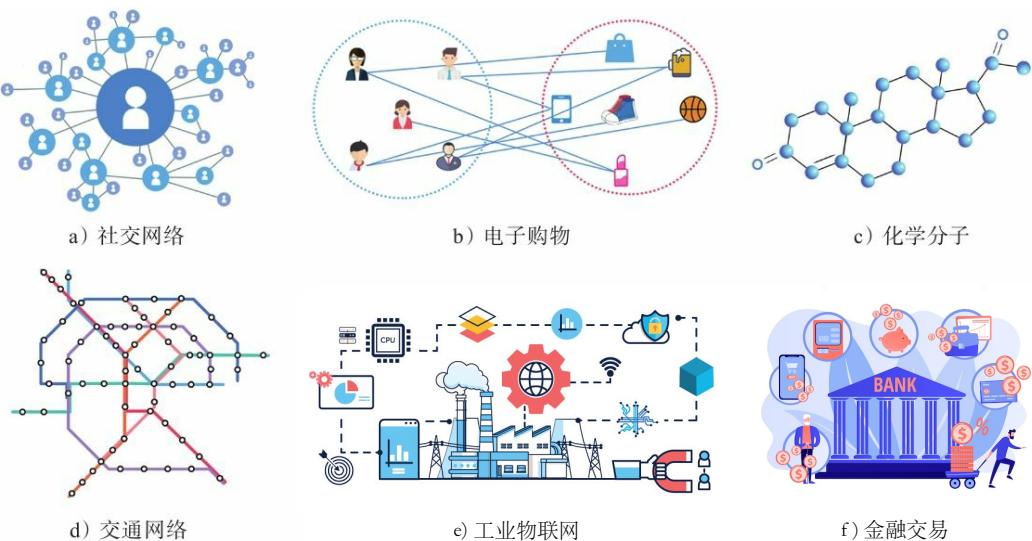


图 1-1 图数据应用示例

Figure 1-1 Graph data application example

传统深度学习算法所处理的数据一般要求数据集合中的任意一对样本点能在欧氏空间中计算距离，人们称这些数据为欧氏数据。如图1-2所示，对于图像数据来说，任意两个像素点可以被视为存在于欧氏空间的两个点，互相计算距离，这个判别方法同理也可以适用于文本、语音、时间序列等等数据中。欧氏数据就是在欧氏空间中规则排布的数据点，而非欧数据在数据空间的分布并不均匀，所以我们没有办法直接在非欧空间中计算数据点和数据点之间的距离。这导致传统的机



17010226

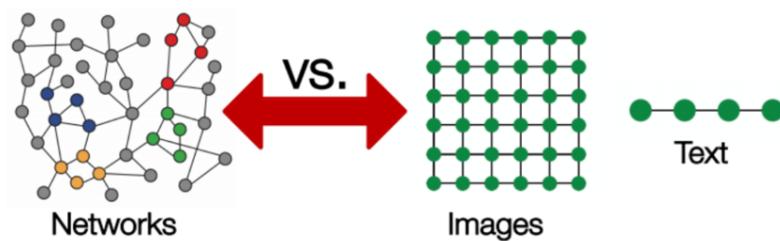


图 1-2 欧氏数据与非欧数据的核心区别

Figure 1-2 Key difference between Euclidean data and non-European data

器学习的计算方法无法直接迁移到数据，在非欧数据域中进行基于深度学习的特征表示学习面临着几个方面的严峻的挑战。首先，例如卷积等一些基本的深度学习算子没有明确的定义，这是因为非欧空间中的数据点之间没有上、下、左、右、前、后的关系，这导致非欧数据不具备在欧氏空间中自然存在的通用座标系、矢量空间结构或者平移不变性等性质<sup>[9]</sup>。其次，欧氏深度学习技术中存在一个核心假设，即模型处理的各个数据实体之间的关系是相同且规整的。而在非欧数据上每个数据实体（节点）与其他实体可能产生各不相同的互相影响。

为了克服这些困难，近年来，许多学者尝试在图这种非欧数据域中定义基本的学习算子，将深度学习方法扩展到了图计算领域，这类方法被称作图神经网络（Graph Neural Network, GNN）或者图深度学习（Deep Graph Learning, DGL）。图深度学习方法主要可以分为基于谱的方法与基于空间的方案<sup>[10]</sup>，基于谱的方法利用谱图理论来设计谱域中的图信号处理滤波器，该方法通常在图的拉普拉斯矩阵（Laplacian matrix）上定义图傅立叶变换及其滤波函数。而基于空间的方法则定义了图上的消息传递过程来迭代地执行图域的特征提取。大量的研究<sup>[11]</sup>表明，图深度学习已经大大推动了以多种图计算任务的发展，与此同时，图深度学习技术的发展也极大地促进了图表示学习在推荐系统、社会分析、交通预测、化学制药、组合优化、医疗卫生以及高能物理等新领域<sup>[12]</sup>中的广泛使用。图深度学习在各个学科中的广泛运用，给研究者们带来了许多新的见解，同时也使得图深度学习的研究成为一个真正的跨学科领域。除了上述的诸多应用场景，图上的动态异常分析也是图深度学习技术可以发挥其优势的潜在场景。本论文的研究致力于推动图深度学习在图上的动态异常分析领域的应用和发展，在该领域中，图深度学习技术同样有着传统欧氏学习模型无法比拟的优势，特别是在复杂性关系分析以及关系影响的传播等方面，下面进行进一步地阐述。

随着我国基础设施的发展，互联网、路网、卫星网、工业物联网等图系统正朝着复杂化、规模化的方向前进，而复杂系统发生故障的概率以及排查的难度都



17010226

成倍增加；随着我国金融市场进一步走向国际舞台，利用复杂金融工具及跨境交易进行洗钱、非法集资、诈骗等的犯罪行为也随之增多，严重危害国家金融和人民财产安全；移动互联网的不断普及给反社会、暴恐、黄赌毒等有害信息在社会中广泛传播提供了温床，而这将会严重扰乱社会舆情并危害青少年的身心健康。这些复杂故障、金融犯罪行为以及有害信息都属于大规模图系统中的小部分异常，这类异常通常难以发现却危害极大，因为它们非常容易快速地借由实体之间的关系传播影响并造成整个系统的紊乱<sup>[13]</sup>。而需要更高效地发现大规模图中的异常并预估异常造成的影响，就必须抓住上一句话所蕴含的图异常分析的三点困难。第一是系统内实体关系复杂，定位异常困难。不同于欧氏数据中的异常，非欧数据的异常可能是由于实体本身的属性导致的，也有可能是由于实体的连接关系导致的异常。如何建模复杂系统内部实体之间的多样性关系是困难的。第二是系统中的异常会向其他部分动态地传播影响。一旦检测出异常之后，从个体的角度，人们也想探明该异常个体会向哪些其他的个体传导；从整体的角度，人们对于预测该异常会对整个图系统造成怎样的影响也非常感兴趣。而图网络中的节点和连接关系在不断变动，过去基于欧氏建模的机器学习无法直接地学习实体之间的统计关系，导致很难对实体的动态演化所产生的连锁影响进行估计。第三则是异常的发生和演化可能非常迅速。异常扰动经常会造成巨大的连锁反应，一旦系统在某时刻发生异常，如何快速准确的检测异常并作出回应是一大挑战，如果不能快速检测出异常那么模型最终只能作为事后复盘的工具，不能在更大的事故损失发生之前，将异常扼杀在摇篮里。

而图深度学习方法为复杂图系统中的异常动态分析提供了良好的建模工具。首先，在由许许多多实体组成的大规模系统中引入图数据建模，不仅可以在考虑实体自身的性质的同时引入关系性建模，可以从实体属性和实体关系的双重角度检测异常，因此引入基于图的数据描述方法对于更准确地检测复杂系统中的异常是至关重要的。第二，图深度学习由于直接引入并建模了关联性，使其可以解决一些传统机器学习无法取得很好效果的图上的关系演化预测问题，这也是图深度学习这类非欧关系型模型相比欧式的深度学习模型的巨大优势。然而，目前的图深度学习方法在图异常动态分析的场景中仍然存在以下的四个主要的弊端。

第一，在图异常检测中，由于有标记的异常数据极端缺乏，正常、异常样本的数据不平衡，因此异常的对象可能并不存在或者仅仅只有少量标签。而目前绝大部分图深度学习方法<sup>[10]</sup>均采用纯粹的有监督或无监督的训练范式，很难在数据不平衡的情况下高效识别图上异常数据的固有模式。这通常会导致模型训练得到的是对整体数据的有偏估计。虽然已经有许多文献<sup>[14]</sup>在欧氏深度学习方法上提出了



17010226

改进策略以试图解决有偏性的问题，但将这些策略直接套用到非欧数据域上没有经验和理论上的保证，很少有文献探究图深度学习技术在图异常检测任务上的建模能力。其次，在图异常的动态分析中，存在着时空协同性不足以及多模态适应能力不足的问题。具体来说，过去的模型在考虑节点个体之间的相互影响时，没有将空间关系和时间关系两者结合考虑，造成了对时间信息的利用率低下。在考虑异常对整体图系统造成的影响时，没有考虑图上不同区域可能存在不同的演化模态，对于多种不同模态的适应能力不足。最后，图深度学习模型虽然在描述复杂系统的场景中存在优势，但图模型不仅仅需要存储实体的特征，还需要存储实体和实体之间的关系，需要额外的存储、查询、计算成本。导致其对数据规模的扩大更加敏感，在某些侧重时效性的场景下难以在实际的大规模系统中实时部署。

下一小节中，我们将详细地阐述这四个问题并引入本论文的研究内容。

## 1.2 主要研究内容

在上一小节中，我们简洁地介绍了本文的图深度学习领域的研究背景和意义，并点出了图深度学习这个方兴未艾的领域所存在的四个主要的问题。为了更清晰地描述以上存在的问题的重要性并解释我们改进方案的价值，下面我们进一步地从四个不同的角度对具体问题进行定义和分析，对前人工作进行总结和梳理，并说明用于解决这些问题的本论文研究工作的现实意义。

1) **类别不平衡时存在数据有偏估计问题：**在欧氏空间上定义的传统异常检测技术<sup>[15]</sup>虽然可以克服类别不平衡的问题，但其忽略了实体之间的关系，无法综合学习实体自身的特征与实体之间的长程相关性，因此很难从大规模的复杂图系统中定位潜在的异常成员。另一方面，一些非深度学习的图嵌入算法则存在非线性表达能力较弱<sup>[16]</sup>、运行效率不高<sup>[17]</sup>以及对超参数敏感<sup>[18]</sup>等问题。

基于以上现状，我们认为图深度学习是解决以上问题的关键工具，其原因如下：第一，图深度学习具有很好的捕获实体关联性的能力。第二，图深度学习的本质是融合了图信息的神经网络，其天然具有高度的非线性建模能力。其次，神经网络方法都具有优秀的并行训练和部署能力，非常适合于大规模的数据挖掘应用。最后，图深度学习能够自动提取图结构特征，并且超参数非常少，具有很低的参数敏感性。虽然图深度学习在图异常检测任务上具有非常优秀的潜质与良好的应用前景，但基于文献<sup>[19]</sup>的结论，图深度学习在类别不平衡情况下相较于欧氏机器学习模型更容易产生有偏估计问题。而其原因主要在于图上的点与点之间存在关联性，不完整的标签会迫使图模型加强一部分点之间的联系并弱化另一部分的联系，而这种偏见则与实际任务无关只与节点是否具有标签有关，这种有偏性



17010226

会损害模型的收敛能力并最终导致局部次优解<sup>[20]</sup>。这制约着图深度学习在图异常检测任务上的应用和发展，我们相信解决这个问题这可以进一步增强人们对复杂系统上异常实体的管控能力。

**2) 动态建模时存在时空协同性不足问题：**当前图深度学习模型在预测个体与其他个体的影响时，普遍存在时空协同性不足的问题。大部分模型<sup>[21-25]</sup>只仅仅考虑了预测实体与其他实体在空间关系上的影响，另一些方法<sup>[26-32]</sup>则只是割裂地分别预测空间和时间关系的影响，这带来了模型的时空协同性不足的问题。空间关系预测是指学习一个映射函数  $(\mathcal{G}_{t_0-}, v^s) \Rightarrow v^d$ ，根据历史的图信息给定的源节点  $v^s$  来预测下一个目标节点  $v^d$ 。如果模型的训练目标是与时间无关的纯空间关系预测，那么显而易见的是模型可能不会有强烈动力来编码历史数据  $\mathcal{G}_{t_0-}$  中有关时间的特征表示。而时间关系预测任务学习的是映射  $(\mathcal{G}_{t_0-}, v^s, v^d) \Rightarrow t$ ，即给定历史图信息以及两个节点，预测这两个节点下一次发生交互的时间  $t$ 。时间关系预测任务也存在着其固有弊端，它不具备在图的拓扑结构学习上的挑战性。时间戳预测任务在直觉上并不一定需要利用图上的结构信息。模型甚至可以仅仅通过统计两个节点之间的历史交互频率来预测他们下一次交互的时间。而一旦这种情况发生，模型自然就会丢失在整体图结构建模上的能力。

总得来说，前人工作中采用的空间或时间关系预测的模型训练目标的共性弊端在于，虽然它们在输入的数据中同时提供了有关空间关系和时间关系的信息，但是模型的训练目标却仅仅只需要单独推测空间关系或时间关系，而割裂的训练目标容易导致模型忽略有价值的信息，造成模型的性能下降。而时空协同建模的难点在于同时预测时间和空间关系会造成模型的解空间极具膨胀导致预测困难。所以目前动态图深度学习方案选择在两个嵌入空间分别建模时间和空间关系以回避这个难点，而这则制约了模型对于动态图上的动态特征的捕获能力。我们认为要进行时空协同地动态图建模，并避免解空间膨胀的问题，就必须引入约束，即在统一的特征嵌入空间针对图中节点的空间关系和时间关系进行通用的学习和表达。但统一的表达带来了额外的难点即在于节点在特征嵌入空间的位置不但要反映空间上的远近，还要同时反映时间上的交互关系频率。我们决心直面这项难点，令动态图模型不光可以预测异常节点对于其他节点的在空间关系角度上的影响，而且可以根据时间的不同给出灵活的预测，比单纯的基于空间的关系预测更加适应实际应用的需要。

**3) 关系演化预测时存在多模态适应能力不足的问题：**当前图深度学习模型在预测图系统这个整体的未来演化时，普遍存在多模态适应能力不足的问题。过去的一些动态图建模方法<sup>[33-36]</sup>提出了使用图上的随机点过程利用已存在的实体关系



17010226

来预测实体关系的演化过程。动态图演化建模任务用数学的方法表达就是估计演化的条件概率分布函数  $p(v^d, v^s, t | \mathcal{G}_{t_0-})$ ，即给定历史的动态图，预测该图上未来会发生哪些交互事件。然而，这些文献存在以下三点弊端。首先，一些先进的模型<sup>[26,35-36]</sup>试图仅仅从纯全局的视角，只使用单独的一个随机过程网络来描述整个图的演化，忽略了图上不同的区域可能具有不同模态的演化模式<sup>[37]</sup>，而这导致模型仅仅只能隐式地学习不同演化的模式区别，建模过于粗糙，带来了模型对于不同演化模式的适应能力不足问题。此外，另一些文献<sup>[33-34]</sup>同样基于随机时序点过程来进行演化预测，不同的是它们从纯局部的视角出发，在每两个节点之间建立一个独立的点过程概率模型，以表示两者在未来某时刻发生交互的概率。但这同时意味着模型需要  $\mathcal{V}^2$  复杂度（ $\mathcal{V}$  是图中的节点数量）的计算资源，当图的规模过大时算法存在运行效率瓶颈，通常不会在大规模的场景下被优先采用。

只使用一种全局的点过程模型概括全图的演化过程存在适应能力不足的问题，而针对每对节点建立局部的独立点过程模型又面临着高复杂度不可用的问题。我们认为解决问题的最有前景的手段就是，令模型根据不同的模态采用不同的随机过程参数模型来预测未来的演化，然而如何界定不同的模态，如何将不同的随机过程融合，并且保持全局和局部的信息是一个非常困难的要求。图上的实体关系演化过程预测对于人们理解图上的动力学演化过程具有非常大的意义，它是解决许多人类社会学难题的有潜力的方法<sup>[38-39]</sup>。

4) 图神经网络的高复杂度低实时性问题：图神经网络存在计算和存储复杂度高、难以实时性部署的问题，无法满足现实场景中对高频率、大规模的动态图数据的快速推理要求<sup>[23]</sup>。数亿规模和高频率变动的图数据场景（如金融网、电商网、社交网等等）越来越常见，比如金融交易系统中，如果不能准确地、实时地拦截某笔金融犯罪交易，犯罪份子可能会将赃款快速转移脱离监控，给金融平台和用户造成不可估量的经济和声誉损失。如何在这样的场景下部署图深度学习模型并令其实时推理结果是一个巨大的挑战。试想，就算图深度学习算法从理论上来说可以准确地捕捉到动态图系统中的异常的交易，但如果由于模型的推理速度不够，导致我们不能在更大的损失发生之前将异常扼杀在摇篮里，那么模型最终只能作为事后复盘的工具。

目前已经有许多研究成果致力于加速 GNN 的计算过程<sup>[40]</sup>，而这些优化思路主要有包括硬件加速<sup>[41-42]</sup>、软件加速<sup>[43-45]</sup>、和数据库加速<sup>[46]</sup>等三类，我们将在2.4.5小节对前人的工作的具体思路展开进一步地分析，这里只说明他们的工作普遍存在的问题。第一，硬件加速的方法存在难以动态调整的问题，其大部分只能适用某几种最原始的算法，针对层出不穷的新型图算法研发新的加速电路非常成本高



17010226

昂。第二，软件框架加速的方法的适用范围较窄。目前只封装了一部分常用的图稀疏计算算子，对于特殊的应用场景或者新的算子不能很好地支持。第三，图数据库加速方法始终有一个由计算机数据结构理论限制的优化上限，而通常来说该上限在某些场景下还远远达不到时效性要求。基于此，人们的确需要一种适用于大部分主流动态图深度学习算法的推理加速方案，这种方案的提出可以降低图神经网络算法的设计和部署难度，并进一步推进异步图神经网络在推荐系统、金融系统、社交网络等大规模复杂系统的应用。

### 1.3 行文安排与整体架构

本文总共有七章，各章的主要内容分别总结如下。

本章为绪论，首先我们在上一小节介绍了图深度学习的应用和研究背景。在本小节中，我们将介绍本论文的其余章节安排，并从四个方面层层递进地展开对本文研究方案的说明。之后，我们会汇总本论文的整体框架，梳理本论文所给出的四个主要工作之间的联系。

第二章为图学习基础理论。该章提供了较为严谨的数学形式，重点介绍图的理论知识，包括图的分类学、图的基本概念以及图中的矩阵理论，之后我们介绍了图深度学习（神经网络）的一般形式，最后我们将用数学语言正式地定义本论文所研究的三大图学习任务，并描述了这三大任务之间的关系。最后，我们循序渐进地介绍了图深度学习及其相关模型的理论知识和发展脉络，而这些理论知识将为后续篇章的论文细节理解奠定基础。

在后续的三、四、五、六这四章中，我们分别介绍了本论文的重点内容，也就是本论文针对上一小节的现存问题提出的创新改进方案。

第三章针对图异常检测场景下数据类别不平衡带来的**有偏估计**问题，提出了图深度学习模型的一种新颖的半监督学习范式，其基本动机是利用图模型在建模实体之间的关联性方面的优势来更好地描述图中的异常模式，我们称之为 **One Class Graph Neural Network (OCGNN)**。它试图将正常数据与异常的数据通过图神经网络的映射，并用特征映射空间的超球面学习正常数据的模式并以此区分异常模式。用这种半监督的训练目标指导图深度学习方法进行端到端的训练，成功地弥补了传统异常检测算法、无监督图嵌入算法以及图深度学习算法在这个场景下存在的不足。我们提出的模型不但可以提升图异常检测技术的效果，而且相对于普通的图深度学习方法降低了对于数据标签的要求，这促进了图深度学习方法在图异常检测任务上的大规模应用。

在第四章中，为了解决动态图深度学习存在的**时空协同性不足**的问题，我们



17010226

分别从训练目标和模型架构上进行了重要的创新。首先我们改进了动态图模型的训练目标，提出了相较传统静态关系推理更困难的时空间的联合建模任务——**动态关系推理**：给定人们所关心的节点，预测这些节点在未来的何时分别与图中的哪些其他节点产生交互关系。为了便于理解，我们举了几个例子来说明该训练目标的意义：以交通网络为例，假设某一个路口发生了车祸拥堵等异常拥堵事件，该拥堵会依次向哪些路口传播导致连锁性的拥堵<sup>[47]</sup>？而在工业物联网中，假如其中一个设备负载过高导致停转，这些负载缺口会不会依次迭代地传导到其他的设备，从而导致其他设备宕机进而造成更大的负载缺口，并酿成大范围生产事故？亦或者是电商平台的用户什么时候会购买新的商品、传染病什么时候会感染新的人等等。动态关系推理任务要求模型不光要预测实体与其他实体的交互，还要令模型准确预测该交互的时间戳，这迫使模型必须学会从空间和时间维度上进行联合的概率建模，解决了过去工作存在的时间-空间信息损失问题。

在模型架构的优化方面，我们则提出用于时间-空间联合建模的图神经网络算法，即动态关系推理网络（**TEmporal Relational Reasoning NEtwork, TERRINE**）模型，它在综合考虑节点特征和空间相关性的基础上，在空间相关性的表示空间中加入了两种时空关系约束：时空三角闭合约束以及时间向量的范数单调性约束。首先，我们认为一个交互事件“源节点-时间-目标节点”三个变量在特征向量空间中应该满足三角闭合关系，即源节点向量 + 时间向量 ≈ 目标节点向量。其次，我们认为如果事件的时间距离当前时刻越远，该事件的时间在特征嵌入空间对应的时间向量的范数也要越大。而由于第一个三角假设，时间向量越大，代表两个节点在嵌入空间的位置越远，同时也代表两个节点需要更长地时间才有可能发生交互。由此，节点在嵌入空间中的位置就同时包含了时间和空间的双元语义信息。在得到时间、空间关系在同一特征向量空间上的表示之后，给定源节点，我们利用节点之间的相对空间位置的远近预测目标节点并计算时间向量，最后利用对时间向量的岭回归预测出具体的交互的时间。该模型从特征学习的角度提出了时空间的统一特征描述来增加模型时间空间联合建模的能力。该模型建立了动态图上的时间、空间映射的依赖关系，从实验上证明了动态图上的空间和时间关系可以在通用的嵌入空间中进行描述，并取得更好的事件预测效果。

在第五章中，针对图上的演化预测模型应对不同区域上的不同演化模式存在的**多模态适应能力不足问题**，我们仍然分别从训练目标和模型架构上进行了非平凡的改进。首先我们认为不同的社群<sup>①</sup>天然具有不同的演化模式，而一个社群的

<sup>①</sup> “物以类聚，人以群分”，网络中节点会在某些情况下自然地分成一些节点组，组内节点间存在较多的边相连，不同组之间的节点连接相对稀少，满足这一特征的子图结构称之为社群（Community）<sup>[48]</sup>，它是用于观察和理解网络拓扑的一种重要特征。



17010226

内部演化模式则相对更好地被加以归纳。并且大规模图上的关系演化建模任务原本是难以求解的，我们使用“分群而治之”的思路，将困难问题分解为数个易解决问题的组合：针对图上不同的社群的演化模式采用不同的随机过程进行建模。这种方式可以显著地增强模型对于图的全局-局部信息的捕获。更何况研究某个社群内的未来事件演化本身就具有非常重要的意义。例如，基于社群的关系演化预测可以为社区内部的流行病（比如新冠病毒）传播作提前干预、采取精确的防控决策<sup>[49]</sup>；也可以帮助相关部门为某个经济状况<sup>[50]</sup>或政治倾向<sup>[51]</sup>发生突然变化的社区提供更多的关注和帮助；还可以帮助银行预测并优先打击会给社会带来更大危害的金融犯罪团伙。因此我们将整个图上的演化预测任务拆分为对各个社群分别进行演化预测，提出了**社群关系演化预测**：给定人们所关心的社群，其内部未来会发生怎样的模式的演化（连接关系的变化）？即该社群会逐渐发展壮大，还是慢慢销声匿迹，抑或是持续稳定地按照某种模式运行？

此外在模型架构的改进方面，我们提出了基于层次化随机点过程的图神经网络模型（Community Event Predicting task with a graph Point Process model, **CEP3**）用于预测社群内部的交互演化。受益于图神经网络带来的关联性建模能力，我们可以对一个图上的交互事件“源节点-时间-目标节点”进行层次化级联建模，通过建立的贝叶斯层次化条件概率链模型，先估计事件发生的时间的概率分布，再分别估计事件的源节点和目标节点的对应概率。同时我们也设计了基于动态信息传播的自回归框架，它可以根据前一事件的发生对社区造成的影响来决定当前事件的预测结果。该模型不光可以显式地根据不同的图社群采用不同的随机过程参数模型来预测社群未来的演化，而且该模型通过在编码器部分使用纯的注意力结构替代了传统随机过程模型中的循环神经网络，实现了大规模图数据中的高效率随机并行训练。值得一提的是我们的模型同时兼顾了可应用性与理论依据。模型不但保持了图深度学习对于复杂图数据的建模能力，而且还将随机过程理论与图深度学习结合，从随机过程理论的角度描述了社群关系演化的过程，这有利于人们对于社群理论的相关研究，推动相关技术的进步。

在第六章我们首先针对图深度学习算法处理的三个阶段——图查询、图计算和模型推理——分别进行了时间复杂度分析，得出了结论是图查询阶段的时间复杂度与图的点边规模的乘积成正比，即  $O(|V| \cdot |E|)$ ，而其他的两个阶段的时间复杂度都远远小于图查询阶段，因此图查询是阻碍图算法实时部署的瓶颈。为了克服这个瓶颈问题，我们重新调整了图神经网络模型的计算和图数据查询机制，并提出了一种高实时性优化的动态图表示学习方法——异步信息传播注意力网络（Asynchronous Propagation Attention Network, **APAN**）。具体来说，我们从根本上



17010226

重新设计了动态图算法的工作流程，使图神经网络模型的图查询和图计算阶段转移到模型推理之后，这保证了模型推理步骤是在线实时完成，而图查询和图计算步骤则是异步离线完成。我们还通过一种网络中的消息信箱机制来保证就算在异步查询计算步骤阻塞的情况下，在线推理步骤也可以保证较高的准确度。这种方法使推理步骤和图查询步骤解耦，繁重的图查询操作将不会影响模型推理的速度，成功地将复杂的图算法从在线业务决策系统中分离出来，从而获得更高的系统稳定性和可扩展性。我们引入图深度学习的异步计算机制可以改善动态图深度学习模型推理速度慢的问题。该异步计算机制的适用范围非常广泛，其可以适用于绝大多数的动态图深度学习模型，并替换其中的动态图嵌入模块，而不仅仅是针对某一种模型。

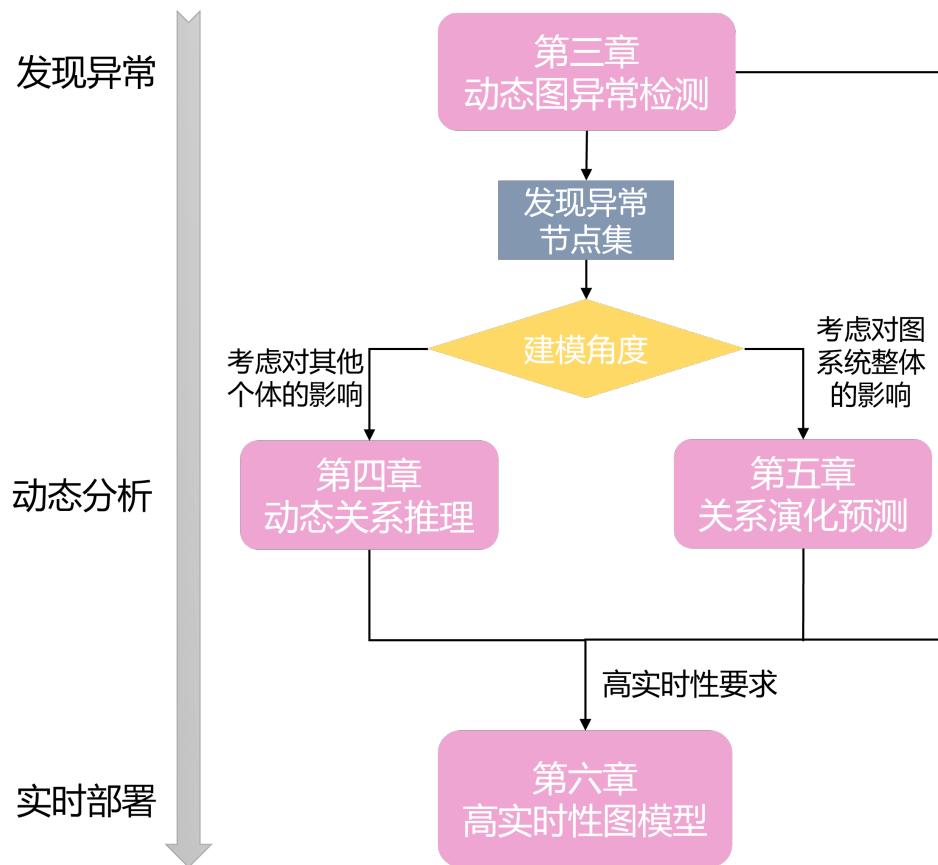


图 1-3 论文整体架构

Figure 1-3 The entire framework of this thesis.

本文通过四项重点不同的工作解决了图深度学习领域中的四点问题。值得一提的是，本文的四项工作其实具有层层递进和相辅相成的特性。本文的研究框架如图1-3所示，该图展示了上文所述的四个研究的在应用场景上的关系。如果需要



17010226

利用图上的实体关系定位异常的根因位置、从图上挖掘潜在的异常节点，则需要进行图异常检测；如果检出的节点都是零零散散的、不存在社群聚类性，人们会关心这些异常实体如何通过连接关系影响图上的其他正常部分，则可以进行针对单个异常节点分别进行动态关系推理以估计异常的传播；如果检出的节点呈现出明显的社群聚类特性，则应该使用关系演化预测的手段，从全图的整体角度来监控和预测异常社群的动态，比如会不断发展壮大、销声匿迹还是平稳发展；最后，不管是什么算法，实时性部署是将其应用于真实系统上中的最重要的关卡之一，而由于图深度学习模型不光需要存储实体的属性，还需要存储实体和实体之间的关系，因此对数据规模的扩大更加敏感。所以本论文的最后提出了一个高实时性的图深度学习模型来解决这个问题。

总结来说，虽然图深度学习技术的引入可以提升和扩大现有深度学习算法的效果和应用范围，但其仍然存在类别不平衡下的有偏估计、时空间建模不协同、多模态适应能力不足以及无法实时性部署等四点问题。本文利用了图深度学习技术在处理非欧关系型数据上的理论基础和性能优势，针对以上问题提出了至关重要的改进。本论文在图异常检测、动态关系推理、社群关系演化、图模型实时部署这四个方向拓展了图深度学习模型的应用范围，并对应提出了半监督的图异常检测、时空间的联合统一描述、图上的层次化随机过程以及异步图计算机制的四种提升方案进一步丰富了图学习的相关理论。本论文中的研究工作为图深度学习领域注入了新鲜血液，具备创新性和实际应用价值，不但可以为丰富现有的图学习模型理论，还可以为相关产业和经济发展提供了一套较为完整的学术和工业化解决方案。

## 1.4 关键技术与主要创新点

我们首先在 1.1 和 1.2 小节分别介绍了本课题的应用价值和研究内容，并在图1–3中描述了四章主要内容在应用背景角度上的联系。在图1–4中，我们从关键技术的角度重新审视一下本论文四章主要内容的整体贡献以及创新点。

本论文所提出的四项解决方案并不是通过添加额外的模块以增加已有方法的准确度，而是希望提出新思路来解决目前已有方案解决起来存在困难的问题。“如无必要，勿增实体”，贯穿本论文的通用方法论就是通过引入某些新的假说或新的理论，从而巧妙地将原本的复杂问题转化或分解为一些易解的子问题，从而达到降低原本模型的复杂度或是提高原本推理效果的目的。

在第三章中，由于异常数据非常稀少，模型很难学习到异常数据的描述。而我们将其转化为先学习正常数据的描述边界，再根据数据远离边界的程度来判定



17010226

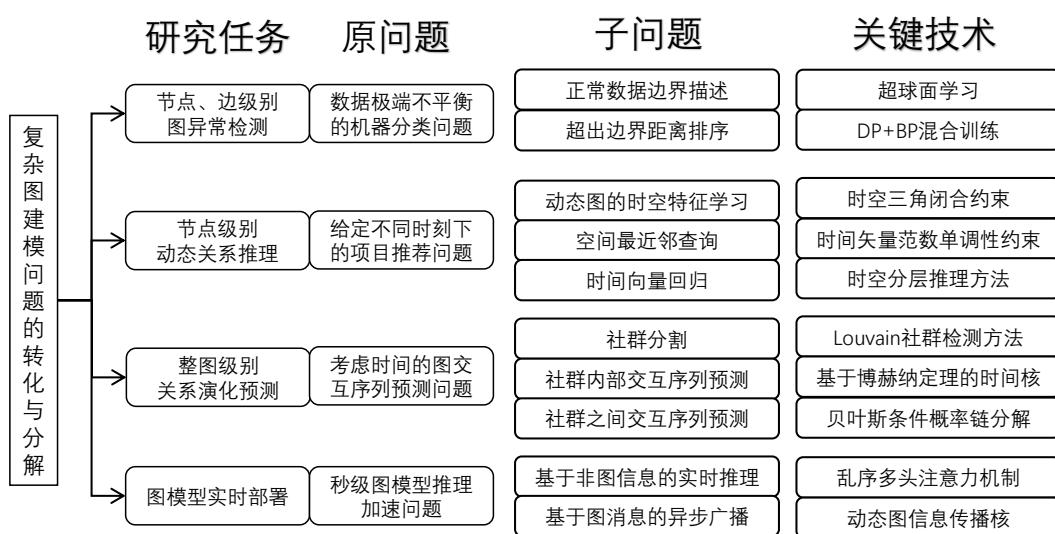


图 1-4 关键技术：复杂图建模问题的转化与分解

Figure 1-4 Key techniques: transformation and decomposition of complex graph modelling problems

异常。该章节中主要用到的两个核心技术为无监督的超球面学习技术以及基于动态规划（Dynamic Programming, DP）+ 反向传播（Back Propagation, BP）混合的图神经网络训练方法。前者用于学习图中的所有无异常节点并将其归纳为正常模式的机制，后者则是确保超球面可以根据不同模式自适应调整的条件。

在第四章中，由于图中的节点过多，求解原问题中待推荐的节点存在可行解过多的问题。我们为了避免这个解空间膨胀问题，我们首先将原推荐问题分解为节点特征的动态学习问题以及在节点特征表示空间上的空间、时间查询问题。首先为了更好地学习到节点特征，我们引入了两种时空关系假设以压缩解空间并保持时空关联性的语义信息，之后，我们提出了时空分层推理方法，使用简单的时间最近邻查询 + 时间岭回归预测来推理动态关系。

在第五章中，由于过去的关系演化建模方法存在求解规模过于庞大以及对全局-局部信息学习效率不高的问题，我们将难解的整图关系演化预测分解为基于社群的关系演化预测。我们将问题分解为三个步骤，它们分别是：社群检测、社群内部演化建模、社群之间的演化建模。在社群检测步骤中，我们采用了可拓展性最强的 Louvain<sup>[52]</sup>算法；而在剩余两个步骤中，我们以基于博赫纳定理 (Bochner's Theorem)<sup>[53]</sup>的时间核表示方法以及贝叶斯条件概率链分解方法为核心，提出了我们的全局-局部（即社群之间-社群内部）的演化建模模型。

而在第六章中，我们将原本复杂度高的图模型分解为低复杂度的非图计算部分以及高复杂度的图计算部分，并将两者分别部署在同步和异步链路上从而实现



17010226

在线推理部署。在异步链路中，我们提出了新型的动态图信息扩散框架，以将当前发生的图事件向其他邻居广播。而在同步链路中，我们提出了乱序的多头注意力信息聚合机制，以保证模型仅仅使用异步扩散的乱序信息仍然可以维持较高的学习能力。

## 1.5 本文使用的公开数据集

	Cora						
	Citeseer	Wikipedia	Reddit	Github	Mooc	SocialEvo	Alipay
	Pubmed						
数据来源	论文检索网站	维基百科	Reddit 论坛	开源代码网站	慕课网	MIT	支付宝
节点规模	数千	9 千	1 万	282	7 千	83	76 万
边规模	数万	14 万	60 万	2 万	40 万	6 万	277 万
特征维度	1024	172	172	10	4	10	101
节点描述	论文	用户和百科	用户和话题	用户和代码库	学生和课程	学生	用户
边描述	引用	用户编辑百科	用户发帖	用户编辑代码	学生上课	邮件消息	金钱交易
异常描述	稀少类别	恶意编辑	机器发贴	/	/	/	欺诈转账
异常比例	11%	0.11%	0.05%	/	/	/	0.04%
被使用	三	三四五六	三四五六	四五	四五	四五	六

表 1-1 本论文所用的公开数据集

Table 1-1 The public datasets we use in this thesis

表格1-1中展示了本论文所使用的数据集的统计概括和应用背景。我们一共使用了 9 种不同的数据集作为本论文的实际验证。这些数据的规模非常广泛，最小的数千，最大的达到了 300 万。数据集的背景也非常广，有学术社区、维基百科、论坛、支付宝等等。从表格中我们可以看出，本论文有两个发挥主轴作用的数据集，分别是 Wikipedia 以及 Reddit，它们是目前动态图建模以及动态异常检测最流行的公开数据集，使用它们的原因在于可以非常方便地与其他公开基线模型相对比。而从第三到第六章中，我们分别也引入了新的数据集以证明我们的模型在更广泛的领域也具有同样优秀的效果。

举例来说，在第三章，为了便于与静态的图神经网络方法相比较，我们引入了该领域最常用的引文数据集进行性能和效率上的严格对比。在第四第五章，由于 Wikipedia 以及 Reddit 都属于边特征非常丰富的数据集，为了证明我们的模型在属性特征较为稀少的数据集中也具有良好效果，引入了 Github、Mooc 和 SocialEvo 数据集以证明模型在属性特征及其稀少的情况下，也可以根据图结构特征进行学



17010226

习。在第六章中，为了展示模型在超大规模数据上的效果，我们引入了来自于规模为 300 万的支付宝的金融交易动态图。

此外，除了在第四第五章中使用的 Github、Mooc 以及 SocialEvo 数据集，其他的数据集都是跟异常检测任务相关。这些异常检测相关数据集的描述见表1-1的“异常描述”一栏，这些异常的显著特征为某些节点为了达成某种目的，密集地连接其他节点，形成局部的小型异常簇。而异常数据和正常数据的数量分布通常有较大的有偏性，该有偏性反映在异常数据占总数据的比值上。从表格中可以看出，异常数据占比通常都在 1% 以下，传统的有监督算法无法检测如此极端稀少的异常，更加无法对异常数据的远期演化进行建模，因此才有了本文的一系列异常检测以及异常动态分析等升级方法。



17010226

## 第二章 图学习理论基础与相关工作

在上一章中，我们使用通俗的语言介绍了本论文的课题背景、研究意义以及创新性等内容。而在本章中，我们使用数学语言对本文的研究对象以及任务进行了形式化定义。主要包括网络拓扑（图）的定义，以及图领域的基本概念；并针对图数据的不同场景介绍图的分类学；对于本文所聚焦的应用任务部分，我们层层递进地介绍了图上的异常检测，动态关系推理以及关系演化预测任务。此外，我们对于文章中的数学表示做了一般性的规定，除非特殊说明，我们使用英文或者希腊字母代表一个元素，用大写的英文或希腊字母代表一种函数映射关系，用花体大写字母代表一个集合，用粗体的小写字母代表一个向量，用粗体的大写字母代表一个矩阵。

### 2.1 图论基础

**定义 2.1 (图<sup>[54]</sup>)** 图 (Graph) 由顶点 (Vertex) 和边 (Edge) 组成<sup>①</sup>，节点代表我们需要研究的实体对象，而边则表示对象之间的关系。图可以被记为  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ ，其中节点集合  $\mathcal{V}$  是一个非空有限集合，边集合  $\mathcal{E}$  是一个有限多重集合。节点集  $\mathcal{V} = \{v_i | i = 1, \dots\}$  的节点数目为  $|\mathcal{V}|$ ，边集  $\mathcal{E} = \{e_{sd} | v^s, v^d \in \mathcal{V}\}$  中的边的个数为  $|\mathcal{E}|$ 。因为该边  $e_{sd}$  是由  $v^s$  出发连接  $v^d$  的，其中  $v^s$  和  $v^d$  分别被称为该条边上的源节点 (Source vertex) 和目标节点 (Destination vertex)。

举例说明，如图2-1a所示，此时该图的节点集  $\mathcal{V} = \{v_1, v_2, v_3, v_4, v_5, v_6\}$ ，边集为  $\mathcal{E} = \{e_{12}, e_{23}, e_{34}, e_{41}, e_{25}, e_{53}, e_{63}\}$ 。

#### 2.1.1 图的分类学

此外，从分类学的角度来说，图可以简单地从几种角度进行分类，它们分别是，无向图和有向图、无属性图和属性图、静态图和动态图以及同构图和异构图。

##### 2.1.1.1 无向图和有向图

**定义 2.2 (有向边和无向边<sup>[54]</sup>)** 对于图  $\mathcal{G}$  中的任意一条边：如果该边是从节点  $v^s$  出发到节点  $v^d$  的有方向的边，则称其为有向边，用序偶  $\langle v^s, v^d \rangle$  表示，其中第一

<sup>①</sup> 顶点通常也被叫做节点 (Node)，边也被称为链接 (Link)，在本文中，节点-节点以及边-链接这两对概念视为同一种指代，可以混用。



17010226

元素  $v^s$  称为有向边的源节点，第二元素  $v^d$  称为有向边的目标节点；如果该边是连接  $v_i$  和  $v_j$  的无方向的边，称为无向边，用无序偶  $\langle v_i, v_j \rangle$  表示。其中  $v_i$  和  $v_j$  均称为无向边的端点。

**例 2.1(有向图和无向图)** 如果图中的边存在方向性，那么称这种边为有向边 (Directed edge)。如果图中存在一条及以上数量的单向边，那么该图即为有向图，反之，如果图中所有的边都为无向边，那么称该图为无向图。使用数学语言描述为，如果对一个图  $G = (\mathcal{V}, \mathcal{E})$ ,  $\exists e_{ij} \notin \mathcal{E}$  and  $e_{ji} \in \mathcal{E}$ ，则该图为有向图。无向图是有向图的一种特殊情况，因为一条无向边可以用两条方向相反的单向边表示<sup>①</sup>。

通常来说，这两种图的构建方式不存在好与坏的区别，它们针对不同的任务通常具有不同的表现。例如在类似于微博这种中心化的社交媒体中，如果一个微博用户关注了一个明星，则一般认为该明星的行为会对其粉丝产生影响，而反之，该粉丝的行为几乎不会对明星产生影响，在这种场景中，适合用有向图来建模用户数据。而在类似于微信这种扁平化的社交媒体中，好友之间的相互关系和影响通常都是对等的，此时更加适用于无向图的建模方法。

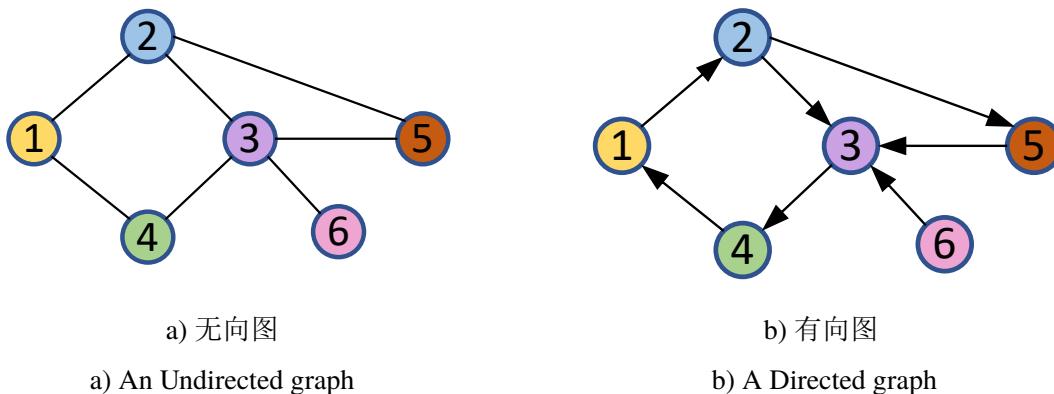


图 2-1 无向图和有向图。无向图认为图上的连接关系不存在方向性，一旦一条边建立，则认为该边上的端点互相存在关联；而有向图认为图上的连接关系仅存在单向的方向性。

Figure 2-1 The instruction of undirected and directed graphs. The edges on undirected graphs represent bi-directed relationship between two vertexes, whereas ones on directed graphs only contain single directed relationship.

### 2.1.1.2 无属性图和属性图

无属性图是由节点集和边集组成的，即  $G = (\mathcal{V}, \mathcal{E})$ 。而属性图中，节点或者边被赋予了特征向量用来描述该实体或者链接关系的特征，属性图  $G =$

<sup>①</sup> 在本文中，除非特殊说明，所有的图均指无向图。



17010226

$(\mathcal{V}, \mathcal{E}, \mathbf{F}^v, \mathbf{F}^e)$  不光存在节点集和边集，还存在着节点的特征矩阵  $\mathbf{F}^v \in \mathcal{R}^{|\mathcal{V}| \times d_v}$  以及边特征矩阵  $\mathbf{F}^e \in \mathcal{R}^{|\mathcal{E}| \times d_e}$ 。其中  $d_v$  和  $d_e$  分别为节点特征和边特征的特征维数。最简单的一种属性图就是加权图，加权图中的每条边都有一个实数权重与之对应，代表着两个节点之间的连接强度。在实际应用中，该权重可以代表两地之间的距离、运输成本或者原子之间的化学键强度。

### 2.1.1.3 动态图和静态图

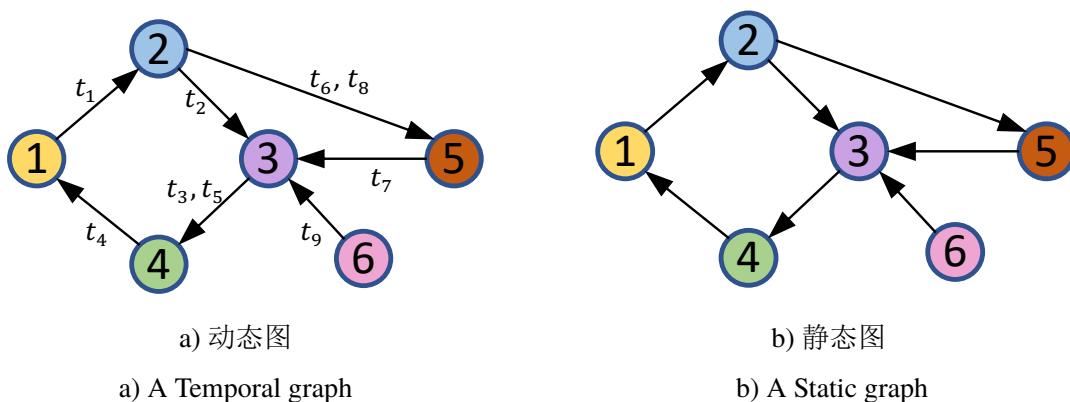


图 2–2 静态图和动态图。**(a)** 动态图中的每条边对应着一个唯一的时间戳，这意味着每条边的建立遵循着时间顺序，而且两个节点之间可以在不同的时间发生多次的交互。**(b)** 静态图忽略时间信息，认为每条边的出现没有先后之分，即静态图认为建模的是图系统在较长时间周期内的稳态性质，而不是动态性。

Figure 2–2 The instruction of static and temporal graphs. **(a)** The edges of temporal Graphs are labelled by timestamp. The interactions occur sequentially according to the order of timestamp; two nodes could have multiple interactions at different times. **(b)** Static Graphs ignore time information and assume that the appearance of each edge has no order of time, which indicates that static graphs capture the stable pattern in the long-term period, instead of the dynamic pattern.

**定义 2.3 (多重图和线图<sup>[54]</sup>)** 对于任意一个有向图，如果两节点间（包括节点自身间），若有同始点且同终点的多条边，则称这些边为平行边；两节点间平行边的条数称为边的重数；含有平行边的图称为多重图；非多重图称为线图。

虽然大多数的数据通常都建模成线图的形式，多重图也有很广泛的应用，其中最大的应用表现在对于动态图数据的建模能力上：如果两个节点之间在不同的时刻互相连接，我们采用如图2–2a的方式，给每条平行边赋予一个单独的时间戳，这样就可以规避普通的线图无法显式建模动态图的缺点。



17010226

如果图里的每条边都有一个时间戳 (Timestamp) 与之对应，我们称这样的图为动态图<sup>①</sup>，在很多实际的系统中，动态图的应用更加自然和广泛，例如，用户的社交圈总是随着时间不断地演化，用户对于商品的喜爱也会不断变化。如图2-2a所示，图中的每条边都有着明确的建立时间。而如果我们使用如图 2-2b 所示的方法，忽略时间信息，将蕴含时间信息的动态图简单地建模为静态图， $v_4 \rightarrow v_1 \rightarrow v_2$  这条路径会被视为合法的路径。但是这跟实际情况不符，由于  $t_4$  晚于  $t_1$  和  $t_2$ ，这个路径在静态图中合法，但在动态图中是非法的。有时，这种缺点在一些系统里是致命的，例如在商品推荐系统中，我们不可能事先探明用户后续会购买哪些产品；在电信设备网络中，我们同样也不清楚究竟何时会遭到攻击。静态图只能对于对象的历史信息进行归纳总结，但却很难利用时间信息，面对着越来越复杂的图系统，发展并完善动态图的相关算法和理论非常必要。

**定义 2.4 (动态图<sup>[54]</sup>)** 由此，我们可以给动态图下数学定义：动态图可以被表示为  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ ，其中  $\mathcal{E} = \{e^\tau : \tau = 1, \dots, T\}$  是动态的边集，而  $e^\tau = (v_s^\tau, v_d^\tau, t^\tau)$ ， $v_s^\tau \in V$  和  $v_d^\tau \in V$  是边上的两个端点，而  $t^\tau$  代表这个边发生时对应的时间戳。边的出现遵循着时间顺序，即  $t^{\tau_1} \leq t^{\tau_2}$  given  $1 \leq \tau_1 \leq \tau_2 \leq T$ 。我们另外定义一个标号  $\mathcal{G}_{t-} = (\mathcal{V}, \mathcal{E}_{t-})$ ，即  $\mathcal{E}_{t-} = \{e^\tau : \tau < t\}$ ，这个标号代表对于一个动态图只取其中时间戳在  $t$  时刻之前的部分。

在这里，我们额外拓展了2.1.2.1小节所定义的图上的邻居的概念，再参照我们定义的动态图概念，我们将动态图上的邻居概念拓展为时态邻居 (Temporal Neighbor)，时态邻居的概念在后面数章都会被使用到。

**定义 2.5 (时态邻居 (Temporal Neighbor))** 如果在时间戳  $t$  之前，存在一条边  $e_{ij}^\tau$  连接节点  $v_i$  和  $v_j$ ，并且  $\tau \leq t$  那么可以称  $v_j$  是节点  $v_i$  在  $t$  时刻的时态邻居节点，由这些时态邻居节点所组成的集合记为  $\mathcal{N}(v_i; t)$ 。 $\mathcal{N}(v_i; t) = \{v_j | \exists e_{ij}^\tau \in \mathcal{E}_{t-}\}$

此外，还有一些文献提出了离散动态图 (Discrete-Time Dynamic Graph, DTDG) 的概念，离散动态图通常将整个动态图按时间间隔转换为一系列静态图，即快照。离散动态图模型的模型表现对窗口大小的选择非常敏感，并且因为每个快照内部是静态图，所以时间变化的信息会丢失在快照中。在本文中，由于我们的工作重点不在于离散动态图，所以我们没有详细展开离散动态图的相关工作，重点在于阐述定义2.4所述的更加一般化的动态图。

<sup>①</sup> 动态图的主流定义主要有两种，分别是基于时间分片的离散时间动态图以及基于事件流的连续时间动态图，离散建模和连续建模几乎被认为是两种截然不同的体系，两者之间的区别详见综述<sup>[55]</sup>。本文主要研究的是连续时间动态图建模领域的相关方法，为了叙述方便，下文将其简称动态图。



17010226

### 2.1.1.4 其他类型的复杂图

除了上文所述的四种关于图的不同分类角度，还存在其他一些类型的图。之所以将其他类型的图收缩在一小节描述，并不是因为它们不重要，由于篇幅和主题所限，它们仅仅只是在本文中不做过多区分，相反，其他类型的图在各类应用中具有不可忽视的作用。

**定义 2.6 (异构图<sup>[54]</sup>)** 一个异构图  $\mathcal{G}$  由一组节点  $\mathcal{V} = \{v_1, \dots, v_N\}$  和一组边  $\mathcal{E} = \{e_1, \dots, e_M\}$  表示。其中每个节点和每条边都对应一种类型。用  $\mathcal{T}_V$  代表节点类型的集合，用  $\mathcal{T}_E$  表示边类型的集合。一个异构图有两个映射函数，分别是将每个节点映射到对应类型的  $\phi_V : \mathcal{V} \Rightarrow \mathcal{T}_V$  以及将每条边映射到对应类型的  $\phi_E : \mathcal{E} \Rightarrow \mathcal{T}_E$ 。

如果图中的节点类型和边类型都各自仅有一种，那么称这种图为同构图，同构图是实际的图系统的一种最简单的情况。而有时实际图系统中存在着各种各样不同的对象和关系，例如在购物网站中，存在用户、商家和商品等多种类型的对象，同时也存在浏览、点击、购买和评论等多种不同的关联行为。异构图就是用来描述多种不同类型对象的复杂关系的一种图数据结构，其图中的节点类型或者边类型大于等于一种。本文主要研究的是同构图建模的关键技术，因此不再对异构图进行细节性的描述。

**定义 2.7 (二部图<sup>[54]</sup>)** 给定一个图  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ ，当且仅当  $\mathcal{V} = \mathcal{V}_1 \cup \mathcal{V}_2$ ,  $\mathcal{V}_1 \cap \mathcal{V}_2 = \emptyset$  且  $\mathcal{V}_1, \mathcal{V}_2 \subset \mathcal{V}$  时，并且  $\forall e = (v^s, v^d) \in \mathcal{E}$ ，都有  $v^s \in \mathcal{V}_1, v^d \in \mathcal{V}_2$  时，则  $\mathcal{G}$  是二部图。

在二部图 (Bipartite Graph)  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  中，它的节点集  $\mathcal{V}$  可以分为两个互不相交的子集  $\mathcal{V}_1$  和  $\mathcal{V}_2$ ，而边集  $\mathcal{E}$  中的每条边都连接着分别来自  $\mathcal{V}_1$  和  $\mathcal{V}_2$  的两个节点。二部图广泛地被用于两个不同的对象之间的互动模式建模，比如在许多电子商务平台中，用户的点击历史可以被建模为一个二部图，其中，商品和用户是两个不相交的节点集，而用户的行为构成了二部图上的边。

## 2.1.2 图的基本概念

### 2.1.2.1 邻居和度

如果存在一条边  $e_{ij}$  连接节点  $v_i$  和  $v_j$ ，那么可以称  $v_j$  是节点  $v_i$  的邻居节点，我们记  $\mathbf{N}(v_i) = \{v_j | \exists e_{ij} \in \mathcal{E}\}$  为节点  $v_i$  的邻居集合。节点  $v_i$  的邻居个数也被称为该节点的度 (Degree)，即  $\deg(v_i) = |\mathbf{N}(v_i)|$ 。



17010226

### 2.1.2.2 路径和距离

在图  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  中, 若从节点  $v_i$  出发, 沿着一些图  $\mathcal{G}$  中的边经过一些其他节点  $\{v_{p1}, v_{p2}, \dots, v_{pm}\}$  到达节点  $v_j$ , 那么则称该边序列  $P_{ij} = \{e_{ip1}, e_{p1p2}, \dots, e_{pmj}\}$  为从节点  $v_i$  出发到节点  $v_j$  的一条路径 (Path) (通常来说, 两个节点之间不单单只有唯一一条路径)。由此, 我们可以定义两个节点之间的距离为连接两个节点的最短路径  $d(v_i, v_j) = \min(|P_{ij}|)$ 。注意, 节点  $v_i$  到其自身的距离  $d(v_i, v_i)$  为 0, 而两个无连接的孤立节点之间的距离为  $+\infty$ 。

有了路径和距离的概念, 再结合上文所述的邻居的概念, 由此我们引申出了  $k$  阶邻居的概念。 $k$  阶邻居 ( $k$ -hop neighbor): 若  $d(v_i, v_j) = k$ , 则我们称节点  $v_j$  是节点  $v_i$  的  $k$  阶邻居。

### 2.1.2.3 图的遍历

遍历是一种计算机术语, 是指沿着某条搜索路线, 依次对某种数据结构 (如树或图) 中每个节点均做一次访问。图的遍历是指从图上的某一个节点出发, 按照某种搜索算法沿着图中的边对图上的所有节点访问且仅访问一次。图的遍历主要有两种算法, 分别是: 广度优先搜索 (BFS, Breadth-First-Search) 算法和深度优先搜索算法 (Depth-First-Search) 算法。

广度优先搜索算法是一个层次化递进的搜索算法, 其算法思想是, 从图中某个节点  $v_i$  出发, 依次访问其邻居节点  $n_1, n_2, \dots$ ; 然后再按顺序访问这些邻居节点的还未被访问到的邻居 (即节点  $v_i$  的二阶邻居), 这种层次化的访问会一直执行下去, 直到图中所有的节点都被访问过一次。如果我们用图 2-2b 举例, 从 1 号节点出发的广度优先搜索序列为:  $v_1 \rightarrow v_2 \rightarrow v_3 \rightarrow v_5 \rightarrow v_4$ 。

深度优先搜索算法使用递归的思想解决图遍历问题, 其算法思想是: 从图中某一节点  $v_i$  出发, 访问它的任意一个邻居  $n_1$ ; 再从  $n_1$  出发, 访问  $n_1$  的所有邻居中, 没被访问过的节点  $n_2$ ; 然后再从节点  $n_2$  出发依次访问下去, 直到访问到某一个节点, 该节点的所有邻居节点都已经被访问过了。紧接着, 回退一层到前一次访问到的节点, 查询该节点是否存在没有被访问过的邻居, 如果有, 则访问该邻居并从该邻居出发进行与之前类似的访问, 如果没有, 就再回退一层。重复上述过程, 直到图上的所有节点都被访问过一次为止。同样地, 我们以图 2-2b 举例, 从 1 号节点出发的深度优先搜索序列为:  $v_1 \rightarrow v_2 \rightarrow v_3 \rightarrow v_4 \rightarrow v_5$ 。



17010226

## 2.1.2.4 子图

对于图  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ , 若存在由节点集  $\mathcal{V}' \subset \mathcal{V}$  和边集  $\mathcal{E}' \subset \mathcal{E}$  所组成的图  $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$ , 那么称图  $\mathcal{G}'$  是图  $\mathcal{G}$  的子图 (Subgraph)。通常来说, 当人们需要进行图计算来获取图中某一部分的性质时, 人们通常不会将整张大图输入算法进行处理, 而是只在全图中考虑某些感兴趣的子图。例如, 对于图在社交网络中的应用来说, 人们更加关注以某个节点为中心的子图, 查询某个用户的关联子图就可以找到该用户的好友关系, 从而分析出用户的潜在属性。借由上一小节所述的“广度优先搜索”的概念, 我们引入了节点的 K 阶子图, 它通常用来反映节点在图中的性质, 许多图深度学习算法利用节点 K 阶子图来计算。

K 阶子图 (k-hop subgraph): 对于节点  $v_i$  来说, 其 k 阶子图是由广度优先搜索产生的。根据上一小节所述, 广度优先搜索是一种分层搜索算法, 我们只需从节点  $v_i$  出发, 执行 k 层的广度优先搜索, 该搜索经过的节点和边就是该节点  $v_i$  的 k 阶子图。以图 2-3 为例, 在这个网络中, 2 号节点的一阶邻居为 3 号和 5 号节点, 2 号节点的二阶邻居为 4 号节点。则由 2 号节点所激发的一阶子图则为图 2-3a 中黄色区域所覆盖的部分, 注意, 从 5 号节点到 3 号节点的边不被包含在内; 而 2 号节点的二阶子图则为图 2-3b 中的黄色部分。

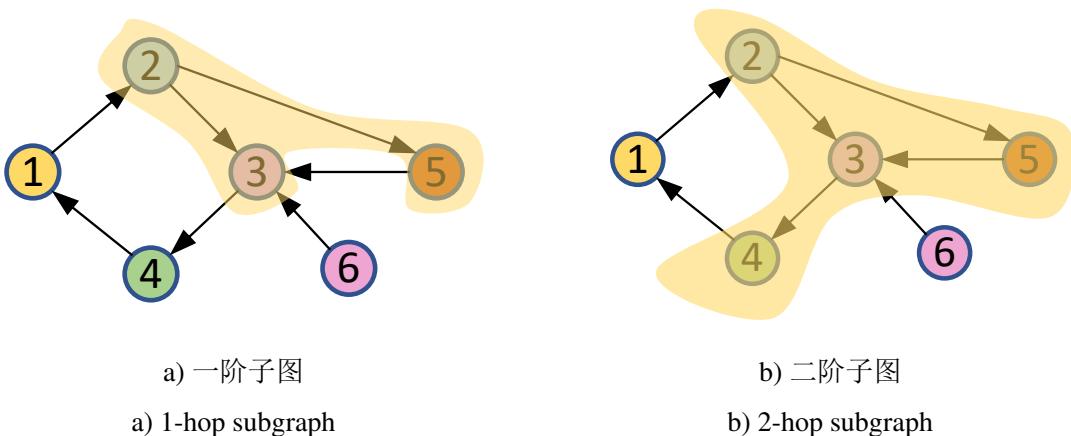


图 2-3 2 号节点的一阶和二阶子图

Figure 2-3 The 1-hop and 2-hop subgraph of Vertex No. 2



17010226

### 2.1.3 图与矩阵

#### 2.1.3.1 邻接矩阵

在计算机系统中，最常见图的存储表示方法就是邻接矩阵（Adjacency matrix）法。

**定义 2.8 (邻接矩阵<sup>[54]</sup>)** 对于图  $G = (\mathcal{V}, \mathcal{E})$ ，用一个  $n$  阶方阵  $\mathbf{A} = (a_{ij})_{n \times n}$  可以描述图中的各个节点之间的关联。 $a_{ij}$  表示图中从节点  $v_i$  出发到节点  $v_j$  之间是否有边相连，若有则  $a_{ij}$  为连接的边数  $k$ ，否则  $a_{ij} = 0$ 。其定义为：

$$a_{ij} = \begin{cases} k, & (v_i, v_j) \in E \text{ and } i \neq j \\ 0, & \text{otherwise} \end{cases} \quad (2-1)$$

从上述定义可以看出，如果  $G = (\mathcal{V}, \mathcal{E})$  是一个线图，则其邻接矩阵为布尔矩阵。此外，需要注意的是，要确定一个图的邻接矩阵，首先必须对图中的所有节点进行编号，邻接矩阵的表示与节点编号的次序紧密相关，不同的节点编号对应于不同的邻接矩阵，节点编号次序的交换对应于邻接矩阵行列的交换。

此外，对于无向图来说，图上的每条边不带方向，或理解为每条边同时具有两个方向，故其表示的对应节点的二元关系具有对称性。因此，图的邻接矩阵是一个对称矩阵（Symmetric matrix），即  $a_{ij} = a_{ji}$ 。对于有向图，其表示的二元关系不一定具有对称性，故其邻接矩阵不一定是对称矩阵。

在实际应用中，由于邻接矩阵的空间复杂度是  $\mathcal{R}^{|\mathcal{V}| \times |\mathcal{V}|}$ ，当  $N$  值（也就是节点数量）很大时，会消耗大量的存储空间。在实际应用中，由于邻接矩阵中存在大量的 0 值，我们可以使用稀疏矩阵的方式，只记录矩阵中非 0 值的位置，不存储 0 值。这样可以将邻接矩阵的空间复杂度降低到  $O(|\mathcal{E}|)$ ，节约大量的存储资源。

#### 2.1.3.2 度矩阵

除了邻接矩阵外，度矩阵也是图论中重要的矩阵之一。对于图  $G = (\mathcal{V}, \mathcal{E})$ ，其度矩阵  $\mathbf{D} \in \mathcal{R}^{|\mathcal{V}| \times |\mathcal{V}|}$  是一个对角矩阵（Diagonal matrix），即该矩阵主对角线之外的元素皆为 0。度矩阵的主对角线元素代表节点的度，即  $\mathbf{D}_{ii} = \deg(v_i)$ 。

度矩阵也可以通过邻接矩阵计算得到，度矩阵第  $i$  个对角线元素，即节点  $v_i$  的邻居个数为：

$$\deg(v_i) = \sum_{k=1}^n a_{ik} + a_{ii} \quad (2-2)$$



17010226

### 2.1.3.3 拉普拉斯矩阵

拉普拉斯矩阵 (Laplacian matrix) 也叫导纳矩阵, 其反映了在图中节点和节点之间的信息传播关系。拉普拉斯矩阵的计算方式非常简单,  $\mathbf{L} = \mathbf{D} - \mathbf{A}$ , 显而易见,  $\mathbf{L} \in \mathcal{R}^{|\mathcal{V}| \times |\mathcal{V}|}$ 。由此我们可以获知拉普拉斯矩阵的单个元素表示:

$$L_{ij} = \begin{cases} \deg(v_i), & i = j \\ -1, & (v_i, v_j) \in \mathcal{E} \text{ and } i \neq j \\ 0, & \text{otherwise} \end{cases} \quad (2-3)$$

拉普拉斯矩阵在图学习的领域非常重要, 然而由于拉普拉斯矩阵的对角线元素是节点的度, 不同节点存在不同范围的度, 有些节点的度可能是 1 或者 2, 而有些节点的度可能达到 100 以上。即对于不同的图来说, 其拉普拉斯矩阵的量纲是不同的, 为了解决这个问题, 研究者提出了归一化的拉普拉斯矩阵的概念。

归一化拉普拉斯矩阵 (Symmetric normalized Laplacian) 的定义如下:

$$\begin{aligned} \mathbf{L}^{sym} &= \mathbf{D}^{-\frac{1}{2}} \mathbf{L} \mathbf{D}^{-\frac{1}{2}} \\ &= \mathbf{I} - \mathbf{D}^{-\frac{1}{2}} \mathbf{A} \mathbf{D}^{-\frac{1}{2}} \end{aligned} \quad (2-4)$$

其中, 该矩阵的每个元素表示为:

$$\mathbf{L}_{ij}^{sym} = \begin{cases} 1 & \text{if } i = j \text{ and } d(v_i) \neq 0 \\ -\frac{1}{\sqrt{d(v_i)d(v_j)}} & \text{if } \{v_i, v_j\} \in E \text{ and } i \neq j \\ 0 & \text{otherwise.} \end{cases} \quad (2-5)$$

### 2.1.3.4 拉普拉斯矩阵的谱分解

由于拉普拉斯矩阵是实对称矩阵, 而实对称矩阵一定可以用正交矩阵进行正交相似对角化。因此该矩阵可以用公式2-6中描述的谱分解方法, 将图的拉普拉斯矩阵分解为由其特征值和特征向量表示的矩阵之积。

$$\begin{aligned} \mathbf{L} &= \mathbf{U} \Lambda \mathbf{U}^{-1} \\ \mathbf{L} &= \mathbf{U} \begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_{|\mathcal{V}|} \end{pmatrix} \mathbf{U}^{-1} \end{aligned} \quad (2-6)$$

此外, 又因为拉普拉斯矩阵对称矩阵的特征向量相互正交, 即  $\mathbf{U}$  为正交矩阵  $\mathbf{U}\mathbf{U}^T = \mathbf{I}$ 。所以拉普拉斯矩阵的谱分解也可以写成  $\mathbf{L} = \mathbf{U} \Lambda \mathbf{U}^T$ 。



17010226

## 2.2 图深度学习的数学形式

在本节中，我们从频域和空间域的两个角度介绍的图神经网络的基本数学形式。

### 2.2.1 基于谱域的图深度学习

既然图数据在空间域上无法很好地定义卷积操作，受到信号傅里叶变换中“时域卷积等于频域点积”理论的启发，人们提出了一些基于谱域的图深度学习模型。试图通过定义图上的傅里叶变换来执行原本十分困难的图卷积操作，在信号处理理论中，一个连续非周期信号可以经由公式2-7所述的傅里叶变换和傅里叶逆变换在时域和频域中进行转换。傅里叶变换将一个信号分解成为一系列不同频率的复指数形式  $e^{-i\omega t}$ ，称之为傅里叶变换的正交基函数。

$$\begin{aligned} F(\omega) &= \mathcal{F}[f(t)] = \int_{-\infty}^{\infty} f(t) e^{-i\omega t} dt \\ f(t) &= \mathcal{F}^{-1}[F(\omega)] = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) e^{i\omega t} d\omega \end{aligned} \quad (2-7)$$

而基于谱域的图深度学习算法最核心的思想认为图中的所有节点上的特征向量共同组成了一种图信号，那么类似地只要通过定义图傅里叶变换的基函数，就可以将图上的信号从空间域转换到频域中进行点积运算。而由于图拉普拉斯矩阵的特征向量  $\mathbf{U} = \{\mathbf{u}_1, \dots, \mathbf{u}_{|\mathcal{V}|}\}$  两两正交，天然适合作为图傅里叶变换的基函数，而如果特征向量作为变换基，则其对应的频率为拉普拉斯矩阵的特征值  $\{\lambda_i, i = 1, \dots, |\mathcal{V}|\}$ 。因此，图上的傅里叶变换被定义为：

$$\begin{pmatrix} \hat{f}(\lambda_1) \\ \hat{f}(\lambda_2) \\ \vdots \\ \hat{f}(\lambda_{|\mathcal{V}|}) \end{pmatrix} = \begin{pmatrix} u_1(1) & u_1(2) & \dots & u_1(|\mathcal{V}|) \\ u_2(1) & u_2(2) & \dots & u_2(|\mathcal{V}|) \\ \vdots & \vdots & \ddots & \vdots \\ u_{|\mathcal{V}|}(1) & u_{|\mathcal{V}|}(2) & \dots & u_{|\mathcal{V}|}(|\mathcal{V}|) \end{pmatrix} \begin{pmatrix} f(1) \\ f(2) \\ \vdots \\ f(|\mathcal{V}|) \end{pmatrix} \quad (2-8)$$

如果将图的傅里叶变换和逆变换写成矩阵形式，则可以被分别表达为： $\hat{f} = U^T f$  和  $f = U \hat{f}$ 。

在定义了图上的傅里叶变换之后，不同的基于谱域的 GNN 模型的最大区别就是其设计了不同的图信号的卷积滤波器  $\gamma(\lambda_{(\cdot)})$ ，以调制图信号的频率  $\lambda$ ，使得一些重要的频率分量被保留，次要的频率分量被缩减。一个设计良好的图卷积滤波器  $\gamma(\lambda_{(\cdot)})$  可以提取图信号信息量较大的部分，削减信号不重要的部分或者噪音部分。附带图卷积滤波器的图傅里叶变换的形式如下：



17010226

$$\begin{aligned}\hat{\mathbf{f}}' &= \gamma(\boldsymbol{\Lambda}) \cdot \mathbf{U}^\top \mathbf{f} \\ &= \begin{pmatrix} \gamma(\lambda_1) & & 0 \\ & \ddots & \\ 0 & & \gamma(\lambda_{|\mathcal{V}|}) \end{pmatrix} \cdot \mathbf{U}^\top \mathbf{f}\end{aligned}\quad (2-9)$$

### 2.2.2 基于空间域的图深度学习

基于空间的图深度学习方法通常都遵循着一种名为消息传递神经网络（Message Passing Neural Networks, MPNN）的通用框架。消息传播网络的核心在于其放弃了严格的图卷积的数学形式，转而定义节点之间通用的简单聚合函数来捕获图上的局部结构信息。基于这种节点聚合算子，每个节点可以表示为自身的信息与周围节点的信息的叠加，以达到和理论上的图卷积网络相比可媲美的效果。消息传播网络的计算方式主要分为两个步骤，首先将消息聚合函数作用在每个节点上，得到节点的局部结构表达；然后，将节点更新函数作用在自身的局部结构表示上，得到当前节点的新表达。MPNN 框架所定义的计算形式非常简单却又有效，对于图上的节点  $v_i$ ，MPNN 框架按照如下的数学形式更新节点特征：

$$\begin{aligned}\mathbf{m}_i &= \sum_{v_j \in \mathcal{N}(v_i)} M(\mathbf{f}_i^v, \mathbf{f}_j^v, \mathbf{f}_{ij}^e) \\ \mathbf{f}_i^{v'} &= U(\mathbf{f}_i^v, \mathbf{m}_i)\end{aligned}\quad (2-10)$$

其中， $M(\cdot)$  表示消息聚合函数， $U(\cdot)$  表示节点更新函数， $\mathbf{f}_{(\cdot)}^v$  和  $\mathbf{f}_{(\cdot,\cdot)}^e$  分别代表节点和边上的特征。通过将图神经网络的每一层设计成上述的消息聚合函数和节点更新函数，每个节点可以不断从邻近节点获取源信息从而更新自身的特征。在基于空间的图深度学习框架下，一些方法不再依赖于拉普拉斯矩阵的分解，而是设计某种神经网络来学习定义聚合函数。这些方法学到的聚合函数可以自适应于不同的任务和图结构，相比基于谱域的图深度学习方法具有更强的灵活性。

### 2.3 动态图学习任务定义

根据2.1.1.3小节所定义的动态图的形式，动态图跟静态图的区别在于其边上存在一个对应的时间戳。由于时间信息的引入，研究人员针对这种图上的时间动态特性定义了一些细致化的图学习任务。比如传统的静态图上有节点分类任务，旨在给某个节点分配类别，例如，判定某个用户是否是机器人自动化操作，如果是就限制机器人的权限。然而，在动态图的背景下，判断用户是否是机器人变成了



17010226

一个具有时间背景的任务，即判断用户在某个时刻是否会变成机器人（一些正常的用户很可能将自己的账户借给他人进行刷单），从而增强机器人账户限制的准确性。

再例，静态图上的关系推理任务本来指的是推断图中两个节点的关系，例如，在制药领域中，我们无法知道所有的药物和蛋白质之间的关系，因为做这种大规模的人体药物实验的成本非常之高。关系推理任务的目的在于只知道一部分的药物-蛋白质相互作用的情况下，对我们缺失的相互作用做出很好的猜测。此外，在电商或者社交软件的推荐系统中<sup>[56]</sup>，关系推理任务技术可以为用户推荐商品也可以为推荐新的好友。静态关系推理任务也可以跟动态图的时间信息产生奇妙的化学反应：在静态图中的关系推理任务只能推断用户是否会购买某个商品，而一旦利用图上的时间信息，就可以完成更进一步的时间推断，即用户何时会购买某个商品。通过这种方式，商家可以根据库存、物流以及流行趋势动态地推荐优质商品。

鉴于本文的主要研究对象就是动态图，以及动态图的异常检测，我们定义了三种与异常检测息息相关的崭新任务：动态图异常检测、动态关系推理以及关系演化预测，这三个任务即构成了本文的主要研究脉络。下面我们分节对这三种任务进行描述。

### 2.3.1 动态图上的节点异常检测

我们首先在这里介绍传统的异常检测任务以及静态图上的异常检测任务。数据泛滥的时代已经引起了人们对异常检测技术的兴趣<sup>[15]</sup>。一般来说，异常检测任务可以被视为一个半监督的二分类机器学习问题，该任务假设训练集中所有的样本都是无异常的样本，或者只存在极为少量的异常样本，要求模型在没有（或有极少）异常数据的情况下，学会区分异常和正常数据的数据分布差异。这种定义是紧密贴近实际且合理的，因为在大部分系统中，鉴于获取真实的异常数据的成本高昂，有标记的异常数据极端缺乏，异常的样本数量远远少于正常的样本。比如在正常运作的工业生产系统中，存在故障的设备总是极少数；在数亿的社交网站用户中，机器人账户也非常的稀少。

给定训练数据集  $\{x_i, i = 1, \dots, k, \forall x_i \in \mathcal{R}^d\}$ ，其中  $d$  是样本的数据维度， $k$  是训练集的样本个数。异常检测模型致力于归纳训练集中正常数据的概率分布。在测试过程中，针对未见数据  $x_u$ ，异常检测模型根据  $x_u$  偏离正常分布的大小而产生一个异常得分  $S(x_u)$ 。 $x_u$  的异常得分  $S(x_u)$  越高，则越有可能是异常数据，反之越低。如果模型认为数据样本  $x_u$  是异常数据，则给其分配标签  $y_u = 1$ ，反之如果被



17010226

认为是正常数据，则分配标签  $y_u = 0$ 。

基于上述异常检测任务的定义，我们可以将其推广到图数据领域。给定图  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ 。通常来说，图中仅仅只有一部分节点的标签是可见的，并且这部分标签可见的节点都属于无异常的节点，即  $y_i = 0, \forall v_i \in \mathcal{V}_{train}, \mathcal{V}_{train} \subset \mathcal{V}$ 。节点层面的学习任务旨在给每个无标签的节点  $v_i$  分配一个标签  $y_i = 0 \text{ or } 1$ ，0 代表正常，1 代表存在异常。节点的异常检测通常都跟复杂系统的安全性挂钩，比如在金融交易网络中识别异常的欺诈、洗钱、赌博用户，在电信网络中识别黑客或者宕机服务器节点等等。

动态性质在异常检测中尤其重要，利用时间信息可以给图上的异常检测模型带来两个不可忽视的好处。首先，因为大部分的异常情况都是由正常情况所转化而来的，例如工业设备的故障演化总是从正常运转状态到疲劳易损状态再到故障出现；金融欺诈团伙在刚刚成立的时候总是会隐秘地把自己伪装成普通用户进行日常交易，等待时机成熟再行犯罪事实，进行密集的洗钱、盗用和欺诈行为。

再者说，时间信息本身就会对是否是异常的判断产生影响，如图2-4所示，在静态图的视角中，从用户 1 到用户 6 看上去好像形成了一个洗钱的闭环回路，从用户 1 账户上转出的黑灰产资金经过中介的层层划转最终洗白又回到用户 1 手中。然而事实有可能与静态图视角的判断相违背，在动态图的视角中，这个从用户 1 出发再返回用户 1 的回路有可能是偶然形成，因为该回路的时间不是连续的（假设  $t_1 < t_2 < \dots < t_7$ ）。用户 2 在  $t_7$  时刻将钱转给了用户 3，而用户 3 早在  $t_3$  时刻就已经完成了转账，通常来说这种转账应该是不具备洗钱的嫌疑的。再举一个类似的例子，如果几个用户之间在白天发生了密集的相互转账，那么这很有可能是正常的相互往来；而如果凌晨时突然发生密集的转账，则很有可能有较大的犯罪团伙嫌疑。

由此我们得出结论，学会捕获并利用动态图上的时间信息来发现异常模式和正常模式的区别是非常重要的，此外，并且准确预测出某个节点何时进入了异常情况是非常有价值的。下面我们给动态图的异常检测任务下数学化的定义：

**定义 2.9 (动态图异常检测<sup>[57]</sup>)** 给定动态图  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ ，边集  $\mathcal{E} = \{e_\tau : \tau = 1, \dots, T\}$  是有序集合，其中  $e_\tau = (v_\tau^s, v_\tau^d, t_\tau)$ ， $v_\tau^s \in V$  和  $v_\tau^d \in V$  是边上的两个端点，而  $t_\tau$  代表这个边发生时对应的时间戳。边的出现遵循着时间顺序，即  $t_{\tau 1} \leq t_{\tau 2}$  given  $1 \leq \tau 1 \leq \tau 2 \leq T$ 。动态图异常检测模型的目的是给定待查询的节点  $v$  以及查询时间戳  $t_0$ ，根据节点  $v$  过去的历史交互信息，推断在  $t_0$  时刻，节点  $v$  是否存在异常行为。

这类任务比静态的图异常检测任务困难数倍，因为静态的图异常检测仅仅是



17010226

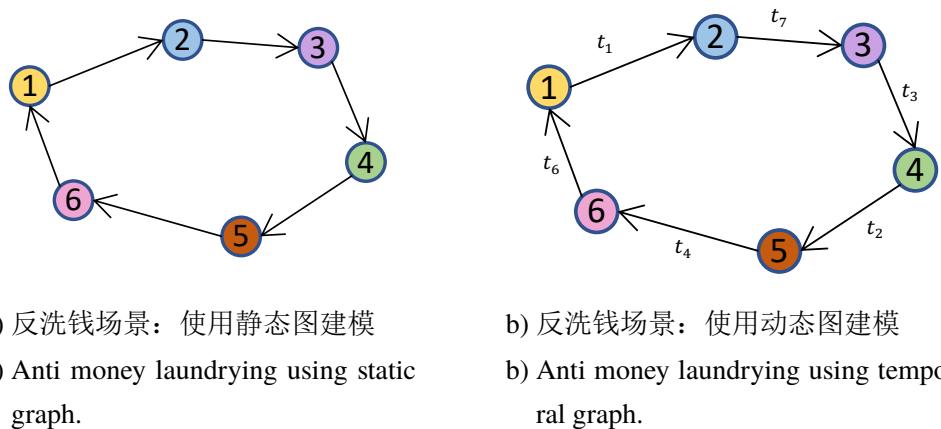


图 2-4 动态图建模在反洗钱场景的优势。在静态图中，从用户 1 到用户 6 看上去好像形成了一个洗钱的闭环回路；然而在动态图的视角中，由于该回路存在时间的非连续性，则不构成一条合法的回路。

Figure 2-4 The advantages of temporal graph in anti money laundrying. In the view of static graph, the user ranged from No. 1 to No.6 constructed a cycle loop, whereas in temporal graph, the cycle loop is illegal because the timestamps in the loop is not continuous.

判别某个节点是否存在异常，而动态的异常检测则是要根据时间不同来判断该时刻的状态是否是异常，这蕴含了时间变化的适应性，也就是说模型不仅仅学习了时间不变性的特征，而且往往学会了时间变化性的特征。

### 2.3.2 动态图上的关系推理

在很多应用场景中，人们需要知道图上的两个节点之间潜在的关系。比如在制药领域中，我们无法知道所有的药物和蛋白质之间的关系，因为做这种大规模的人体药物实验的成本非常之高。如果我们只知道一些特定的药物-蛋白质相互作用，但我们想对缺失的相互作用做出很好的猜测，我们可以使用机器学习来推断图中节点之间的边吗<sup>[58]</sup>？此外，关系推理技术的使用场景还有很多，在电商或者社交软件的推荐系统中<sup>[56]</sup>，如果推荐系统预测一条由用户节点指向商品节点的边，软件给该用户推荐商品；如果推荐系统预测一条由用户指向用户的边，软件就会给该用户推荐新的好友。

**定义 2.10 (关系推理)** 关系推理，又称链路预测、图补全。给定一个不完整的图  $\mathcal{G} = (\mathcal{V}, \mathcal{E}_{train})$ ，节点和节点之间不光由现存的边集  $\mathcal{E}_{train}$  所连接，而是存在着其他潜在的连接关系，关系推理旨在根据现存的不完整边集  $\mathcal{E}_{train}$ ，补全其他潜在的连接关系组成新的边集  $\mathcal{E}_{new} = \mathcal{E} \setminus \mathcal{E}_{train} = \{e_{sd} | v^s, v^d \in \mathcal{V}, e_{sd} \notin \mathcal{E}\}$ 。



17010226

普通的关系推理任务存在一个弊端，即假设节点之间的关系是静态的，在关系推理时并没有考虑关系的动态变化特性。而当应用场景切换到动态图时，关系推理的定义急需拓展以满足动态图建模的需求。因此我们定义了动态图上的关系推理任务，该任务不光要预测一个异常节点会如何影响其他的节点（即建立一条新的连边），同时也要尽可能准确预测该连边所产生时的时间戳。能否预测关联所发生的时间是区分普通的静态关系推理和动态关系推理的最大区别，另外一个重要区别就是静态关系推理相比动态关系推理丧失了利用时间信息建模的能力。

**定义 2.11 (动态关系推理)** 给定动态图  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ ，给定待查询的节点  $v^s \in \mathcal{V}$  以及查询时间戳  $t_0$ ，根据节点  $v^s$  过去的历史交互信息，预测一个二元组  $(v^d, t)$ ，其中  $v^d \in \mathcal{V}, v^d \neq v^s$  并且  $t > t_0$ 。意思是推断源节点  $v^s$  在查询时间戳  $t_0$  之后，最有可能首先跟哪个节点  $v^d$  相连，以及这条从节点  $v^s$  出发到节点  $v^d$  的连边的时间戳  $t$  是什么。

### 2.3.3 动态图上的关系演化预测

进一步地，给定历史的图信息，我们很可能会好奇该图在未来的演变过程，这对实际的产业和社会场景有着可见的裨益。例如，在以监控新冠病毒传播为目的的人群社交网络中，一旦发现了一个存在疑似病例的社交圈，政府必须要想办法预测该社交圈中疾病传播的进一步发展，以便为下一步决策做准备。在异常检测的应用中，有时我们会通过某些图异常检测算法或者异常社群发现算法发现一些可疑的异常社群，关系演化预测也可以被认为是发现异常社群之后的下一阶段任务，比如监管部门发现了一个疑似的诈骗团伙，如果可以使用关系演化预测的工具来推演该诈骗团伙的未来一个阶段的行为，就可以有的放矢地进行拦截并对可能被骗的用户发出警告。再比如说，关系演化预测技术可以为流行病（比如新冠病毒）传播作提前干预<sup>[49]</sup>；也可以对某个交通拥堵区域的拥堵传播情况作监控和预测<sup>[47]</sup>，方便交警提前布置人手干预；如果模型预测某个社区的经济状况<sup>[50]</sup>或政治倾向<sup>[51]</sup>发生突然变化，则代表社区需要政府和相关部门更多的关注和帮助。

关系演化任务的目的是在给定历史数据（即图  $\mathcal{G}$ ）的情况下，模型预测在下一阶段，该图上的节点之间接下来可能会发生怎样的交互，并且预测这些交互的时间戳。由此，模型就可以把图上的演化模拟出来，预估其未来的发展，为传染病防治以及诈骗团伙监控等应用提供帮助。下面我们使用更加数学化的语言给关系演化预测任务下一个定义：

**定义 2.12 (关系演化预测<sup>[35]</sup>)** 给定动态图  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ ，边集  $\mathcal{E} = \{e_\tau : \tau = 1, \dots, T\}$  是有序集合，其中  $e_\tau = (v_\tau^s, v_\tau^d, t_\tau)$ 。关系演化预测的目的是，根据时间戳在  $t$  时刻之



17010226

前的历史动态图  $\mathcal{G}_{t-} = (\mathcal{V}_{t-}, \mathcal{E}_{t-})$ , 直接推理  $k$  个的未来事件  $\{e_i, i = 1, \dots, k\}$ , 其中  $e_i \notin \mathcal{E}_{t-}$ , 并且  $e_i$  是一个三元组  $(v_i^s, v_i^d, t_i)$ , 其中  $v_i^s, v_i^d \in \mathcal{V}$  并且  $t_i > \max_{e_\tau \in \mathcal{E}} t_\tau$ , 这个三元组代表的是图内部的一次交互。

### 2.3.4 动态图任务之间的数学关系

更重要的是, 我们希望在图2-5中对本节中所提到的三大主要动态图学习任务做一个关系梳理。不同于我们在第一章的图1-3中对于三个任务在应用方式的区别叙述, 在此, 我们对于这三者之间的数学区别再一次进行强调。



图 2-5 三个动态图任务之间的数学关系

Figure 2-5 The mathematical relationship among three tasks.

三种动态图任务之间基本上呈现着由易到难的关系, 本文也是按照先介绍基础知识再逐渐加深难度的方法来行文的。首先是动态图异常检测任务, 它的输入是动态图本身, 以及待查询的节点, 输出是其节点的标签 (是否有异常), 可以认为这是一个二选一的任务。其次是动态关系推理任务, 它给模型的输入是动态图以及一个源节点, 要求模型从图中的所有其他节点中选择这相当于是一个  $|\mathcal{V}|$  选 1 的任务,  $|\mathcal{V}|$  为图中出现的节点个数, 就光考虑节点预测这个任务就已经比第一个任务困难的多, 更别提动态关系推理任务还要同时预测时间戳。然后是关系演化预测任务, 关系演化预测任务要求模型先从所有的  $|\mathcal{V}|$  节点中挑一个作为源节点, 再从剩下的  $|\mathcal{V}| - 1$  个节点中挑一个作为目标节点, 并还要预测时间戳, 显而易见, 任务的困难度又升了一个档次。最后, 不管在数学上定义多么优美的模型始终要想办法部署在实际系统中, 我们第六章的模型就是解决这些复杂图模型的实时部署问题。



17010226

## 2.4 相关工作综述

### 2.4.1 图表示学习

具有良好表达能力的特征是进行分类、回归等机器学习下游任务的前提条件，为了更好的利用现代机器学习模型进行图运算，有效地学习出图的特征表示就显得尤为重要。因此，近十年来，图表示学习任务得到了前所未有的发展，成为了图学习领域中的一个重要研究方向。图表示学习算法的基本目标是为图上的每一个节点生成其特征表达，该表达必须要充分考虑节点本身的特征以及节点周围网络的结构特性。该领域的发展主要可以分为三个时代：流形学习时代、图嵌入时代和图深度学习时代。下面我们沿着该领域的发展脉络，分别针对这三个时代的相关方法进行简短的文献综述。

#### 2.4.1.1 流形学习

流形学习技术是第一代的图表示学习技术，其原本的目的不是学习已有的图特征，而是针对高维度数据点进行降维。流形学习技术以经典的基于图的降维技术为基础，该技术主要包括 Isomap<sup>[59]</sup>、LLE<sup>[60]</sup>以及 Eigenmap<sup>[61]</sup>等。这类技术认为欧氏空间中的高维数据在黎曼空间中的存在特殊数据流形结构（即图结构）。流形学习通常基于预定义的相似度函数，根据数据的原始特征将构建亲和图来表达这种流形结构，然后通过某种映射保留亲和图上的邻域关系信息来学习节点的特征表示。然而，较高的计算复杂度阻碍了流形学习在现实大规模数据上的应用和发展。

#### 2.4.1.2 图嵌入

以 Word2Vec<sup>[62]</sup>为代表的 Skip-Gram<sup>[63]</sup>模型在图表示学习领域的广泛应用标志着该领域进入了第二个时代——图嵌入<sup>[12]</sup>时代，这类方法的目的是在低维向量空间中嵌入静态图的节点，可以在保留网络的拓扑属性和语义信息的情况下对整个图上的节点、边、子图进行分析。

Word2Vec 是一种从大量的文本中学习单词含义的方法，它通过保留人类语言中单词之间的共现关系来学习单词在特征空间中的表达向量。而 DeepWalk<sup>[64]</sup>巧妙地将图视为文本，将图中的节点视为语言文本中的单词，并通过在图上进行随机游走生成该语言的句子，这样就把图表示学习问题转化成了自然语言理解问题。具体来说，DeepWalk<sup>[64]</sup>首先在图上通过深度优先随机游走生成多条节点序列，对图结构信息进行采样，并保留了节点在序列中的共现关系，之后将节点序列放入从 Word2Vec 模型中学习，从而产生节点特征嵌入。



17010226

而 Node2Vec<sup>[65]</sup>平衡了图上广度优先采样和深度优先采样，以获得同源性和结构等效性的特性。此外，LINE<sup>[66]</sup>和 SDNE<sup>[67]</sup>基于一阶甚至更高阶节点相似性和临近性，而不是随机游走策略构建相似矩阵。也可以通过非负矩阵因式分解从邻接矩阵的拉普拉斯算子中导出表示<sup>[68-70]</sup>。在应用图嵌入方法解决具体的问题时，研究人员通常将其分为两个阶段问题。以节点分类为例，图嵌入方法在第一阶段为每个节点学习统一长度的特征表达向量，在第二阶段将节点表达作为输入，训练传统的机器学习分类模型。

然而，这些图嵌入方法存在以下三个弊端。第一，基于随机游走的网络嵌入方法由于其直推性的训练设置（Transductive Learning）而难以处理在训练时未见的节点，意思是每当图中新增加一批节点时，这些网络嵌入模型必须重新训练以适应新的网络结构，而不能直接根据现有的信息推理新的节点特征。第二，图嵌入方法只能处理图结构信息，而不能处理图的属性信息，这造成了结构和属性关系建模的割裂，降低了算法的表现。第三，图嵌入方法采用无监督学习以生成的节点特征，而这些节点特征对于下游的异常检测、节点识别或关系推理任务没有针对性的贡献，而是仅仅捕获了一些节点在网络结构上的通用特征。

#### 2.4.1.3 图神经网络

图神经网络（Graph Neural Network, GNN）将深度学习技术迁移到图数据上进行端到端的建模，是一种很有前途的方法。GNN 算子的设计方式主要分为两种，一种是基于谱图理论的图滤波器算子，它通过对于图拉普拉斯矩阵的分解和图傅立叶变换来提取有用的图信息；而另一种则是基于空间的图聚合算子，它通过聚合邻域和自身的信息来更新节点的特征以直接处理图的结构。在 GNN 的邻域聚合方案中，每个节点聚合其邻域的特征信息以逐层计算其新的特征向量。经过多次信息的消息传递和聚合迭代后，节点的特征向量将捕获节点邻域中的结构信息。

Yann Lecun 等人<sup>[71]</sup>首次系统地提出了基于谱图理论的图滤波器解决方案，作者设计了一个全通滤波器，利用一个多层次感知机来调整该滤波器的滤波系数。为了解决该文献面临的计算成本高的问题，文献<sup>[72]</sup>则提出了使用一组正交基的 K 阶截断的切比雪夫多项式函数来调整图滤波器的滤波参数。

根据信息聚合的方式，基于空间的 GNN 可以分为两种形式：各向同性模型和各向异性模型。各向同性模型假设每个邻居对中心节点的更新贡献相等。典型的各向同性 GNN 是图卷积网络<sup>[73]</sup>（Graph Convolutional Network, GCN）和图同构网络<sup>[74]</sup>（Graph Isomorphism Network, GIN）。图卷积网络利用基于谱方法的卷积



17010226

聚合器收集信息，而图同构网络在图卷积网络的基础上引入图同构判别算子，提高了 GNN 在图分类任务上的能力。第二类 GNN 模型为各向异性模型，其假设每个邻居对中心节点的重要性不相等，并使用不同的权重通过某种机制从相邻节点聚合信息。例如，图注意力网络<sup>[75]</sup>（Graph Attention Network, GAT）引入了一种注意力机制，其根据注意力得分的大小分配给每个不同邻居不同的重要性，或残差门控图卷积网络<sup>[76]</sup>（Residual Gated Graph ConvNets, GatedGCN）中的门控机制，根据节点的特征关联确定哪些节点的信息应该被保留。

一些工作针对 GNN 在大规模图数据上存在无法大规模化的问题提出了许多改进方法。例如 SAGE<sup>[77]</sup>，它是 GCN 的一种综合改进，它认为 GNN 在进行信息传播时，节点不需要聚合其全部邻居的信息，只需要采样少部分邻居，即可在满足一定的性能要求的前提下，大幅度提高模型的训练速度。顺着 SAGE 提出的“采样-聚合”的技术思路，FastGCN<sup>[78]</sup>、ClusterGCN<sup>[79]</sup>和 GraphSAINT<sup>[80]</sup>等模型通过使用不同的图采样方法改进了 GCN 模型在训练时的效率问题。

而另一些文献则认为 GNN 仍然存在表达能力不足的问题，这是因为每个节点在每层网络中的信息传播过程中只能访问 1 阶邻居，而不能自适应地访问 k 阶邻居，而不能直接访问 k 阶邻居就意味着节点不能学习自身的局部临域结构信息。因此，越来越多的研究人员，提出了例如跳跃知识网络（Jump Knowledge Network）<sup>[81]</sup>和非局部神经网络（Non-local Neural Networks, NLNN）等模型，他们试图在图神经网络层中添加更多的跳跃连接，使每个节点自适应地聚集更多来自 k 阶之外邻居的信息，来学习自适应的、结构感知的表示。另一些文献则通过提出结构化编码的方法，将节点局部子结构信息通过某种提出的编码方式加入到节点的特征学习中，这一类典型文献主要由 Distance Encoding GNN<sup>[82]</sup>、Position-aware GNN<sup>[83]</sup>以及 Random Walk GNN<sup>[84]</sup>等模型组成。

然而，所有这些 GNN 方法都侧重于学习通用的节点表示，可以推广到主要的下游任务，如节点分类、关系推理或者图分类，而目前尚不清楚如何挖掘 GNN 在图异常检测任务等专业下游任务中的潜力。

#### 2.4.2 图与异常检测

在本小节中，我们首先介绍了传统的数据异常检测的相关工作，接着，我们介绍了目前已有的在图数据挖掘领域中的少数异常检测工作。



17010226

#### 2.4.2.1 传统异常检测

数据泛滥的时代已经引起了人们对异常检测技术的兴趣，尤其是在多维数据点集合中发现异常<sup>[15]</sup>。一般来说，异常检测任务可以被视为一个特殊的二分类机器学习问题。异常检测任务假设训练集中所有的样本都是无异常的样本，或者只存在极为少量的异常样本，要求模型在没有（或有极少）异常数据的情况下，学会区分异常和正常数据的数据分布差异。这种定义是紧密贴近实际且合理的，因为在大部分系统中，鉴于获取真实的异常数据的成本高昂，有标记的异常数据极端缺乏，异常的样本数量远远少于正常的样本。比如在正常运作的工业生产系统中，存在故障的设备总是极少数；在数亿的社交网站用户中，机器人账户也非常的稀少。

一些工作<sup>[85-86]</sup>尝试将异常检测任务视为一种数据极端不平衡的有监督分类任务，而由于异常标签过少，各类工作都证明传统的有监督分类机器学习模型很难胜任；此外，基于规则的异常检测系统风险也极大，因为人为制定的规则引擎存在着规则设计不完备、规则从内部泄密和被犯罪分子针对性伪装的可能。

为了克服这些问题，大部分异常检测模型采取的策略是学习大规模正常数据的概率分布模型并进行归纳总结，一旦待测数据偏离已归纳的数据分布超过一定限度，即进行异常报警<sup>[15]</sup>。这些异常检测方法可分为五大类。

1. 基于概率建模的方法，例如混合高斯模型（Gaussian Mixture Model, GMM）<sup>[87]</sup>和概率密度估计技术（KDE）<sup>[88]</sup>，使用统计学方法估计正常数据的概率密度分布函数。拥有低概率密度的数据沉淀会被视为异常数据。
2. 基于距离的方法，如局部离群因子（Local Outlier Factor, LOF）<sup>[89]</sup>和孤立森林（Isolation Forest, IForest）<sup>[90]</sup>，通常假设正常数据紧密聚集，而异常数据远离其最近的邻居。这些方法通常会通过会定义不同的衡量两个数据点之间的相似度指标来取得更好的异常检测效果。
3. 基于边界的方法，例如支持向量数据描述（Support Vector Data Description, SVDD）<sup>[91]</sup>和一类分类支持向量机（One Class Support Vector Machine, OCSVM）<sup>[92]</sup>，通常试图围绕正常类数据定义边界。位于边界之外的未知数据被定义为异常值。
4. 基于重构的方法假设异常不能从低维投影有效地被重建。在这一类别中，主成分分析（Principal Component Analysis, PCA）<sup>[93]</sup>及其修改版本鲁棒主成分分析<sup>[94]</sup>（Robust Principal Component Analysis, RPCA）和核主成分分析<sup>[95]</sup>（Kernel Principal Component Analysis, KPCA）被广泛使用，是检测异常的有效技术。
5. 深度学习方法，如自动编码器<sup>[96-97]</sup>（AutoEncoder, AE）、变分自动编码器



17010226

(Variational AutoEncoders, VAE)<sup>[98-99]</sup>和生成性对抗网络 (Generative Adversarial Networks, GAN)<sup>[100-101]</sup>, 学习无异常数据集中的高维概率分布, 然后检测偏离该分布的异常值。由于随着数据量的增加, 浅层异常检测几乎不可能有效地发现异常值, 因此基于深度学习的异常检测算法受到了越来越多的关注。此外, Ruff 等人<sup>[102]</sup>将浅层 SVDD 方法扩展到了深层, 并在图像数据上显示了有竞争力的结果。

异常检测技术已经在包括互联网<sup>[103]</sup>、金融系统<sup>[104]</sup>、工业系统<sup>[105]</sup>、医疗健康管理<sup>[106]</sup>以及社会治安<sup>[107]</sup>等重点领域得到了广泛应用, 并为社会经济和国家安全做出突出了贡献。

然而, 传统的异常检测方法无法直接处理图数据, 而通常只能只能在2.4.1.2小节所述的图嵌入算法的帮助下处理图数据。首先, 图嵌入方法可以将节点之间的拓扑连接关系转换成为固定长度的图嵌入向量; 然后, 传统异常检测算法就可以使用图嵌入特征向量来间接地进行图上的异常检测。然而, 图嵌入方法学到的节点嵌入是与任务无关的而且没有专门针对异常检测任务做优化, 所以在经验上, 这类图嵌入方法针对图上的异常检测任务一般无法取得良好的效果。

#### 2.4.2.2 图异常检测

虽然传统异常检测算法已经被成功应用于各个领域, 但是如何将这类算法迁移到图数据仍然存在疑问。图数据的复杂性对现有的机器学习算法提出了重大挑战。由于图可能是不规则的, 因此图可能具有可变大小的无序节点, 并且图中的节点可能具有不同数量的邻居, 从而导致一些重要的操作 (例如卷积) 在欧氏数据域中易于计算, 但是应用于图域却是困难的。并且, 传统的图像、文本等数据假设各个数据点之间是互相独立的, 而图的节点之间却具有相互的长程作用关系。

现有的图异常检测方法<sup>[13]</sup>可分为三类: 基于结构的、基于子空间的和基于残差的。

1. 早期的图异常检测方法关注如何发现图网络中的社群结构, 假设异常节点总是从属于一个密集异常社群, 先挖掘社群再挖掘异常节点。他们仅使用图上的结构信息, 如社群分析<sup>[108]</sup>和中心子图 (ego-net) 的特征<sup>[109]</sup>来检测图上的异常。
2. 除此之外, 研究人员还发现, 属性网络中的异常现象可以在节点属性空间中的子空间中被发现, 例如<sup>[110]</sup>。虽然这些基于子空间的方法可以获得更好的结果, 但它们使用的网络结构信息很少, 这无疑会降低图数据的学习效率。
3. 最近, 基于残差分析的方法引用<sup>[111-112]</sup>和图自动编码器方法<sup>[113-114]</sup>利用了图结



17010226

构信息及其与节点属性的一致性，并对图异常检测社群显示出了越来越大的影响。这类方法的基本假设是正常节点的特征可以被其他相关节点的特征及其连接关系所近似表示，而异常节点则更加独立。然而，前一种方法使用矩阵分解表征和残差估计技术，但这些简单的残差分析机制不足以满足当今对大规模非线性数据的计算需求。另一方面，图自动编码器方法的目标是学习低维节点嵌入，重建节点和节点之间的连接关系，这对属性图上的异常检测没有直接的贡献。

然而，在图异常检测任务中仍然存在着几个严峻的挑战：(i) 数据非线性。现实场景中节点特征和节点间的相互作用通常是高度非线性的，而许多基于启发式的图异常检测方法<sup>[16]</sup>依赖于线性机制，其通常没有复杂的非线性表达能力。(ii) 计算复杂性。随着大规模图数据的应用场景越来越多，真实的图系统很容易有数百万个乃至数亿个节点和边。经典图异常检测方法通常依赖于高复杂度的图核<sup>[18]</sup>或者图傅立叶变换<sup>[17]</sup>，过高的计算开销阻碍了它们对大规模图的适用性，其应用在如淘宝、微博、微信等大规模社交电商网络中会面临着严重的运行效率问题。(iii) 研究人员和开发人员精通简单的现成欧氏数据的异常检测技术。基于统计图特征的图异常检测方法引入了大量图统计理论和图频谱领域相关的超参数，增加了它们的学习、理解和部署成本，削弱了人们将其在产业界中大规模推广和应用的动力，并进一步地限制了这类算法在理论上的创新速度。

我们认为，GNN 是解决以上问题的关键工具。由于 GNN 的本质是融合了图信息的神经网络，其天然具有高度的非线性建模能力。其次，神经网络在当今时代已经被证明是可以在大规模数据下使用的，其优秀的并行训练和部署能力非常适合于大规模云计算的技术发展路线。最后，由于 GNN 能够自动提取图结构特征，并且 GNN 的超参数非常少，具有很低的参数敏感性。如果能将 GNN 融入到异常检测任务中，则用户可以直接使用封装好的 GNN 模型做建模，而无需深入理解图理论。

### 2.4.3 动态图学习与链路预测

动态图嵌入学习是指，通过某种下游的学习任务，将动态图中的节点投影表示到欧氏向量空间中。而通常来说，由于动态图中最能提现其动态性的就是链路关系，因此链路预测任务就成为了当前动态图学习模型最常采用的下游任务。因此，在本小节中，我们将首先介绍在静态图中进行链路预测的经典方案，接着，我们将介绍目前主流的动态图嵌入方法，这些方法的共同点在于他们几乎都采用了链路预测任务作为模型的训练目标。



17010226

#### 2.4.3.1 链路预测

链路预测，又叫关系推理，是预测网络中节点之间丢失或未来的链路。它有着广泛的应用场景，如社交网络中的朋友推荐<sup>[115]</sup>、Netflix 中的电影推荐<sup>[116]</sup>、蛋白质相互作用预测<sup>[117]</sup>和药物反应预测<sup>[118]</sup>。传统的链路预测方法包括启发式方法、基于图嵌入的方法和基于图深度学习的方法。启发式方法通过某种启发式规则计算节点相似性，并将该相似性得分作为存在链接概率值<sup>[119]</sup>，例如共同邻居、优先连接<sup>[120]</sup>和 Katz 指数<sup>[121]</sup>。它们可以看作是一些预定义的图结构特征。

基于图嵌入的方法，如矩阵因子分解机（Matrix Factorization, MF）<sup>[122]</sup>和 node2vec<sup>[65]</sup>，以一种矩阵变换的方式，并以一种无模型的方法学习节点嵌入。然后人们可以通过对学习到的节点嵌入计算内积等方式，定量估计节点之间的潜在关系进而推测它们之间的链路存在性。

最近，图神经网络（GNN）成为学习图结构化数据的强大工具<sup>[10]</sup>。通过以统一的方式学习图结构和节点特征，GNN 在链路预测任务中表现出了优异的性能<sup>[83,123-125]</sup>。基于图深度学习的链路预测方法主要有两种。一种是图自动编码器（GAE）<sup>[123]</sup>，其中 GNN 首先应用于整个网络以计算每个节点的表示。然后对源节点和目标节点的表示进行聚合，以预测目标链路。它的变分版本被称为 VGAE。第二种类型是 SEAL<sup>[124]</sup>，在每个目标链接周围提取一个局部封闭的子图。然后，每个封闭子图中的节点根据其到源节点和目标节点的距离进行不同的标记。最后，将 GNN 应用于每个封闭子图，学习用于链接预测的链接表示。乍一看，GAE 和 SEAL 看起来非常相似，它们都使用 GNN 来了解目标链接周围的局部结构和特性。然而，正如我们将看到的，SEAL 从根本上说具有更好的链接表示学习能力。关键在于它的节点标记步骤。

然而，上述的链路预测方法无法直接利用动态图上的动态性链接关系，所以也很难执行动态图嵌入学习的任务。

#### 2.4.3.2 动态图嵌入学习

动态图嵌入算法的提出是为了处理图上节点之间成批的时序交互  $(v^s, v^d, t)$ ，其旨在利用结构信息和时态信息来学习节点嵌入。由于复杂的网络演化和动态性，只有少数关于表征学习的研究能够满足现实系统中时间捕获的需求。有一些文献试图在静态图嵌入模型中融入动态的建模特性。CTDNE<sup>[126]</sup>算法在静态图随机游走模型 DeepWalk<sup>[64]</sup>之上加入了沿时间路径动态游走的概念，因果匿名游走（Causal Anonymous Walks, CAW）模型<sup>[127]</sup>提出了因果匿名游走机制并融入 skip-gram 模型<sup>[62]</sup>，以学习动态图上的因果关系以及动态模式和特征。



17010226

而在另一些文献中，动态图神经网络（Temporal Graph Neural Network, TGNN）通过在 GNN 的消息传递过程中加入时态信息，将 GNN 扩展到动态图领域中。JODIE<sup>[21]</sup>使用循环神经网络更新相关节点的节点状态状态，以达到跟踪每个节点的历史信息的目的。TigeCMN 模型<sup>[25]</sup>使用了基于注意力机制的键-值对记忆网络<sup>[128]</sup>来更新节点的状态。相同点在于，这两篇文献都引入了一种注意力结构来读取节点状态状态并生成最终的节点嵌入向量。然而，JODIE 和 TigeCMN 没有明确地学习图的拓扑结构信息，因为它们只更新边的相关两个节点，这意味着它们不能直接遍历两跳邻居。

动态图注意力（Temporal Graph Attention, TGAT）模型<sup>[22]</sup>引入了傅立叶时间编码核方法来增强静态的图注意力模型对于时间建模的能力。动态图网络（Temporal Graph Neural Network, TGN）<sup>[24]</sup>结合了 JODIE 和 TGAT 的优点，将节点上的内存更新策略引入 TGAT 的时间聚合阶段。简而言之，JODIE 和 TigeCMN 本质上是基于序列建模的动态图算法，而 TGAT 和 TGN 是基于图聚合的动态图算法。

综上所述，上述的动态图嵌入方法均是试图利用某种动态的模型试图捕获动态关系，从而解决静态的链路预测模型无法很好解决的问题。然而，这些方法所处理的任务在本质上仍然是链路预测，即给定源节点  $v^s$ ，预测目标节点  $v^d$  的问题，没有考虑动态图上最重要的属性——时间。现有的动态图方法的数据输入虽然包含时间信息，但训练目标还是传统的静态链路预测。在本论文的第四章，我们提出的动态图嵌入学习方法在预测任务上综合考虑了时间与空间的关系，在数据的输入和训练目标的输出都是动态的，在动态图的建模上具有显著的优势。

#### 2.4.4 动态图关系演化与随机点过程

大多数现实生活中的图系统是动态的，例如，在社交网络中，由于热点事件，用户通常会在短时间内将兴趣转移到其他实体；在经济网络中，欺诈者往往会突然实施一系列犯罪，然后在最短的时间内提取非法资金。在由许许多多实体组成的大规模系统中引入图数据建模，不仅可以在考虑实体自身的性质的同时引入静态的关系性建模，而且需要考虑历史事件和即将到来的实时事件的影响，并可以利用这种关系进一步对该系统的未来演化进行预测。

在本节中，我们首先阐述了随机时序点过程的基本原理以及在动态图上应用所面临的固有弊端；最后，我们进一步地介绍了随机时序点过程在和在动态图关系演化预测中的重要性。



17010226

#### 2.4.4.1 随机时序点过程

如果我们需要在判断动态图中下一个所发生的事件是什么，并且预测该事件什么时候发生，能实现这个目的的最方便的工具就是随机点过程。随机点过程不同于传统的时间序列预测，时间序列预测预测的是“下一步”会发生什么事件，是一种离散时间的预测；而随机点过程预测的是下一个事件是什么，在何时发生，是一种连续时间的预测。

时序点过程<sup>[129]</sup> (temporal point process, TPP) 是一种随机过程，旨在建模一个事件序列所发生的时间戳  $t_1, \dots, t_n$  的概率分布。TPP 的建模核心是条件强度函数  $\lambda_i(t)$ ，根据该函数，在给定历史交互的时间戳信息  $\{t_i : t_i < t\}$  的情况下，计算发生在  $t$  和  $t + dt$  之间的事件的条件概率，其中  $dt$  是一个很小的时间间隔。强度函数既可以预先定义，也可以是可训练的参数，预定义的条件强度函数主要包括泊松过程、霍克斯过程<sup>[130]</sup>、瑞利过程等。根据文献<sup>[131]</sup>，在时间  $t$  发生的事件的对数条件概率密度可以公式化为

$$f(t) = \log \lambda(t) - \int_{t_n}^t \lambda(\tau) d\tau \quad (2-11)$$

此外，标记时序点过程 (Marked Temporal Point Process, MTPP) 将每个事件与通常被视为事件类型的“标记”(marker)  $y_i$  相关联。因此，MTPP 不仅对事件发生的时间进行建模，还对事件的类型进行建模。如果将图上所发生事件的节点对作为标记，则动态图上的事件预测也可以被建模为一个 MTPP。整个 MTPP 的条件强度可以记为每个单独标记的条件强度之和： $\lambda = \sum_m \lambda_m$ 。这使得它可以首先预测事件时间，然后以时间为条件，通过从分类分布中取样来预测标记，即  $m \sim \text{Categorical}(\lambda_m)$ ，这种模式成功地降低了模型复杂度，避免了时间和标记联合建模所带来的额外复杂度。这种思想已被广泛应用于后续工作中，如循环标记时序点过程 (Recurrent Marked Temporal Point Process, RMTPP)<sup>[33]</sup>。RMTPP 使用递归神经网络 (RNN) 参数化建模条件强度和标记分布。RMTPP 的变体包括 CyanRNN<sup>[132]</sup> 和 ARTPP<sup>[133]</sup>。

具体来说，RMTPP 模型中的 RNN 模块首先将事件序列  $(y_1, t_1), \dots, (y_n, t_n)$  编码为隐藏表示  $h_n$ 。条件强度函数  $\lambda_{n+1}(t)$  和标记  $y_{n+1}$  的生成分布表示为：

$$\begin{aligned} y_{n+1} &\sim \text{Softmax}(\mathbf{W}_{\text{marker}} h_n + \mathbf{b}_{\text{marker}}) \\ \lambda_{n+1}(t) &= \exp(\mathbf{M}_{\text{time}} h_n + \mathbf{W}_{\text{time}}(t - t_n) + \mathbf{b}_{\text{time}}) \end{aligned} \quad (2-12)$$

其中所有的  $\mathbf{W}, \mathbf{M}, \mathbf{b}$  是可训练的参数，Softmax 是从分类概率函数，其根据每种类别的特征分布将其映射到概率空间中。它利用递归神经网络和可学习嵌入方法对不同类型事件的复杂强度函数进行建模。该模型在时间预测和标记选择上均采用



17010226

极大似然估计进行优化。在金融和医疗数据集上的实验证明了它在相对较少的分数和短序列上建模和预测事件的有效性。

此外，我们还演示了与 TPP 相关的其他一些相关工作和应用<sup>[134]</sup>。其中一个是 MTPP 的变种 CoEvolving<sup>[35]</sup>，它使用 Hawkes 过程分别对用户项和商品项的交互进行建模。由于原始的霍克斯过程认为强度函数只能增加不能减少，神经霍克斯过程（NeuralHawkes）<sup>[34]</sup>通过引入自调节模型，根据过去的历史随机调整未来事件的强度，去除了霍克斯点过程中的正向影响假设。文献<sup>[135]</sup>使用 Wasserstein 距离<sup>[136]</sup>作为度量，使用生成性对抗学习的方法隐式学习强度函数，并获得了更好的性能。DeepTPP<sup>[137]</sup>将事件生成问题建模为一个随机策略，并应用逆强化学习来有效地学习 TPP。LANTERN<sup>[138]</sup>扩展了之前的 MTPP 工作，提出了一个潜在结构强度模型，该模型在没有明确的图结构情况下估计事件之间的关系，以建立强度函数。文献<sup>[36]</sup>通过 TPP 在社交网络上建立链接和转发预测模型，并提供从 TPP 模型生成社交网络交互的模拟仿真算法。它与我们的任务非常相似，只是它专注于特定的社交网络环境，并且没有定量评估其模拟模型的质量。

然而，由于以下原因，这些方法不能直接应用于我们所提出的动态关系预测和关系演化预测任务。首先，动态图本质上是一个由动态边组成的事件序列，其长度从数万到数亿不等。过去使用 RNN 建模时间序列的方法都是将序列划分为多个较短的窗口，这将使事件与事件之间的关系在给定窗口内断开，所以 RNN 在处理很长的序列时有困难。因此，在动态图学习的领域中，这无法探索随时间上距离遥远但具有拓扑连接（即共享任一节点）的事件之间的依赖关系。还可以考虑在长序列中用截断的 BPTT<sup>[139]</sup>来训练一个 RMTPP，但由于其不支持并行化的训练，所以一次只能展开一个事件，毫无疑问这是重复性且低效的。其次，直接建模标记生成分布将生成一个空间复杂度为  $O(|V|^2)$  的标记向量，因为事件标记建模的是图上所有的节点对。这对于大型图数据来说是不可取的。

#### 2.4.4.2 动态图上的随机时序点过程

有一些图模型试图将随机时序点过程<sup>[129]</sup>（temporal point process, TPP）的机制融入动态图模型之中，从而达到预测时间戳的目的。其中，将点过程应用与动态图领域最重要的步骤就是条件强度函数  $\lambda_i(t)dt$  的设定。

过去的许多文献<sup>[26,28-29]</sup>将动态图中新节点和边的出现建模为随机点过程，他们认为图上的事件是一种离散事件，但是其事件发生的时间戳是连续的，自然而然地是符合随机时序点过程的定义。这些文献使用循环神经网络的体系结构参数化更新条件强度函数  $\lambda_i(t)$ ，从而估计在一个小时时间窗  $dt$  内观察事件的条件概率，



17010226

近似估计图上的时序点过程。例如，DyRep<sup>[26]</sup>联合学习节点之间的拓扑演化和活动，其中节点  $v_i$  的特征嵌入表示在  $v_i$  涉及到某个事件后更新。文献<sup>[28]</sup>提出了图偏向的时序点过程（Graph Biased Temporal Point Process, GBTTP）用于建立事件是否从节点  $v_i$  传播到其邻域  $N(v_i)$  的概率模型。

然而，大多数将点过程和图学习相结合的文献为了建模图上的事件序列之间的时间依赖关系，通常都采用了循环神经网络这类递归结构建模。在递归网络结构中，历史信息需要逐个顺序地传递到递归单元，这使得它们无法并行地、分布式地进行随机小批量训练，这成为在图学习领域一个不可接受的特性，尤其是在大规模图的训练中。

MMDNE<sup>[30]</sup>、HTNE<sup>[31]</sup>以及 DSPP<sup>[32]</sup>试图通过类似注意力机制聚合邻域信息，然后直接生成强度函数，以此定义时序点过程捕捉图上的事件以及演化。他们成功避免了在建模时序点过程时使用循环神经结构。然而，他们的任务目标仅仅在于静态的关系推理或者时间戳预测，将他们的方法直接应用在动态关系推理或关系演化预测任务中是困难的，需要进行非常非同寻常的更改。

#### 2.4.5 图深度学习算法的加速

对于传统的欧氏神经网络来说，加速模型推理的最主要方法就是网络压缩<sup>[140]</sup>方法。网络压缩方法主要包括网络剪枝<sup>[141]</sup>、参数量化<sup>[142]</sup>以及知识蒸馏<sup>[143]</sup>等。这些方法的共同目的就是在保证准确率不大幅下降的情况下，使用一个小容量的模型来替代原本的大模型。然而，网络压缩方法压缩的对象是模型规模本身，而不是大规模的图。图神经网络在实际部署中面临的复杂度瓶颈只跟节点数量  $|\mathcal{V}|$  和边数量  $|\mathcal{E}|$  有关，与图神经网络本身的规模无关。这是因为图神经网络的宽度本身可能是上百最多上千，而在大规模图中的节点数目可以轻松破亿。所以网络规模在复杂度分析中可以忽略不计，图的规模才是我们主要考虑的部分。因此，在欧氏模型中发光发热的网络压缩方法到了非欧的深度学习模型上失效了，被排除在本章的讨论范围之外。

已经有一些研究者采取措施来克服这些非欧数据的图深度学习算法中存在的速度问题，Sergi 等人<sup>[40]</sup>在综述中归纳总结了用以提高 GNN 模型训练和推理能力的各种软件和硬件加速方案。而这些优化思路主要有以下三类。第一，利用专用硬件电路加速，设计并封装适用于图模型稀疏计算的计算核。Zhang 等人<sup>[144]</sup>以及 Hanqing 等人<sup>[41]</sup>使用 CPU-FPGA<sup>[145]</sup>的联合计算架构分别研究 GNN 在训练和推理步骤的加速。而 EnGN<sup>[42]</sup>、HyGCN<sup>[146]</sup>和 GCNAXN<sup>[147]</sup>等文献则分别提出新颖的计算结构去适应图上的某些稀疏化操作。第二，在软件方面，由于图深度学习模型需



17010226

要交替地进行图的稀疏运算以及稠密矩阵神经网络运算。传统的深度学习库，如 Pytorch<sup>[148]</sup>和 Tensorflow<sup>[149]</sup>，擅长加速顶点和边上的稠密矩阵计算操作，但在图的聚合过程中表现不佳。而图形处理框架<sup>[150-151]</sup>在遍历图时可以很好地管理不规则的内存访问，但在矩阵运算方面效率低下。为了弥合这一差距，亚马逊公司的 DGL<sup>[43]</sup>、阿里巴巴集团的 AliGraph<sup>[44]</sup>和腾讯的 AGL<sup>[45]</sup>等图深度学习加速框架从系统的底层封装图计算常用的稀疏算子，将这些算子融合到传统的深度学习库中，达到同时加速稠密特征矩阵运算和稀疏图计算的目的。第三，大幅提升图数据库的性能。比如使用一些注重实时图计算的数据库产品，比如 Ultipa 和 TigerGraph 等<sup>[46]</sup>，它们专门优化了数据实体之间的随机访问性能，从而为图模型提供了更加快速的数据供给能力。然而，由于上述 GNN 计算的优化仍然受到基本 GNN 框架的限制，而在这种条件下不论是在软件、硬件层面上优化最终都有一个理论优化上限，而通常来说该上限还远远达不到图模型的时效性要求。此外，根据综述文章<sup>[40]</sup>4.3 小节的结论，这些软件硬件的框架普遍存在度量指标不统一、适用范围窄以及不支持动态图建模的问题。

## 2.5 本章小结

本章首先简要介绍了图的基本概念，图的分类学、图的矩阵表示。然后，本章从数学理论上介绍了图深度学习模型的两种最基本的范式，有助于读者更好地理解本论文所提出的新方法。此外，本章引入了本文的三大主要任务，动态异常检测、动态关系推理以及关系演化预测的定义，以及这三种任务之间在理论上以及应用上的联系。这为下文第三、四、五、六章的叙述铺平了道路。之后，我们针对本文从第三到第六章的主要内容，我们梳理了目前最新的相关文献，并且说明了这些文献中存在的性能、效率以及适用范围狭窄的等等问题。



17010226

## 第三章 基于超球面学习的图异常检测

### 3.1 引言

如今，图结构数据越来越多地被用于复杂系统的建模，比如社交媒体网络<sup>[152]</sup>，交通网络<sup>[153]</sup>和金融网络<sup>[154]</sup>。图是由一组节点  $\mathcal{V}$  构成的结构，其中一些节点对由边集  $\mathcal{E}$  关联。通过引入实体对象之间的边，图为有效捕获它们之间的相关性提供了一个强大的工具。同时，从图中检测异常已经成为迫切需要社会关注的一个重要研究问题<sup>[13]</sup>。首先，具有异常的图数据会破坏图机器学习算法的性能，并带来严重后果。其次，图异常检测（Graph Anomaly Detection, GAD）在许多安全相关领域有着重要的应用，例如发现可疑的金融交易、监控交通堵塞以及揭露社交网络中的恶意用户。因此，在这项工作中，我们研究了一个新颖的问题，即探索在图上进行异常节点检测。

在欧氏机器学习领域中，异常的定义就是与大多数其他数据不同的稀有数据，异常检测任务已应用于多种不同的领域<sup>[15]</sup>。解决异常检测的问题主要分为两种手段。第一是在有监督算法的框架下，通过对稀有的异常数据采用 SMOTE<sup>[155]</sup>、ADASYN<sup>[156]</sup>或生成对抗网络<sup>[157]</sup>等上采样方法，从而修正异常数据与正常数据之间的规模不平衡性。而第二种方法，我们已经在2.4.2.1小节中对传统的异常检测算法进行了概括性的描述，这类方法是通过某种半监督的学习算法描述正常数据的边界或者统计量，然后通过数据偏离该边界的程度来确定异常，比如混合高斯模型<sup>[87]</sup>以及主成分分析<sup>[93]</sup>等等。

在欧几里得空间中，多维数据被视为是独立数据点分布在空间中；然而，图中的节点可能表现出相互依赖性，这意味着它们内在地相互关联。图上的异常检测同样地也拓展了欧氏数据中的异常检测概念。图上的异常节点有可能是在本身的特征和属性上与其他节点存在不同，也有可能是节点与其他节点的连接关系存在异常。而在传统欧氏数据中广泛适用的异常数据上采样方法在图这种非欧数据域上很难适用，其原因在于以下两点。首先，上采样方法很难为合成的新样本产生关系信息，就算生成了新的节点，算法也不知道这个新生成的节点会如何跟其他节点连接。第二，数据上采样技术在本质上都是目标样本和其最近的邻居之间的插值来生成新的训练样本，而图上的边却是稀疏的、不连续的，盲目地插值会破坏其原本的拓扑结构。

对于第二类基于半监督学习的异常检测算法来说，虽然它们只能处理欧几里得数据（例如图像、文本和语音），而不能直接处理非欧几里得的关系型数据（图），



17010226

但是只要与图学习技术相结合，此类方法就可以在图异常检测的领域中广泛使用。比如将异常检测与图嵌入方法结合。图嵌入方法，如 DeepWalk<sup>[64]</sup>、Struc2Vec<sup>[158]</sup>以及 LINE<sup>[66]</sup>，可以将节点关系信息提取为固定长度的特征向量。然后，传统的异常检测方法可以借由该向量得到训练。然而，这种两阶段的方法学习到的特征向量与异常检测任务是解耦的，因为图嵌入学习方法是通用的、与任务无关的，不是为图异常检测定制的，对检测异常节点的贡献是有限且间接的。再比如另一些图异常检测文献<sup>[13]</sup>采用图统计特征提取与异常检测技术相结合的方案。通常采用图矩阵分解来定位异常节点或首先找到目标群体，然后在预定义的子空间中检测异常值。然而，GAD 任务中存在几个挑战：(i) 数据非线性。节点特征和节点间的相互作用通常是高度非线性的，而许多 GAD 方法依赖于线性机制。(ii) 计算复杂性。在数据泛滥的时代，真实的图很容易有数百万个乃至数亿个节点和边。经典 GAD 方法的计算开销阻碍了它们对大规模图的适用性。(iii) 研究人员和开发人员精通简单的现成欧氏数据的异常检测技术。以前的基于统计图特征的 GAD 方法增加了它们的理解和部署成本，这极大地限制了它们在图挖掘领域的发展。

图神经网络（Graph Neural Networks, GNNs）<sup>[10]</sup>是直接对图结构数据进行运算的重要一步，也是解决上述问题的一种很有前途的方法。GNN 本质上是一种消息传递（Message Passing）方案，其中每个节点聚合其邻居的特征信息以计算其新的特征向量。经过多次信息聚合迭代后，节点的特征向量将捕获节点邻域中的结构信息。GNN 具有非线性激活函数和并行化能力，分别可以解决数据非线性和计算复杂性问题。尽管 GNNs 在许多图挖掘任务中表现出了优异的性能<sup>[152-154]</sup>，但绝大部分图深度学习方法<sup>[10]</sup>采用完整标签或无标签的情况下训练范式，而在很多场景（比如图上的异常检测）中，某几类待识别的对象可能并不存在或者仅有少量标签，这通常会导致模型训练得到的是对整体数据的有偏估计。虽然已经有许多文献<sup>[14]</sup>在欧氏深度学习方法上提出了改进策略以试图解决有偏性的问题，但将这些策略直接套用到非欧数据域上没有经验和理论上的保证，很少有文献探究图深度学习在标签不完全情况下的建模能力。

据我们所知，将 GNN 应用于 GAD 任务并解决数据不平衡问题的进展甚微。Ding 等人<sup>[113]</sup>以及 Li 等人<sup>[114]</sup>尝试使用图自动编码器<sup>[123]</sup>（Graph Auto-Encoder, GAE）来解决这个问题。他们使用一种简单的 GNN 形式——图卷积网络<sup>[73]</sup>（Graph Convolutional Network, GCN）——来组成自动编码器的编码器部分和解码器部分。图自动编码器的目标是将完整的图上的完备边集删除一部分，然后试图通过图编码器降维再通过图解码器升维，试图补充残缺的边集从而还原完整的图。因此，图自动编码器可以从图的结构信息以及节点属性中提取节点嵌入。然而，图自动编



17010226

码器方法的目标是学习低维节点嵌入，重建节点属性和节点间的连接关系，而不是直接针对异常检测任务。这种方法存在一个上下游任务不统一的问题，会造成性能损失以及可解释性降低。

人们需要一种端到端、易于理解且功能强大的图异常检测技术。在这项工作中，借由传统的机器学习方法 OCSVM<sup>[92]</sup>给我们的启发，我们提出了用于一类分类<sup>①</sup>的图神经网络（One-Class Graph Neural Network，OCGNN），它是一种基于超球面的图学习框架也是 OCSVM 在图数据领域的自然扩展。

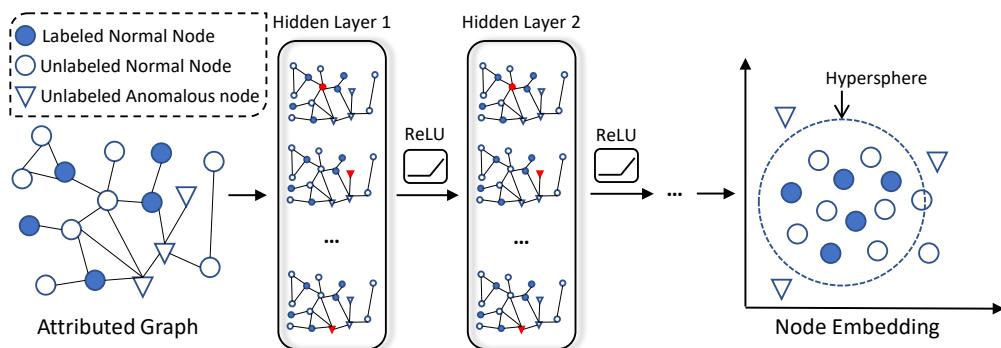


图 3-1 基于超球面学习的图异常检测模型的整体架构图

Figure 3-1 The overall framework of the proposed OCGNN

如图3-1所示，OCGNN 旨在将图神经网络强大的表示能力与经典的超球面学习目标相结合，以检测异常。由于图上的拓扑信息由 GNN 自动提取，OCGNN 的用户可以在不深入理解图论的情况下处理图异常检测的问题。此外，OCGNN 是一种端到端的方法，因为 OCGNN 学习到的节点表示与图异常检测任务高度相关，这意味着从 OCGNN 学习到的嵌入比从 DeepWalk<sup>[64]</sup>等方法学习到的嵌入对下游异常检测任务更友好。由于 OCGNN 可以从数据集中提取特征，并在没有标签信息的情况下学习节点嵌入，因此 OCGNN 也可以被视为一种针对图异常检测设计的半监督节点嵌入方法。

在提出的超球学习目标的指导下，通过聚集邻域信息，OCGNN 学习将每个可观测节点映射到特征嵌入空间，并将正常节点的嵌入包围到以向量  $\mathbf{c}$  为圆心、半径  $r$  为半径的超球中。同时，超球学习目标是在欧氏的特征空间中，最小化包含这些节点特征嵌入向量的超球的体积。而在特征空间中，节点的嵌入位置如果在超球之外，那么该节点就将被视为异常节点。OCGNN 是一个基于超球面学习的通用图神经网络框架，因此人们可以在 OCGNN 中使用任何形式的现有图神经网

<sup>①</sup> 一类分类（One-Class Classification）即只在提供一种正常类别数据的情况下，识别出其他异常类别的分类方法



17010226

络层结构。OCGNN 框架除了具有良好的对于几乎任何图神经网络模型的良好适应能力以外，OCGNN 甚至还拥有从静态图像动态图迁移的能力。在本章中，我们不光在几个静态的图数据集中测试了 OCGNN 的性能，还额外测试了 OCGNN 在动态图上的异常检测能力。

具体来说，我们的贡献如下：

1. 我们提出了一种新的端到端 OCGNN 框架，旨在将图神经网络强大的表示能力应用到图异常检测任务中。并且克服了图深度学习在异常检测领域中存在的数据不平衡难以学习的问题。
2. 我们提出了超球面学习（Hypersphere Learning）目标来驱动图神经网络的训练。由于图数据中的结构信息会由图神经网络自动提取，人们可以直接使用 OCGNN 来处理图异常检测问题，而不需要任何复杂的图论基础，这将极大地促进图挖掘技术的进步和应用的发展。
3. 我们的 OCGNN 的源代码已经在一个公开的代码存储库中发布<sup>①</sup>。此外，我们还提供了多达 14 种不同的基线模型，以尽量探究传统的异常检测方法在图异常检测领域能达到的边界性能。
4. OCGNN 框架可以由几乎任何的 GNN 层范例构成，同时 OCGNN 可以胜任静态图和动态图两种不同的场景。为了验证 OCGNN 框架的通用性，我们测试了三种流行的 GNN 层的性能，并使用三个广泛使用的公共图数据集来说明 OCGNN 的优越性；并且我们还验证了 OCGNN 框架在动态图数据集上的有效性。

本文的其余部分组织如下。第三节阐述了我们的一类分类图神经网络（OCGNN）方法。在第四部分中，我们在数个真实数据集上进行了实验，以展示我们的方法的竞争性和效率；最后，在第五部分，我们提出了我们的结论。

## 3.2 基于超球面学习的图异常检测模型

除了本论文主要符号对照表中所注解的符号，为了便于检索，我们在本章中额外总结了其他的一些本章常用的符号，如表格3-1所示。

### 3.2.1 问题定义

在本节中我们首先明确了本章所处理的图数据的特点，然后介绍图异常检测任务的定义。

<sup>①</sup> <https://github.com/WangXuhongCN/OCGNN>



17010226

符号	描述
$\mathcal{V} = \{v_1, \dots, v_N\}$	图中的节点个数
$\mathcal{V}_{tr} \subseteq \mathcal{V},  \mathcal{V}_{tr}  = K$	图中的可供训练的节点集
$\mathbf{F}^v \in \mathcal{R}^{N \times D}$	节点特征矩阵
$\mathbf{A} \in \mathcal{R}^{N \times N}$	图的邻接矩阵
$g(\mathbf{F}^v, \mathbf{A}; \mathcal{W})$	一个图神经网络
$\mathcal{W} = \{\mathbf{W}^{(1)}, \dots, \mathbf{W}^{(L)}\}$	$g$ 网络的可训练参数
$\mathbf{W}^{(l)}$	图神经网络第 $l$ 层的权重 $r$
$\mathbf{Z} \in \mathcal{R}^{N \times F}$	节点嵌入矩阵
$[\cdot]^+ = \max(0, \cdot)$	非负运算符
$c \in \mathcal{R}^F$	超球面球心
$r \in \mathcal{R}^+$	超球面半径

表 3-1 本章的常用符号对照表

Table 3-1 Commonly Used Notations in this Chapter

**定义 3.1 (属性图 (Attributed Graph) )** 本章中所描述的属性图均指节点被赋予了特征向量用来描述该实体特征的图。属性图  $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{F}^v)$  不光存在节点集和边集，还存在着节点的特征矩阵  $\mathbf{F}^v \in \mathcal{R}^{N \times D}$ 。其中  $N$  为图上的节点个数， $D$  为节点特征的特征维数。

节点的异常检测通常都跟复杂系统的安全性挂钩，比如在金融交易网络中识别异常的欺诈、洗钱、赌博用户，在电信网络中识别黑客或者宕机服务器节点等等。

**定义 3.2 (图异常检测)** 给定图  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ ，该图既可以是属性图也可以是无属性图。图中仅仅只有一部分节点的标签是可见的，并且这部分标签可见的节点都属于无异常的节点，即  $y_i = 0, \forall v_i \in \mathcal{V}_{train}, \mathcal{V}_{train} \subset \mathcal{V}$ 。节点层面的学习任务旨在给每个无标签的节点  $v_i$  分配一个标签  $y_i = 0 \text{ or } 1$ ，0 代表正常，1 代表存在异常。

### 3.2.2 超球面学习

超球学习最初是在数据向量描述方法 (SVDD)<sup>[91]</sup> 中提出的，其目的是学习一个紧凑的超球边界以覆盖所有训练数据，并检测哪些 (新) 对象类似于此训练集。在异常检测的应用中，训练数据都是正常的样本，因此超球面学习模型可以获得正常数据样本的描述边界，以学习如何区分异常。



17010226

我们令  $\mathcal{X} \subseteq \mathcal{R}^d$  为训练数据集的数据空间,  $\phi_k : \mathcal{X} \rightarrow \mathcal{F}_k$  是一个从数据空间  $\mathcal{X}$  到希尔伯特<sup>[159]</sup> (Hilbert) 特征空间  $\mathcal{F}_k$  的映射函数, 其中  $k : \mathcal{X} \times \mathcal{X} \rightarrow [0, \infty)$  是一个正定的核函数。SVDD 的目标是在希尔伯特特征空间  $\mathcal{F}_k$  中描述一个最小的、球心为  $c \in \mathcal{F}_k$  和半径为  $r > 0$  超球面, 该超球面可以容纳大部分的训练数据。

给定训练数据集  $\mathcal{X}_K = \{x_i \in \mathcal{X}, i = 1, \dots, K\}$ , 为了达成以上的训练目标, SVDD 定义了如下形式的数学求解问题:

$$\begin{aligned} & \min_{r, c} r^2 + \frac{1}{\beta K} \sum_{i=1}^K \xi_i \\ \text{s.t. } & \|\phi_k(x_i) - c\|_{\mathcal{F}_k}^2 \leq r^2 + \xi_i, \xi_i \geq 0, \forall i \end{aligned} \quad (3-1)$$

其中  $\xi_i$  是非负松弛变量, 该变量使得数据点  $\phi_k(x_i) \in \mathcal{F}_k$  并不会严格地被限制在超球体内部, 但是, 距离边界太远的数据会受到惩罚。超参数  $\beta \in (0, 1]$  控制了球体体积和惩罚之间的权衡, 如果  $\beta$  趋紧于 1, 则惩罚项趋紧于 0, 模型所学习的超球体积会过于小以至于失去意义; 反之, 如果  $\beta$  趋紧于 0, 则惩罚项趋紧于无穷大, 超球面会过于庞大以至于将整个数据空间包含在内。引入松弛变量允许了理论上都为正常样本的训练数据集中可以存在异常值的污染, 可以放宽算法的约束假设, 从而使算法更加适应实际的应用需求。在完成最小化公式3-1的目标之后, 就可以获得由球心  $c$  和半径  $r$  所定义的超球面。落在超球体之外的数据点  $x_i$ , 即  $x_i$  满足  $\|\phi_k(x_i) - c\|_{\mathcal{F}_k}^2 > r^2$ , 就会被定义为异常数据点。

### 3.2.3 训练目标

SVDD 模型以经典的超球学习为目标, 建立了给定数据的最小体积超球估计。而我们的目的是学习图的节点表示, 为了实现这一目标, 我们引入了 GNN, 它通过考虑节点属性和关系来学习节点嵌入, 并将节点嵌入保持在最小超球面中。OCGNN 框架就是将图深度学习技术与超球面学习目标联合在一起, 用半监督学习方法学习图节点表征的一种通用方法。

在计算节点特征嵌入  $\mathbf{Z} \in \mathcal{R}^{N \times F}$  时, GNN 考虑节点属性  $\mathbf{F}^v \in \mathcal{R}^{N \times D}$  以及图邻接矩阵  $\mathbf{A} \in \mathcal{R}^{N \times N}$ (一些 GNN 模型使用节点邻居信息聚合来取代对于  $\mathbf{A}$  的矩阵运算)。因此, 我们使用  $g(\mathbf{F}^v, \mathbf{A}; \mathcal{W})$  来表示图神经网络, 其中  $\mathcal{W} = \{\mathbf{W}^{(1)}, \dots, \mathbf{W}^{(L)}\}$  代表 GNN 的网络参数,  $L \in \mathcal{N}$  是 GNN 网络的层数。对于第  $l$  层来说, GNN 的前向传播规则可以总结为:

$$\mathbf{H}^{(l+1)} = g\left(\mathbf{H}^{(l)}, \mathbf{A}; \mathbf{W}^{(l)}\right), \quad (3-2)$$

其中  $\mathbf{H}^{(l)}$  是第  $l$  层 GNN 网络的输入,  $\mathbf{H}^{(l+1)}$  是该层网络的输出。图上的节点属性  $\mathbf{F}^v$  是第一层 GNN 的输入, 即  $\mathbf{H}^{(0)} = \mathbf{F}^v$ 。节点特征矩阵  $\mathbf{Z}$  是 GNN 最终的输出, 即



17010226

$\mathbf{Z} = \mathbf{H}^{(L)}$ 。由于强大的 GNN 节点嵌入，它有助于改善下游任务的性能，例如节点级和图级分类以及关系推理。

OCGNN 的目的是联合学习网络参数  $\mathcal{W}$  以及最小化以半径  $r \in \mathcal{R}^+$  和球心  $c \in \mathcal{R}^F$  为特征的数据描述超球体的体积。给定由  $(\mathbf{F}^v, \mathbf{A})$  定义的图以及  $K$  个有标签供训练的节点  $\mathcal{V}_{tr} \subseteq \mathcal{V}, \forall K = |\{i : v_i \in \mathcal{V}_{tr}\}|$ ，OCGNN 的训练目标设定为：

$$\mathcal{L}(r, \mathcal{W}) = \frac{1}{\beta K} \sum_{v_i \in \mathcal{V}_{tr}} [\|g(\mathbf{F}^v, \mathbf{A}; \mathcal{W})_{v_i} - c\|^2 - r^2]^+ + r^2 + \frac{\lambda}{2} \sum_{l=1}^L \|\mathcal{W}^{(l)}\|^2 \quad (3-3)$$

在前向传播中，OCGNN 接收图中的所有节点和边信息，然后输出节点嵌入矩阵  $\mathbf{Z} = g(\mathbf{F}^v, \mathbf{A}; \mathcal{W}), \mathbf{Z} \in \mathcal{R}^{N \times F}$ 。值得注意的是，只有  $K$  个节点的特征嵌入  $\{\mathbf{Z}_{v_i}, v_i \in \mathcal{V}_{tr}\}$  在网络损失函数的训练中被用到。

公式3-3的第一项是对位于超球面之外的节点嵌入的惩罚，即如果嵌入向量与中心  $\mathbf{c}$  之间的距离大于半径  $r$ ，则在损失函数中给予惩罚。超参数  $\beta \in (0, 1]$  控制球体体积和惩罚之间的权衡（我们将在后面讨论）。与经典的 SVDD 一样，第二项，最小化  $r^2$ ，是最小化球体的体积。最后一项是 OCGNN 网络参数  $\mathcal{W}$  上的权重衰减正则化器，超参数  $\lambda > 0$ 。

OCGNN 的训练目标令网络学习如何将节点特征映射到靠近球体中心  $\mathbf{c}$ 。由于训练节点都是正常的数据点，OCGNN 将提取给定节点的公共特征，从而得到正常节点的描述边界，用这个正常节点的描述边界来检测异常节点。

对于给定图中的节点  $v_i$ ，其异常分数  $S(v_i)$  可以通过节点嵌入相对于球体的位置来定义：

$$S(v_i) = \|g(\mathbf{F}^v, \mathbf{A}; \mathcal{W}^*)_{v_i} - c\|^2 - r^{*2} \quad (3-4)$$

如果  $S(v_i) > 0$ ，则证明节点  $v_i$  被映射到超球面之外，即它是异常的；反之如果  $S(v_i) < 0$ ，它就是一个正常的节点。注意，在设定超球面的中心  $\mathbf{c} = \mathbf{0}$  的情况下，只需要网络参数  $\mathcal{W}^*$  和学习的半径  $r^*$  这两者即可以表征一个经过训练的 OCGNN 模型。OCGNN 的存储复杂度非常低，因为模型预测不需要存储所有的数据，只需要使用数据训练网络并使用该网络进行推理。

为了清晰地证明在训练过程中  $\beta$  的作用，我们定义：

$$\begin{aligned} \xi &= \frac{\lambda}{2} \sum_{l=1}^L \|\mathcal{W}^{(l)}\|^2 \\ d_{v_i} &= \|g(\mathbf{F}^v, \mathbf{A}; \mathcal{W}) - c\|_{v_i}^2 \end{aligned} \quad (3-5)$$

其中  $i = 1, \dots, K$ 。假设有一个节点子集  $\mathcal{V}_o \subset \mathcal{V}_{tr}$  的节点被 OCGNN 映射到球体之外，由  $K_o = |\{i : d_{v_i} > r^2, v_i \in \mathcal{V}_{tr}\}|$  节点组成。因此，关于  $r$  的 OCGNN 优化问



17010226

题可以写成：

$$\begin{aligned} & \underset{r}{\operatorname{argmin}} \quad r^2 + \frac{1}{\beta K} \sum_{v_i \in V_o} (d_{v_i} - r^2) + \xi \\ &= \underset{r}{\operatorname{argmin}} \quad \left(1 - \frac{K_o}{\beta K}\right) r^2 + \frac{1}{\beta K} \sum_{v_i \in V_o} d_{v_i} + \xi \\ &\Rightarrow \underset{r}{\operatorname{argmin}} \quad \left(1 - \frac{K_o}{\beta K}\right) r^2 \end{aligned} \quad (3-6)$$

由于公式3-6的第二项和第三项是非负的，所以只要  $K_o \leq \beta K$ ，半径  $r$  就会继续优化被减小。也就是说，只有当  $\frac{K_o}{K} \leq \beta$  时，该式才满足最优值，这意味着  $\beta$  是异常训练节点分数的上界。最优半径  $r^*$  由满足不等式的最大  $K_o$  确定，因为如果不满足不等式，模型就不能继续最小化  $r$ 。所以优化后的结果一定满足  $\frac{K_o+1}{K} > \beta$ 。 $\beta$  是映射到超球体之外样本的下限， $\beta$  允许将一些节点映射到球体之外，否则，为了包围所有训练节点，半径  $r$  将非常大，以至于 OCGNN 无法检测到异常值。

### 3.2.4 OCGNN 的各种范式

我们的 OCGNN 是一个基于 GNN 的图异常检测框架，因此 OCGNN 可以由任何合适的 GNN 层构成，例如图卷积网络 (GCN)<sup>[73]</sup>、图注意网络 (GAT)<sup>[75]</sup> 和 SAGE<sup>[77]</sup>。在本节中，我们以这些网络为例来说明 OCGNN 如何学习节点表示。如果考虑其他形式的 GNN 层，人们只需要用其他邻居节点聚合函数替换公式3-7。GCN 是图学习模型的最佳范例之一。我们在实验中建立了一个多层的 GCN 模型，它将公式3-2扩展如下：

$$g\left(\mathbf{H}^{(l)}, \mathbf{A}; \mathbf{W}^{(l)}\right) = \sigma\left(\tilde{\mathbf{D}}^{-\frac{1}{2}} \tilde{\mathbf{A}} \tilde{\mathbf{D}}^{-\frac{1}{2}} \mathbf{H}^{(l)} \mathbf{W}^{(l)}\right), \quad (3-7)$$

其中  $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}_N$  是在我们在2.1.3小节中提到的图邻接矩阵，注意，我们在使用邻接矩阵之前，为图上的每个节点添加了由自身向自身的连边，以保证自身的信息可以在层次化的迭代中保留，体现在矩阵计算上，只需要把图邻接矩阵跟同等维度的单位矩阵相加即可。 $\tilde{\mathbf{D}}$  代表了根据公式2-2由  $\tilde{\mathbf{A}}$  计算得到的度矩阵，其中  $\tilde{\mathbf{D}}_{ii} = \sum_j \tilde{\mathbf{A}}_{ij}$ 。 $\tilde{\mathbf{D}}^{-\frac{1}{2}} \tilde{\mathbf{A}} \tilde{\mathbf{D}}^{-\frac{1}{2}}$  是邻接矩阵  $\tilde{\mathbf{A}}$  的对称归一化，顺便一提，这部分较为复杂的矩阵计算仅仅只会进行一次，并且不会体现在模型的训练过程之中，因为我们可以直接在数据的预处理过程中计算  $\hat{\mathbf{A}}$ 。 $\sigma(\cdot)$  是一种非线性激活函数，如 ReLU<sup>[160]</sup>。注意，网络权重矩阵  $\mathbf{W}^{(l)}$  对于图上所有的节点共享。从图频谱理论  $y$ <sup>[72]</sup> 的角度来讲，GCN 通过聚合自身的特征  $\mathbf{F}_i^v$  和一阶邻居的特征  $\mathbf{F}_j^v$  来获取图结信息，通过堆叠多个 GCN 层，模型就可以捕获  $k$  阶邻域的信息，从而学习得到节点嵌入  $\mathbf{Z}_i$ ，其中， $j \in \mathcal{N}(v_i)$ 。



17010226

SAGE<sup>[77]</sup>模型的英文全名是 Graph SAmpLe and aggreGatE，意思是图上的采样和聚合。SAGE 是当今最简单的一种图模型，如果要学习节点的特征表示  $h_i$ ，它的做法仅仅是将节点  $v_i$  的邻居信息进行一下简单聚合，再加一个多层感知机的映射。一个典型的 SAGE 层为：

$$\begin{aligned}\mathbf{h}_i^l &\leftarrow \sigma \cdot \mathcal{W}^l \cdot \left( \mathbf{h}_i^{l-1} || \mathbf{h}_{\mathcal{N}^{(k)}(v_i)}^l \right), \\ \mathbf{h}_{\mathcal{N}^{(k)}(v_i)}^l &\leftarrow \text{AGGREGATE} \left( \left\{ \mathbf{h}_j^{l-1}, \forall v_j \in \mathcal{N}^{(k)}(v_i) \right\} \right),\end{aligned}\quad (3-8)$$

其中，参数矩阵  $\mathcal{W}^l$  和聚合函数 AGGREGATE 是可训练的， $\sigma$  是神经网络中的非线性激活函数， $\mathcal{N}^{(k)}(v_i)$  代表节点  $v_i$  的  $k$  阶邻居集合。聚合函数 AGGREGATE 可以有很多选择，比如平均、取最大、按度数归一化等。

图注意模块 (GAT)<sup>[75]</sup>是一种基于自注意力机制的节点嵌入方法。与各项同性的 GCN 网络不同，GCN 假设节点的邻居节点对于该节点的影响力（贡献）等同，而 GAT 假设节点和节点之间的影响力不同。GAT 认为节点  $v_i$  和  $v_j$  之间的影响力  $\alpha_{ij}$  由节点特征之间的注意力机制决定，即该影响力函数是一个可训练映射  $a : \mathcal{R}^N \times \mathcal{R}^N \rightarrow \mathcal{R}$  来描述，它根据节点对  $v_i, v_j$  的特征计算标准化影响力系数  $\alpha_{ij}$ ：

$$\alpha_{ij} = \frac{\exp(a(\mathbf{h}_i, \mathbf{h}_j))}{\sum_{k \in \mathcal{N}(v_i)} \exp(a(\mathbf{h}_i, \mathbf{h}_k))} \quad (3-9)$$

公式中对于每个节点  $v_i$  来说，它仅仅只能跟其邻居节点  $v_j \in \mathcal{N}(v_i)$  的特征进行注意力计算，由此机制即可以将图的拓扑信息加入到网络中。 $\frac{\exp(\cdot)}{\sum \exp(\cdot)}$  是归一化函数，以便该网络可以在不同的节点中进行比较。

最终，GAT 的节点特征计算形式为：

$$\mathbf{h}_i = \sigma \left( \sum_{v_j \in \mathcal{N}(v_i)} \alpha_{ij} \mathbf{W} \mathbf{h}_j \right) \quad (3-10)$$

对于动态图的应用场景来说，我们选用动态图注意力网络 TGAT<sup>[22]</sup>模型作为 OCGNN 在动态图领域中的核心骨架。TGAT 网络是3.2.4小节中所述的图注意力网络 GAT 在动态图下的直接拓展，本章节仅仅使用原理性的公式介绍 TGAT 的功能，详细的理论和计算分析请见第四章的4.2小节。首先，动态图注意力网络将公式3-9和3-10中的节点特征  $\mathbf{h}$  统一加入了时间特征向量为静态模型 GAT 加入了时间上的建模能力，同时，TGAT 还考虑了动态图上的边特征，其形式如下：

$$\alpha_{ij} = \frac{\exp \left( a \left( \mathbf{h}_i, \left( \mathbf{h}_j || \mathbf{f}_{ij}^e || \phi(t - t_j) \right) \right) \right)}{\sum_{k \in \mathcal{N}(v_i)} \exp \left( a \left( \mathbf{h}_i, \left( \mathbf{h}_k || \mathbf{f}_{ik}^e || \phi(t - t_k) \right) \right) \right)} \quad (3-11)$$



17010226

其中,  $\mathbf{f}_{ij}^e$  代表节点  $v_i$  和  $v_j$  在  $t_j$  时刻的交互边特征,  $\phi(t - t_j)$  代表时间戳的映射函数, 它是动态的 TGAT 区别于静态图 GAT 的关键, 该映射函数的形式为:

$$\phi(t) = \cos(\mathbf{w}t + \mathbf{b}) \quad (3-12)$$

其中,  $\mathbf{w}$  和  $\mathbf{b}$  是可学习的网络参数。

在聚合阶段, TGAT 的节点特征计算形式跟 GAT 的公式3-10相同:

$$\mathbf{h}_i = \sigma \left( \sum_{v_j \in N(v_i)} \alpha_{ij} \mathbf{W} \mathbf{h}_j \right) \quad (3-13)$$

### 3.2.5 模型优化过程

---

#### 算法 3-1 训练 OCGNN 模型

---

**Input:** 属性图  $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{F}^v)$ , 正常节点集  $\mathcal{V}_{tr}$ , 松弛参数  $\beta \in (0, 1]$ , 权重惩罚因子

$\lambda > 0$

**Output:** 网络权重  $\mathbf{W}$ , 球心  $c \in \mathcal{R}^F$  和半径  $r \in \mathcal{R}^+$

使用 Glorot 平均法则<sup>[161]</sup>初始化网络权重  $\mathbf{W}$

初始化  $r = 0$ ,  $c = \frac{1}{K} \sum_{v_i \in \mathcal{V}_{tr}} g(\mathbf{F}^v, \mathbf{A}; \mathbf{W})_{v_i}$

**while** 迭代轮次小于规定轮次 **do**

$\mathbf{d}_{\mathcal{V}_{tr}} = \|g(\mathbf{X}, \mathbf{A}; \mathbf{W})_{\mathcal{V}_{tr}} - c\|^2$ ,  $\mathbf{d}_{\mathcal{V}_{tr}} \in \mathcal{R}^K$

$\mathcal{L} = \frac{1}{\beta K} \sum_{v_i \in \mathcal{V}_{tr}} [\mathbf{d}_{v_i} - r^2]^+ + r^2 + \frac{\lambda}{2} \sum_{l=1}^L \|\mathbf{W}^{(l)}\|^2$

$\mathbf{W} \leftarrow \mathbf{W} + \nabla_{\mathbf{W}} (\mathcal{L})$

**if** 每训练  $\phi$  轮 **then**

使用  $\mathbf{d}_{\mathcal{V}_{tr}}$  的  $(1 - \beta) \times 100\%$  百分位数来更新  $r$

**end if**

**end while**

**return**  $\mathbf{W}$ ,  $c$  and  $r$

---

算法3-1总结了 OCGNN 的优化过程。经过训练的 OCGNN 模型可以用三个参数来描述: 权重矩阵  $\mathbf{W}$ 、半径  $r$  和数据中心  $\mathbf{c}$ 。与其他神经网络模型一样, 我们使用随机梯度下降法通过反向传播 (BP) 优化带有 OCGNN 目标的 GNN 模型的参数  $\mathbf{W}$ 。由于半径  $r$  不是 OCGNN 网络的内部参数, BP 算法无法一起优化  $r$  和  $\mathbf{W}$ 。因为, 我们在培训阶段交替更新  $r$  和  $\mathbf{W}$ 。首先, 我们固定  $r$  的值, 训练  $\mathbf{W}$ , 训练轮次为  $\phi$  次。 $\mathbf{W}$  每被训练  $\phi$  轮次,  $r$  可以通过简单的线性百分位数搜索来解出。也就是说, 对于训练节点集  $\mathcal{V}_{tr}$ ,  $|\mathcal{V}_{tr}| = K$ , 我们可以获得每个节点嵌入到球



17010226

	数据集	节点数	边数	特征维度	数据集划分
静态图	Cora	2708	5429	1433	490/246/410
	Citeseer	3327	4732	3703	420/210/352
	Pubmed	19717	44338	500	4725/2364/3936
动态图	Reddit	10984	672447	172	70%/15%/15%
	Wikipedia	9227	157474	172	70%/15%/15%

表 3-2 总结我们实验中使用的数据集

Table 3-2 Summary of the datasets used in our experiments

心距离的集合  $\mathbf{d}_{\mathcal{V}_r} \in \mathcal{R}^K$ 。之后，我们可以将  $\mathbf{d}_{\mathcal{V}_r}$  从小到大进行排序，半径  $r$  可以由  $\mathbf{d}_{\mathcal{V}_r}$  的  $(1 - \beta) \times 100\%$  百分位数计算得出。根据经验和下一节中的实际试验结果，OCGNN 对这个参数  $\phi$  不敏感。在训练阶段， $c \in \mathcal{R}^D$  被固定为  $c_0$ ，这是通过初始前向传播从嵌入的训练节点的平均值计算出来的。要映射目标数据中心周围的节点， $c_0$  是一个不错的选择，因为大多数节点都离节点嵌入空间中的  $c_0$  不太远。

### 3.3 对比实验与分析

在本节中，我们将介绍详细的实验设置和结果，包括数据集、基线模型、网络结构、超参数选择和性能分析。我们模型的源代码是使用 PyTorch<sup>[148]</sup> 和深度图神经网络库 DGL<sup>[43]</sup> 实现的。为了更好的可复现性，模型的相关代码已在 Github 存储库<sup>①</sup> 中发布。此外，在我们的存储库中，我们实现了六个网络嵌入模块、四个异常检测模块和四个图神经网络模块，以及一个从公开地图节点分类数据集生成图异常检测数据集的函数。所有这些函数都可以通过简单的 python 命令自动运行。

#### 3.3.1 数据集

本文使用了三个静态图数据集和两个动态图数据集来验证 OCGNN 框架在不同场景的异常检测性能，表格 3-2 总结了这五个数据集的特征。

静态图数据集中，Cora、Citeseer 和 Pubmed<sup>[162]</sup> 是之前研究中公开获得并广泛使用的引文网络数据集。科学出版物及其引用关系分别表示为图表上的节点和边。表格 3-2 中的静态图中只存在节点特征，而不存在边特征。每个出版物的节点特征由从字典计算出的稀疏词袋<sup>[163]</sup> (bag-of-words) 特征向量描述。我们在本文中使用的节点异常检测数据集是由这三个普通节点分类数据集生成的，其中一个类是正

① <https://github.com/WangXuhongCN/OCGNN>



17010226

常的，其余的是异常的。Cora、Citeseer 和 Pubmed 数据集中的正常类分别是“神经网络”、“IR”和“2 型糖尿病”。三个静态图数据集采用节点数量划分的方法分别分割训练集、验证集和测试集。训练集中的所有节点都属于正常类，而在验证集和测试集中，一半节点是正常的，另一半是从异常类中随机抽样的。此外，我们的 Github 存储库提供了关于这三个 GAD 数据集的更多细节。

Wikipedia 和 Reddit 数据集是两个动态图数据集<sup>①</sup>[21]。Wikipedia，中文名可译为“维基百科”，Wikipedia 的节点代表用户和维基百科页面。节点之间的链接表示百科页面被用户编辑，该数据集是一个二部动态图（二部图的定义详见第二章中的定义2.7），在一个月的时间跨度内，由大约 9300 个节点和 16 万个动态边组成。在该数据集中，大概 1% 的用户在使用过程中被禁止发布，即视为异常用户。Reddit 是一个国外使用最广泛的综合类论坛，该数据集也是一个二部动态图，收集了一个月的用户交互数据，其中包含近 11000 个节点和 70 万条动态边。Reddit 数据集中的交互是指用户通过帖子与 Reddit 论坛上的某个话题进行交互。在该数据集中，0.5% 的用户在使用过程中被禁止在某个话题下发布帖子，对于这部分用户我们视之为异常。在这两个数据集中，用户的编辑由一段文本代表，并转换为 172 维的 LIWC 类别<sup>[164]</sup>（共 172 维）的边特征向量。根据交互时间戳，数据集被拆分为 70%/15%/15%，分别作为训练集/验证集/测试集。

对于本文使用的所有方法，我们使用验证集调整超参数，然后在测试集中评估性能。

### 3.3.2 基线方法

据我们所知，我们的模型是第一个将超球面学习目标应用于属性网络异常检测任务的端到端方法。由于超球学习是最重要的异常检测方法之一，它在图数据中的应用是一种必然的发展。因此，两阶段方法被广泛用于连接图嵌入方法和异常检测方法。在第一阶段，图嵌入学习并将图的结构信息映射到一个固定长度的嵌入向量中。第二阶段，利用嵌入向量和节点特征向量训练欧氏异常检测方法。

我们选择了三种流行的图嵌入方法（DeepWalk<sup>[64]</sup>、Struc2Vec<sup>[158]</sup>和 SDNE<sup>[67]</sup>）来学习 128 维的嵌入向量作为第一阶段的方法。DeepWalk 对图上的每个节点进行随机游走采样，对每个节点形成多个随机游走序列，该方法认为节点随机游走序列代表了该节点的结构信息，可以采用神经网络模型学习从中图嵌入向量。Struc2Vec 度量不同尺度下的局部图相似性，构造多层图来生成节点的结构上下文，Struc2Vec 被认为是保持图结构的最佳嵌入方法之一。SDNE 通过一个深度神经自动编码器

① 下载地址：<http://snap.stanford.edu/jodie>



17010226

模型学习节点表示，以保持节点的 1 阶和 2 阶邻居之间具有近似性。在实验中，我们构建了一个隐藏层大小为 256-128-256 的三层 SDNE。DeepWalk 的“行走长度”、“窗口大小”和“每节点的访问次数”参数分别设置为 5、10 和 50，而对于 Struc2Vec，这三个参数分别为 10、5 和 80。

对于第二阶段的异常检测方法，我们选择了使用四种流行的异常检测方法，并根据验证集中的性能调整异常检测方法的超参数：

1. OCSVM<sup>[92]</sup>是我们本文所描述的 OCGNN 的源方法。它有一个重要的参数  $\beta$ ，被选择为与我们的 OCGNN 模型相同。
2. 孤立森林 (Isolation Forest, IForest)<sup>[90]</sup>是在高维数据集中检测异常数据的有效方法。IF 以递归和随机方式拆分数据特征，并将其存储在树<sup>①</sup>中。离群值被定义为树中路径较短的数据点，因为这意味着这些数据需要从其他多数数据中进行较少的拆分，即很容易与正常数据进行区分。
3. PCA<sup>[93]</sup>和自动编码器这两种常用的基于重建的方法假设异常是不可压缩的，因此无法从低维投影中有效重建。
4. 先使用 SMOTE<sup>[155]</sup>方法对异常节点的图嵌入或者特征值进行上采样，然后使用最常见的 SVM 分类器进行训练。

除了两阶段的图异常检测方法，我们还额外寻找了三种最先进的直接的图异常检测方法，分别是 Dominant (DOM)<sup>[113]</sup>、图自动编码器 GAE<sup>[123]</sup>、以及致力于修正图上数据不平衡的工作 GraphSMOTE<sup>[19]</sup>。DOM 是最近提出的一种基于属性图的最先进的半监督图异常检测方法。我们使用作者推荐的超参数复现了 DOM，并使用了早停 (Early Stop)<sup>[165]</sup>技术。DOM 和 GAE 模型都选择了 GCN 层来构造编码器部分，GAE 和 DOM 的主要区别在于解码器结构和训练目标。GAE 提出了一种内积解码器来预测图上缺失的链接，这相当于重建节点之间的隐藏连接关系。DOM 设计了一个额外的基于 GCN 的解码器，旨在重建节点属性，因此 DOM 有两个不同的解码器，分别用于重建特征和链接，以及它们之间的权衡控制参数（按照作者的建议设置为 0.5）。而 GraphSMOTE 将图上的异常检测问题通过基于图的数据扩充转化为了有监督的节点分类问题，但这种方法的劣势可能在于图数据的维度较高，仅仅只凭少许的异常数据很可能无法生成新的合理异常节点样本以及连接关系。我们的 OCGNN 模型有且仅有一个基于 GCN 的图特征编码器，而 DOM、GAE 和 GraphSMOTE 除了具有一个图编码器之外，还有一个及以上的 GCN 图解码器，毫无疑问这些对比模型的参数更多，会给算法的应用带来额外的负担。

此外，在动态图数据集中，我们选择了上文提到过的四种静态图方法——

① 树是一种计算机程序常用的数据结构



17010226

GAE<sup>[123]</sup>、GAT<sup>[75]</sup>、SAGE<sup>[77]</sup>和 VGAE<sup>[123]</sup>，和四个动态图算法——CTDNE<sup>[126]</sup>、DyRep<sup>[26]</sup>、JODIE<sup>[21]</sup>和 TGAT<sup>[22]</sup>，作为基线来验证 OCGNN 在动态图领域的有效性。除了 VGAE 和 CTDNE 以外，其他六种方法已经在2.4.1.3小节中介绍完毕。VGAE 是 GAE 的贝叶斯变种，两者出自同一篇文献，自动编码器的贝叶斯化由于其学习的是概率分布而不是固定的网络参数，其被视为在时变的数据中可以保持更好的泛化能力。CTDNE 是一种基于序列游走的动态图嵌入方法，该方法通过在静态随机游走图模型 DeepWalk<sup>[64]</sup>中加入时间随机游动来捕获动态图上的空间结构和时间结构特征。

### 3.3.3 实验设置

我们的 OCGNN 是用于图异常检测的 GNN 框架，它可以由任何合适的 GNN 层构成，甚至不区分静态 GNN 还是动态 GNN。为了评估 OCGNN 使用不同骨架模型的不同性能，我们使用了三种流行的 GNN 层范例：GCN<sup>[73]</sup>、GAT<sup>[75]</sup>和 SAGE<sup>[77]</sup>来处理静态图上的异常检测问题，用 TGAT<sup>[22]</sup>模型来处理动态图上的异常检测问题，四者模型网络层数和宽度几乎相同。为了更好地展示 OCGNN 框架的真实性能，我们在所有数据集和 OCGNN 范例中使用相同的超参数进行训练。惩罚参数  $\beta$  设置为 0.1。OCGNN 模型由 Glorot 均匀权重初始化，并由 AdamW<sup>[166]</sup>随机梯度下降优化器优化，学习率为 0.001。在训练期间，我们使用  $\lambda = 0.0005$  应用权重衰减正则化。我们使用早停策略对每个 OCGNN 模型进行训练，早停策略的观察对象为 OCGNN 的损失函数（公式3-3）和验证集上的 AUC 分数。模型最多训练 5000 个批次，如果经历 100 个批次模型没有更好，那么即停止训练。对于 Cora 和 Citeseer 数据集，我们应用了一个隐含层大小为 64-64-32 的三层 OCGNN。对于 Pubmed 数据集，我们使用隐藏大小为 128-64 的两层 OCGNN。对于每个单独的 OCGNN 模型，输出层的维度（节点嵌入的维度）总是隐含层大小的一半。在每个 GNN 层之后，应用 50% 概率的 Dropout 层<sup>[167]</sup>和 ReLU<sup>[160]</sup>激活函数。SAGE 的聚合器类型设置为 **pooling**，GAT 层的注意头设置为 8。请注意，本研究中使用的结构和超参数对于我们的应用来说是足够的，尽管它们仍然可以改进以进一步提升性能。

### 3.3.4 结果分析

表格3-3显示了 OCGNN 以及数个基线模型在静态图数据集下其 AUC 指标的实验结果。对于每个数据集和方法，我们用 10 次随机试验的标准差和平均值以显示其 AUC 指标，单位为%。请注意，最好的结果用加粗字体表示。OC-GAT 和 OC-SAGE 获得了最佳排名，并在三组数据中表现一致。



17010226

## 上海交通大学博士学位论文

Available Data	Methods	Cora	Citeseer	Pubmed
X	IForest	53.09± 0.03	46.33± 0.03	65.57± 0.02
	OCSVM	54.35± 0.02	57.05± 0.03	45.50± 0.01
	PCA	62.17± 0.01	58.10± 0.03	71.06± 0.01
	AE	62.17± 0.01	58.11± 0.03	71.05± 0.01
	SMOTE	42.11± 0.02	38.05± 0.03	61.80± 0.01
DeepWalk	IForest	57.87± 0.02	51.00± 0.03	60.73± 0.01
	OCSVM	52.10± 0.03	43.13± 0.02	60.22± 0.01
	PCA	55.90± 0.03	46.65± 0.02	61.66± 0.01
	AE	55.91± 0.03	46.42± 0.02	61.56± 0.01
	SMOTE	59.12± 0.03	52.22± 0.02	59.18± 0.01
A	IForest	56.07± 0.02	54.07± 0.04	48.53± 0.00
	OCSVM	55.95± 0.02	55.45± 0.02	48.27± 0.00
	PCA	55.82± 0.02	53.60± 0.03	48.57± 0.00
	AE	55.80± 0.02	53.02± 0.03	48.38± 0.01
	SMOTE	55.40± 0.02	56.12± 0.03	58.13± 0.01
SDNE	IForest	57.87± 0.02	51.00± 0.03	60.73± 0.01
	OCSVM	52.10± 0.03	43.13± 0.02	60.22± 0.01
	PCA	55.90± 0.03	46.65± 0.02	61.66± 0.01
	AE	55.91± 0.03	46.42± 0.02	61.66± 0.01
	SMOTE	56.12± 0.03	48.44± 0.02	59.36± 0.01
DeepWalk+Features	IForest	53.56± 0.04	45.55± 0.06	65.60± 0.02
	OCSVM	51.59± 0.03	42.95± 0.02	60.10± 0.01
	PCA	62.38± 0.02	57.96± 0.03	72.04± 0.01
	AE	62.39± 0.02	57.96± 0.03	71.91± 0.01
	SMOTE	54.99± 0.02	51.90± 0.03	63.05± 0.01
Struc2Vec+Features	IForest	53.92± 0.03	50.72± 0.05	58.48± 0.01
	OCSVM	55.07± 0.02	56.03± 0.02	48.35± 0.01
	PCA	63.08± 0.01	57.90± 0.01	68.00± 0.01
	AE	62.83± 0.01	57.70± 0.01	68.02± 0.01
	SMOTE	49.29± 0.02	48.76± 0.03	61.91± 0.01
A+X	IForest	55.07± 0.02	50.47± 0.05	54.90± 0.02
	OCSVM	55.85± 0.03	54.97± 0.02	45.47± 0.01
	PCA	61.85± 0.02	58.03± 0.01	66.10± 0.01
	AE	62.30± 0.02	58.37± 0.01	65.97± 0.01
	SMOTE	55.64± 0.02	50.96± 0.03	65.91± 0.01
SDNE+Features	GAE <sup>[123]</sup>	60.15± 0.18	51.80± 0.03	54.27± 0.02
	DOM <sup>[113]</sup>	64.50± 0.25	62.44± 0.15	50.92± 0.04
	GraphSMOTE <sup>[19]</sup>	73.91± 0.25	75.70± 0.15	57.20± 0.04
	<b>OC-GCN</b>	73.25± 0.02	62.81± 0.01	54.53± 0.01
	<b>OC-GAT</b>	<b>88.19± 0.02</b>	79.06± 0.03	66.98± 0.01
	<b>OC-SAGE</b>	86.97± 0.04	<b>85.62± 0.01</b>	<b>74.72± 0.03</b>

表 3-3 静态图实验数值结果

Table 3-3 The experience results in static graph



17010226

根据实验结果，我们的 OCGNN 模型在所有数据集中都取得了最好的结果。在 OCGNN 的三种变体中，我们认为 OC-SAGE 是最强大、最稳定的一种，尤其是在 Pubmed 这样的大型数据集中，因为 OC-SAGE 由于其简单的邻居聚合策略而具有很强的鲁棒性。OC-GAT 在小型 Cora 数据集中表现良好，但在 Pubmed 中失去了优势。这两个模型具有相同的隐藏层维度，但 OC-GAT 具有更复杂的注意机制，需要更多的隐藏神经元来捕捉更大数据集的共同模式。OC-GCN 和 DOM 之间没有明显的性能差距，但 OC-GCN 比 DOM 具有更少的净参数和更低的时间复杂度，我们将在下一节中解释这一点。

在四种异常检测方法中，只有自动编码器（AE）和主成分分析（PCA）两种方法在原始特征上取得了竞争性的结果。这是因为 PCA 和 AE 更擅长处理高维原始特征，而不是重建稀疏的邻接向量。IForest 和 OCSVM 之所以失败，是因为它们不擅长处理高维数据。而至于数据上采样方法 SMOTE，由于三个数据集的节点特征中存在大量的离散型特征，这种特征在特征空间中是不连续的，因此使用 SMOTE 插值方法可能会生成分布外的样本，并大幅度降低算法的准确度。而当我们仅仅使用连续的图嵌入向量作为节点的特征时，SMOTE 方法相比其他异常检测方法展现了一定的竞争性。虽然像 DeepWalk 这样的方法可以提供低维嵌入，但这种嵌入存在与 GAD 任务无关的问题。比较三种图嵌入方法，我们可以得出结论，不同的网络嵌入方法对性能几乎没有影响，这进一步证明了两阶段方法在 GAD 任务中的经验失败。

OCGNN 优于其他两阶段方法，在 Cora 和 Citeseer 数据集中，与 OCSVM 相比，AUC 提高了 30% 以上。此外，我们可以得出结论，DOM（重建节点属性和邻接向量）和 GAE（仅重建邻接向量）没有显著的性能差异，因为 DOM 中两个任务目标的损失值可能不在同一数量级上，这可能会导致训练期间的不稳定性导致性能下降。GraphSMOTE 方法从不平衡图数据扩充的角度为图上的异常检测提供了一种新的先进思路，然而，在整体上来说该类方法在性能上比不过我们的 OCGNN 方法。

在动态图方面，我们也选择了当前比较先进的静态和动态图学习方法作为我们的基线方法，这些方法的主要原理都在 2.4.1.3 和 2.4.3.2 小节中描述过。根据表格 3-4 的性能展示，OC-TGAT 模型在动态图的异常检测方面有巨大的优势。而且显然，几乎所有基于动态图的方法都优于静态图方法，因为静态图方法没有办法捕获图的动态特性，而动态图数据集中，很多异常节点在刚开始是正常的，存在由正常慢慢演化为异常的特性。而在动态图方法中，由于 CTDNE、DyRep、JODIE 以及 TGAT 方法本身不是为了解决异常检测的问题，所以其学习到的节点特征明



17010226

	静态图方法					动态图方法			
	GAE	GAT	SAGE	VGAE	CTDNE	DyRep	JODIE	TGAT	OC-TGAT
Reddit	58.39	64.52	61.24	57.98	59.43	62.91	59.90	68.63	<b>89.38</b>
Reddit Std.	0.03	0.02	0.01	0.01	0.03	0.01	0.03	0.01	0.02
Wikipedia	74.85	82.34	82.42	73.67	75.89	84.59	83.17	83.69	<b>83.88</b>
Wikipedia Std.	0.03	0.02	0.01	0.02	0.03	0.02	0.04	0.01	0.02

表 3-4 动态图实验数值结果

Table 3-4 The experience results in temporal graph

显不如专职与异常检测任务的 OC-TGAT 好，因此损失了部分性能。

### 3.3.5 可视化分析

我们对 OCGNN 框架学习到的节点嵌入进行了 T-SNE<sup>[168]</sup>可视化分析，以便更好地理解其数据特征在欧氏空间中的分布。在这里，我们只关注 Cora 数据集的分析，并使用 GAT 算法来构成 OCGNN。

图3-2是 OCGNN 模型在 Cora 数据集中节点嵌入的 T-SNE 可视化。顶部和右侧的 KDE 曲线显示了每个维度的概率分布。蓝色和橙色点分别表示正常节点和异常节点。图3-2a是节点原始特征的可视化，图3-2b是从随机初始化的 OCGNN 模型中学习到节点嵌入的可视化分析。图3-2a和3-2b中的节点嵌入可视化表明，我们无法通过原始节点特征或初始化的 OCGNN 检测异常节点。如图3-2c所示，经过 500 次训练后，正常节点和异常节点似乎有分离的趋势。在经过完整的训练过程后，图3-2d中的 KDE 曲线证明，正常节点的概率密度高，异常节点的概率密度低，反之亦然。这意味着，如果模型收敛，大多数正常节点和异常节点可以在嵌入空间中轻松划分。我们可以得出结论，OCGNN 确实能够学习正常节点的特征，并且能够以半监督的训练方式区分异常节点。

### 3.3.6 参数敏感性

为了验证网络结构对性能的影响，我们将隐藏层的数量从 1 调整到 7，并将每个隐藏层中隐藏神经元的数量分别从 16 调整到 128。不同网络结构的结果如图3-3所示，该图展示的是 Cora 数据集中不同 OCGNN 模型的平均 AUC (%) 热力图。横轴表示隐藏层的尺寸，纵轴表示网络中的层数。颜色越浅，模型的 AUC 性能越高。另外，我们也对动态图的 OCGNN 模型进行了参数敏感性分析，在 OC-TGAT

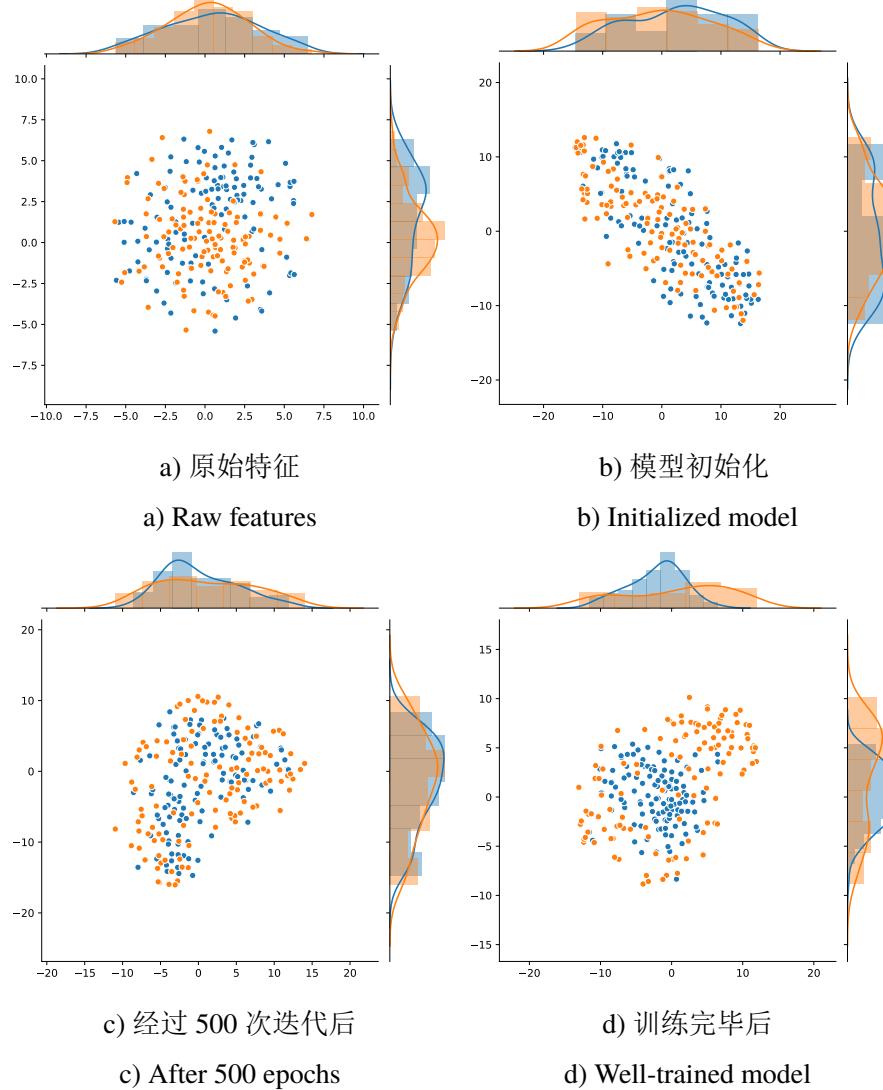


图 3-2 节点特征在嵌入空间的可视化

Figure 3-2 T-SNE visualization of the node embeddings

中，我们将隐藏层的数量从 1 调整到 5，并将嵌入维度分别调整为 16 到 128。得到了跟静态图模型类似的热力图3-4。

在静态图场景中，当网络规模太大（右上）或太小（左下）时，OCGNN 的性能会降低。使用 2 到 4 层的模型和 16 或 32 维的隐含层宽度通常可以带来最好的结果。

而在动态图场景中，OCGNN 可以在给定的跨度内实现有效且稳定的性能，尤其是当层数为 2 到 4，维度为 32 到 64 时。

通过深入观察，我们可以发现，当网络参数容量非常小时，OCGNN 的性能会



17010226

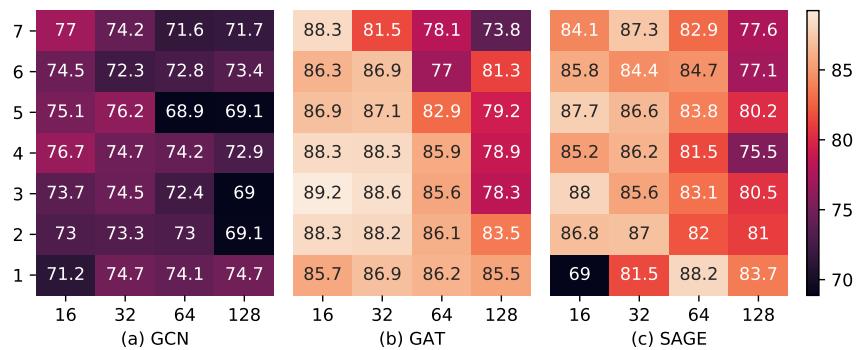


图 3-3 OCGNN 架构在静态图场景中的参数敏感性分析

Figure 3-3 Parameter sensitivity analysis of ognn model in static graph

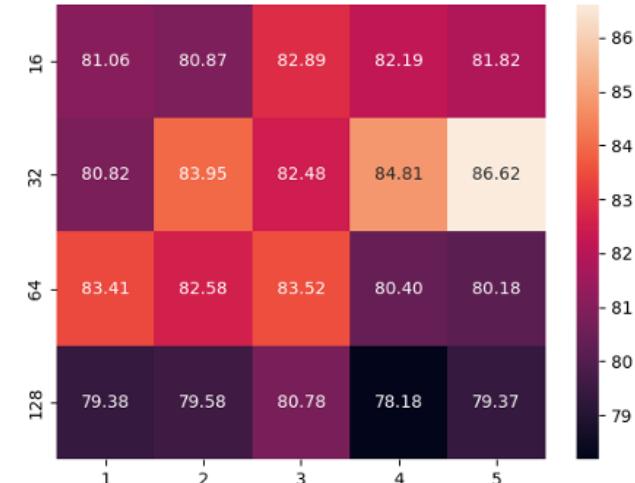


图 3-4 OCGNN 架构在动态图场景中的参数敏感性分析

Figure 3-4 Parameter sensitivity analysis of ognn model in temporal graph

大大降低，因为不足的参数容量无法从正常的训练数据中学习足够的信息。另一方面，当网络太深或太宽时，OCGNN 的效果稍差。深度 GNN 模型通常会遇到过平滑问题<sup>[81]</sup>，因为 GNN 模型逐层迭代地聚合几乎整个图的节点信息，并计算低多样性的节点嵌入。如果网络太宽，维数灾难<sup>[169]</sup>将导致 OCGNN 无法计算损失函数中的距离项，因为数据点的距离在高维空间中接近相同。但是另一方面，只要神经网络的规模足够大，网络的层数或神经元的数量就不会对结果产生显著影响。综合来说，OCGNN 模型的性能对网络结构具有鲁棒性。



17010226

### 3.3.7 运行效率

我们在装有英特尔至强（Xeon）CPU 和英伟达 GeForce GTX 1080Ti GPU 的 Linux 服务器上进行了实验。运行效率实验在 pubmed 数据集上进行，图3-5总结了不同方法的训练和测试运行时以及网络参数的数量。基于 GCN 的模型具有较少的网络参数，但会导致较差的 AUC 性能，如表 3-3所示。DOM 的网络参数是 OC-GCN 的三倍，因为 DOM 是一个三网络模型：一个编码器、一个属性解码器和一个结构解码器，而我们的 OCGNN 架构只有一个编码器。这意味着 DOM 的层数是 OCGNN 模型的三倍，这将增加训练时梯度反向传播的时间。除此之外，DOM 的损失函数是计算高维输入 ( $\mathbf{F}^v$  和  $\mathbf{A}$ ) 与其重建矩阵之间的均方误差，而 OCGNN 的损失是低维嵌入矩阵  $\mathbf{Z}$  和数据中心向量  $\mathbf{c}$  之间的计算。因此，DOM 的时间复杂度和空间复杂度都显著高于我们的 OCGNN 模型。虽然 OC-SAGE 和 OC-GAT 的网络规模比 OC-GCN 大得多，但它们不会导致运行时间的大幅增加，因为这三种 OCGNN 模型的 GNN 层数相同。而 DOM 的层数几乎是 OC-GCN 的三倍，因此比 OC-GCN 慢 15 倍，这是时间复杂度的核心因素。

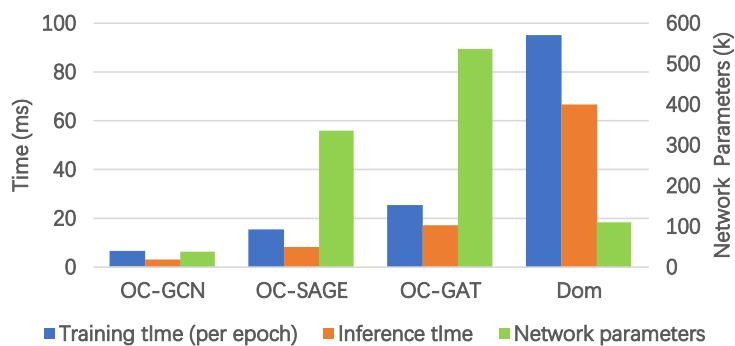


图 3-5 不同方法的训练和推理运行时间以及网络参数量

Figure 3-5 The training and test runtimes and the amount of network parameters for the different approaches

为了给读者提供更丰富的时间效率信息，我们以 Deepwalk+OCSVM 为例，报告了传统异常检测方法的效率。在 Pubmed 数据集中，DeepWalk 预训练的时间开销为 218877 毫秒，OCSVM 的训练和推理时间分别为 18884 毫秒和 3309 毫秒。GNN 比 DeepWalk 快 7000 倍，因为图嵌入方法通常采用更复杂的随机行走操作，而 GNN 使用更轻的邻域采样。此外，OCGNN 比 OCSVM 快 600 倍，因为神经网络方法可以使用 GPU 并行处理大规模数据，而传统的机器学习模型只能串行处理数据。



17010226

### 3.4 本章小节

在本章中，以异常检测为背景，我们解决了图深度学习技术在数据不平衡场景下存在的有偏估计问题，我们提出了基于半监督学习的一类分类图神经网络架构（One-Class GNN，OCGNN）。OCGNN 的目标是通过图神经网络强大的表示能力，将训练节点映射到嵌入空间中的超球面。OCGNN 是一种通用的 GNN 学习框架，因此可以适用于绝大多数不同的 GNN 层，其使用范围甚至可以从静态图泛化到动态图，并且 OCGNN 框架在带来了性能提升的同时，还比最先进的图异常检测方法复杂度更低更易于使用。为了支撑这些结论，我们进行了大量包括性能实验、参数敏感性实验、运行效率实验在内的各项实验，其结果表明，提出的 OCGNN 实现了显著的改进。



17010226

## 第四章 基于时空关系约束映射的动态关系推理

### 4.1 引言

图是一种描述复杂网络的通用数学语言，也是一种表示实体之间关系的常用数据结构，它可以延续网络科学应用的各个领域，如经济网络、通信网络、交通网络、社会网络、交易网络、生物网络等<sup>[12]</sup>。随着图数据的应用越来越广泛，如何对图数据进行建模，并将节点表示为下游任务的低维嵌入向量以及如何表示节点和节点之间的上下文关系已成为研究人员关注的关键问题。图神经网络<sup>[10]</sup>（Graph Neural Network, GNN）有着广泛的应用，成为实现这一目标的一种强大的工具以及很有前途的方法。

然而，大多数现实生活中的图系统是动态的：随着时间的推移，图上的节点和边可能会出现或消失，甚至节点属性也可能会改变。例如，在社交网络中，由于热点事件，用户通常会在短时间内将兴趣转移到其他实体；在经济网络中，欺诈者往往突然实施一系列犯罪，然后在最短的时间内提取非法资金。假设我们采用静态图方法对这些动态网络进行建模，这将更简单、更省时，但我们无法捕捉拓扑结构的演化模式。此外，当我们学习动态网络上的节点表示时，我们需要综合考虑历史事件对当前节点状态的影响。可以想象，动态图的研究难度远高于静态图。

随着静态图深度学习算法的成熟，许多文献<sup>[22,24,26,170]</sup>已经开始提出一些动态GNN算法来克服这些问题，尝试建模第二章中定义2.4所述的动态图，以增强普通GNN方法在动态图中的能力。然而他们的工作<sup>[21-22,24]</sup>大部分只是静态图任务在动态图上的简单延伸，根据历史的图信息，在查询时间戳 $t_0$ 时刻，根据给定的源节点 $v^s$ 来预测下一个目标节点 $v^d$ ，即学习一个映射 $(\mathcal{G}_{t_0}, v^s) \Rightarrow v^d$ 。这些动态图相关文献与其他关于静态关系推理任务<sup>[124,171-172]</sup>的文献的唯一区别在于他们仅仅是将图上的时间信息融入到传统的静态图学习任务中，即旨在使用循环神经网络<sup>[173]</sup>（Recurrent Neural Network, RNN）、注意力网络<sup>[174]</sup>（Attention Network）等等利用时间信息执行如关系推理等传统静态图空间关系学习任务，并不是真的发挥了动态图的全部数据潜力，具体的文献综述详见2.4.3.2小节的相关工作部分。

上述的动态图相关文献所提出的模型适用的学习任务是与时间无关的，那么显而易见的是，机器学习模型就不会有强烈动力来编码时间有关的特征表示。有些文献<sup>[26-32]</sup>注意到了这个问题，并将随机点过程<sup>[134]</sup>和图学习模型相结合，在传统的静态关系推理任务的基础上，都提出了动态图上的“时间戳预测”“任务以证明



17010226

动态图模型在时间建模上的优势：给定查询时间戳  $t_0$  所形成的历史图  $\mathcal{G}_{t_0-}$  和图中的一条边上的源节点  $v^s$  和目标节点  $v^d$ ，组成二元组  $(v^s, v^d)$ ，预测这两个节点下一次发生交互的时间  $t$ 。

然而，时间戳预测任务存在着其固有弊端，它不具备在图的拓扑结构学习上的挑战性。在已知二元组  $(v^s, v^d)$  的情况下预测一个实数的时间戳值在直觉上并不需要利用图上的结构信息。如果我们把时间和空间作为模型可以学习的两个自由度，那么该映射  $(v^s, v^d) \Rightarrow t$  仅仅在时间上有自由度，模型甚至可以仅仅通过统计两个节点之间的历史交互频率来预测他们下一次交互的时间，假如模型可以仅仅只根据两个节点对之间的历史交互来预测时间，自然就会丢失模型在整体图结构建模上的能力。

我们决心改善动态图学习领域中这个关键问题，为动态图社群提供新的方向。我们提出了动态关系推理网络（**TEmporal Relational ReasongIng NEtwork, TER-RINE**）模型，该模型希望预测一系列未来的事件所包含的对象，同时希望模型预测事件所发生的时间。于是我们让模型学习一个更加困难的映射  $\mathcal{G}_{t_0-}, v^s \Rightarrow (v^d, t)$ ，根据历史的图信息  $\mathcal{G}_{t_0-}$ 、源节点  $v^s$  直接预测下一步要交互的目标节点  $v^d$  以及该次交互的时间戳  $t$ 。这意味着模型需要建模一个联合条件概率分布： $p((v^d, t) | (v^s, \mathcal{G}_{t_0-}))$ 。

过去的工作不考虑时间-空间联合建模的原因主要是这种预测模型需要搜索求解的可行解空间太庞大了，首先，模型需要在给定的图中挑选一个节点作为输出的目标节点  $v^d$ ，其次，模型还必须给出相应合理的时间戳  $t$ 。毫无疑问，这是一个双自由度的预测任务，模型不仅要正确预测空间的动态演化，同时也要考虑时间上的动态性，我们称这个任务为“动态关系推理”。与不处理实体交互的时间序列预测以及传统的静态关系推理相比，动态关系推理必须联合考虑由图表征的空间信息和以事件流为特征的时间信号，以便做出更准确的预测。动态关系推理任务已经在2.3小节中的定义2.11中阐述。然而，作为一个有潜力的新型机器学习任务，它没有得到很好的探索，据我们所知，在这项任务上没有使用深度学习的前期工作。为了进一步说明其在应用中的优势，本章中会针对该任务的应用优势给出更加详细的说明。

第一点，动态关系推理任务从本质上讲需要对系统的长期演化、不同实体之间的交互关系、事件之间的时间差进行建模。因此动态关系推理所带来的更多维度的任务可以增加动态图学习模型的泛化和表达能力，如果我们将一个强大的动态图模型进行领域的迁移，就可以借此提升下游的动态图算法的性能，比如知识推理、异常检测等算法，并可以对算法的下游应用如金融、交通、地震或流行病



17010226

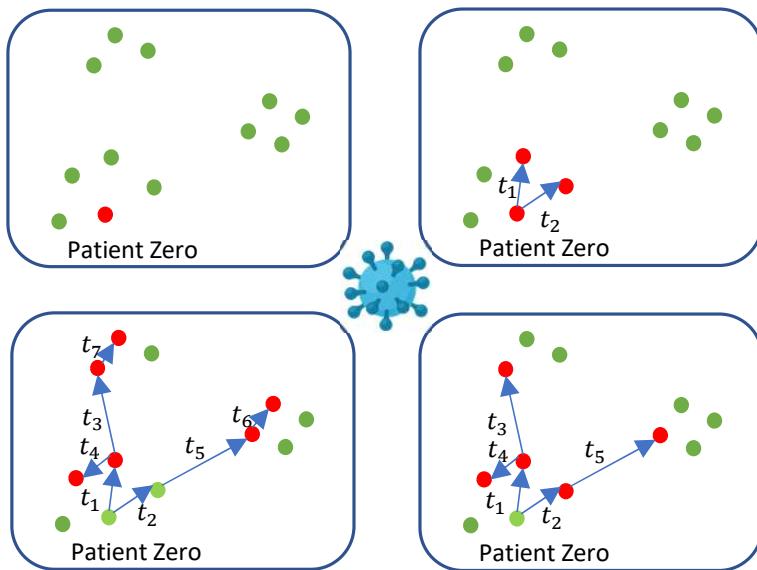


图 4-1 在新冠病毒传播链监控这个场景中，假设我们要预估一位零号病人对其接触者的影响，我们不能不考虑时间的因素。图中每个节点代表一个人，而红色的节点代表感染的个体，蓝色的箭头代表传播链，也即是病人的社交链。如果不考虑时间因素的话，政府很有可能会把已经痊愈的病人的社交也加码纳入风险管理体系中，造成对疫情影响的错误估计。

Figure 4-1 An example of continuous temporal dynamic graph depicting the transmission chain of a virus. Each node represents a person and red nodes represent infected cases. The blue arrows represent the transmission with timestamps. After a period of time, infected cases may heal and become non-infectious.

等数据建模产生积极的影响。

第二点，动态关系推理任务在经济<sup>[50]</sup>、交通<sup>[47]</sup>、医疗保健<sup>[49]</sup>和社会科学<sup>[51]</sup>等等很多方面具有非常大的应用优势，因为他们中的许多场景不仅要预测网络中实体之间的交互，还要预测下一次交互发生的时间，而且更要对系统的长期演化进行建模。典型的例子是 2019 冠状病毒疾病传播，其中，如图4-1所示，如果我们要预测 COVID-19 传播链在本地社群中的演化，我们需要利用个体之间的关系来预测病毒的传播。我们不仅需要预测个体之间的相互作用，还需要预测这些相互作用的时间，因为这取决于病毒是否处于潜伏期。此外，由于一名受感染者可能在潜伏期内也会进一步感染其他人，因此在预测或统计感染者的传播情况时把时间纳入考虑是非常必要的。

为了让模型高效地执行上述的动态关系推理任务，就必须要解决时空联合建模的解空间过于庞大的问题。而压缩可行解空间的关键在于引入某种时空空间的关联假设，通过假设使得一方确定之后，另一方也被唯一确定。而引入时空关联假



17010226

设的约束又带来了新的难点：第一，必须在统一的嵌入空间中表达时空关系，否则无法互相约束。其次，节点在特征嵌入空间中的位置关系必须蕴含时空间双元的语义信息，即不光要反映关系上的远近，还要反映时间上等相对远近关系。

为了解决这两个难题，我们提出了两种不同的时空关系假设，分别是时空三角关系假设以及范数单调性假设。时空三角关系假设旨在将动态图节点之间的时间和空间关系约束在高维空间中的一个闭合三角面内。即对于某个由“源节点-时间-目标节点”三元组组成的交互事件来说，在特征嵌入空间中，源节点向量 + 时间向量  $\approx$  目标节点向量。通过该假设，时间和空间关系就被绑定在一起，在一个统一的空间中进行表示，极大地压缩了模型的可行解空间。第二个假设是范数单调性假设，该假设认为如果事件的时间距离当前时刻越远，该事件的时间在特征嵌入空间对应的时间向量的范数也要越大。而由于第一个三角假设，时间向量越大，代表两个节点在嵌入空间的位置越远，同时也代表两个节点需要更长地时间才有可能发生交互。由此，节点在嵌入空间中的位置就同时包含了时间和空间的双元语义信息。

综上所述，我们的论文做出的贡献如下：我们考虑动态图上的动态关系推理任务，它根据源节点的信息，预测了终节点和时间戳的联合分布。由于解空间膨胀，这比静态的关系推理和时间戳预测任务都要困难得多。我们提出了两个不同的时空关系假设来约束庞大的可行解空间。用白话来说，第一种约束通过时空的耦合确保解空间不过于庞大，第二种约束则要确保这种压缩是合理的且符合时空的语义特征。为了设计出能实现所提时空关系假设的图深度学习模型，我们又提出了正负样本生成、图学习的岭回归技术以及分层推理技术。最后，我们从性能和效率的角度评估了我们的动态关系推理模型，并结合多个最先进的基础模型建立了一个动态关系推理的基准，这可能有助于未来的研究。结果表明，我们的模型在动态关系推理任务中都取得了比其他先进基线更好的性能。

## 4.2 基于时空关系映射的动态关系推理模型

在本节中，我们首先介绍了多头注意力机制，它是模型如何利用节点的邻居信息推导节点的特征嵌入的重要工具。我们介绍了动态图注意层，并展示了如何将时间核方法融合到动态图注意力层中，我们也展示了如何自然地动态图注意力层处理包含边特征的图。最后，在模型的训练阶段，我们提出了空间-时间联合的负采样策略，让我们的动态图神经网络模型学会如何联合预测时间-空间事件。



17010226

#### 4.2.1 多头注意力机制

多头注意力机制主要由两个部分组成，分别是嵌入层和注意力层。嵌入层将一个有序的实体序列  $\{e_i, i = 1, \dots, l\}$  作为输入，这个序列的有序性由位置编码模块来保证。即对于序列中的每个位置  $i$ ，都存在由一个位置映射得到的向量  $\mathbf{pos}_i$  一一对应。对于一个实体序列  $e$  来说，嵌入层通常将实体的特征  $\mathbf{z}_{e_i} \in \mathcal{R}^{l \times d}$  以及位置编码向量  $\mathbf{pos}_i$  进行求和存储作为最终的实体特征，多个实体的特征可以组成一个实体特征的集合，即：

$$\mathbf{Z}_e = [\mathbf{z}_{e_1} + \mathbf{pos}_1, \dots, \mathbf{z}_{e_l} + \mathbf{pos}_l]^T \in \mathcal{R}^{l \times d} \quad (4-1)$$

当然，有些文献也采用向量拼接（即下方公式中的  $\parallel$  操作）的方法代替求和，通过这种方法形成的最终的实体特征与上面公式类似：

$$\mathbf{Z}_e = [\mathbf{z}_{e_1} \parallel \mathbf{pos}_1, \dots, \mathbf{z}_{e_l} \parallel \mathbf{pos}_l]^T \in \mathcal{R}^{l \times (d+d_{\text{pos}})} \quad (4-2)$$

其中  $d_{\text{pos}}$  是位置编码向量的维度。

注意力层负责提取这个最终的实体特征向量的隐藏模式，并生成最终的注意力特征。注意力层采用点积 + 向量缩放机制<sup>[175]</sup>定义这个最终的注意力特征：

$$\text{Attn}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d}}\right)\mathbf{V} \quad (4-3)$$

其中  $\mathbf{Q}$  被称为“查询 (Queries)”， $\mathbf{K}$  和  $\mathbf{V}$  分别被称为“键 (Keys)”和“值 (Values)”。绝大多数注意力机制相关的文献都使用实体特征向量的线性映射作为查询、键和值矩阵，即  $\mathbf{Q} = \mathbf{Z}_e \mathbf{W}_Q$ ， $\mathbf{K} = \mathbf{Z}_e \mathbf{W}_K$  和  $\mathbf{V} = \mathbf{Z}_e \mathbf{W}_V$ ， $\mathbf{W}_Q$ ， $\mathbf{W}_K$  以及  $\mathbf{W}_V$  都是线性映射矩阵。

由于  $\mathbf{Q}$ ， $\mathbf{K}$  和  $\mathbf{V}$  的每一行代表了一个序列中的实体，点积注意力的作用就是寻找这个序列中实体与实体之间的相互关联，如果两个实体向量的点积的值偏大那么代表这两个实体更加相关，反之如果偏小那么可以任务较为不相关。

在实际应用中，注意力模型通常采用多个注意力头来形成多个子空间来学习不同方面的信息并找寻实体之间在不同方面的相关性。为了构建多头注意模块，我们构造了多个注意并将它们连接起来，假设我们需要构建  $k$  个注意头：

$$\begin{aligned} \text{head}_i &= \text{Attn}(\mathbf{Q}_i, \mathbf{K}_i, \mathbf{V}_i), i = 1, \dots, k, \\ \text{MultiHead}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) &= \text{Concat}(\text{head}_1, \dots, \text{head}_k) \mathbf{W}^O \end{aligned} \quad (4-4)$$

模型会分别生成  $k$  个注意力特征向量，并把他们拼接起来再经过一个整体的线性映射，形成最终的特征表示。



17010226

正是由于点积注意力所带来的相关性计算，那么这种点积注意力也可以被应用在图数据的处理中，比如注意力机制可以考虑某个节点与其邻居之间的相关性，以便更好地利用邻居信息以生成节点的特征。在下一小节中，我们介绍了如何将多头注意力机制和时间核函数结合，从而建模动态图数据中的时间相关的特征。

#### 4.2.2 动态图注意力编码

首先，我们回顾一下在2.4小节中定义2.5所定义的时态邻居（Temporal Neighbor）的概念。

**定义 4.1 (时态邻居 (Temporal Neighbor))** 如果在时间戳  $t$  之前，存在一条边  $e_{ij}^\tau$  连接节点  $v_i$  和  $v_j$ ，并且  $\tau \leq t$  那么可以称  $v_j$  是节点  $v_i$  在  $t$  时刻的时态邻居节点，由这些时态邻居节点所组成的集合记为  $N(v_i; t)$ 。 $N(v_i; t) = \{v_j | \exists e_{ij}^\tau \in \mathcal{E}_{t-}\}$

接下来，我们说明如何根据某个节点的时态邻居信息，计算该节点的特征嵌入向量。我们使用了第  $l$  层的动态图注意力神经网络来对某个节点  $i$  的在经过第  $l-1$  层神经网络中邻居特征信息  $\{\tilde{\mathbf{h}}_1^{(l-1)}(t_{i1}), \dots, \tilde{\mathbf{h}}_N^{(l-1)}(t_{iN})\}$  进行聚合，并更新该节点的特征向量  $\tilde{\mathbf{h}}_i^{(l-1)}(t)$ ，按照这种形式执行  $l$  次就可以得到每个节点最终的向量表示。值得注意的是，对于某个节点的特征表示  $\tilde{\mathbf{h}}_i^{(l)}(t)$  是一个时间相关的表示，对于同一个节点来说不同的时刻就会存在不同的特征表示。

由于动态图是一个多重图，节点  $v_i$  和时态邻居  $v_j \in N(v_i; t)$  在多重图中的历史交互发生在多个不同的时间戳  $t_{ij}, \forall t_{ij} < t$  上。所以在我们的实现中，我们使用了多重边的所有交互信息。因此，第  $l$  层神经网络的输入由  $v_i$  在  $l-1$  层的特征表示  $\tilde{\mathbf{h}}_i^{(l-1)}(t)$ ，当前的时间戳  $t$ ， $v_i$  的时态邻居  $N(v_i; t)$  的特征表示  $\{\tilde{\mathbf{h}}_1^{(l-1)}(t_{i1}), \dots, \tilde{\mathbf{h}}_N^{(l-1)}(t_{iN})\}$  以及从  $v_i$  出发的所有边上的特征组成  $\mathbf{f}_{i1}^e(t_{i1}), \dots, \mathbf{f}_{iN}^e(t_{iN})$ 。

另一方面，我们可以用  $\{t-t_1, \dots, t-t_N\}$  来表示交互的时间，这是由于  $|t_i - t_j| = |(t - t_i) - (t - t_j)|$ ，即我们只需要关心时间跨度，而不是绝对的时间值。

最终，我们可以获得节点的时间相关特征矩阵：

$$\mathbf{Z}(t) = \begin{bmatrix} \tilde{\mathbf{h}}_1^{(l)}(t) \\ \vdots \\ \tilde{\mathbf{h}}_N^{(l)}(t) \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{h}}_1^{(l-1)}(t) \|\mathbf{f}_{i1}^e(t_{i1})\| \phi(t - t_{i1}) \\ \vdots \\ \tilde{\mathbf{h}}_N^{(l-1)}(t) \|\mathbf{f}_{iN}^e(t_{iN})\| \phi(t - t_{iN}) \end{bmatrix} \quad (4-5)$$

其中， $\phi(\cdot)$  是一个可训练的时间映射函数，在实现上是使用一个多层次感知机来拟合，它将输入的时间值映射到特征嵌入空间，其具体的形式会在4.2.3小节中描述。



17010226

我们将  $\mathbf{Z}(t)$  分别与三个不同的线性映射矩阵相乘，就可以得到4.2.1小节中所述的“查询”、“键”和“值”：

$$\begin{aligned}\mathbf{q}(t) &= [\mathbf{Z}(t)]_i \mathbf{W}_Q, \\ \mathbf{K}(t) &= \mathbf{Z}(t) \mathbf{W}_K, \\ \mathbf{V}(t) &= \mathbf{Z}(t) \mathbf{W}_V\end{aligned}\tag{4-6}$$

其中， $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V \in \mathcal{R}^{(d+d^T) \times d^h}$  是用来计算时间编码以及节点特征的注意力权重矩阵。该注意力权重矩阵可以使用类似公式4-3中的形式计算：

$$\alpha(\mathbf{q}(t), \mathbf{K}(t)) = \text{softmax} \left( \frac{\mathbf{q}(t) \mathbf{K}(t)^T}{\sqrt{d}} \right)\tag{4-7}$$

上述公式中  $d$  是节点特征  $[\mathbf{Z}(t)]_i$  的维度， $\alpha$  矩阵中的每一行记为  $\{\alpha_i\}_{i=1}^N$ ，它的计算结果来自

$$\alpha_i = \frac{\exp(\mathbf{q}^\top \mathbf{K}_i)}{\sum_q \exp(\mathbf{q}^\top \mathbf{K}_q)}\tag{4-8}$$

其揭示了节点  $v_i$  考虑了其本身的特征  $[\mathbf{Z}(t)]_i$  以及所有时态邻居  $N(v_i; t)$  的特征的相关性（内积），从而计算出注意力权重。因此，注意力机制通过定义了一个局部的动态聚合机制，可以捕获动态图上的节点交互的信息，同时也可以捕获节点的特征以及图上的拓扑结构特征。此外，该机制可以在所有未见的节点上有效共享，以在测试阶段处理新的图信息。

然后，我们将上述点积自我注意输出的行和作为隐藏邻域表示，即：

$$\mathbf{h}(t) = \alpha \mathbf{V}(t)\tag{4-9}$$

为了将某目标节点的邻居信息的表示与目标节点的特征相结合，我们将邻域表示与目标节点的特征向量  $z_0$  连接起来，然后接入了一个前馈型神经网络来使之具有非线性的特征映射能力：

$$\begin{aligned}\tilde{\mathbf{h}}_i^{(l)}(t) &= \text{FFN}(\mathbf{h}(t) \| \mathbf{x}_i) \equiv \text{ReLU} \left( [\mathbf{h}(t) \| \mathbf{x}_i] \mathbf{W}_0^{(l)} + \mathbf{b}_0^{(l)} \right) \mathbf{W}_1^{(l)} + \mathbf{b}_1^{(l)} \\ \mathbf{W}_0^{(l)} &\in \mathbb{R}^{(d_h+d_0) \times d_f}, \mathbf{W}_1^{(l)} \in \mathbb{R}^{d_f \times d}, \mathbf{b}_0^{(l)} \in \mathbb{R}^{d_f}, \mathbf{b}_1^{(l)} \in \mathbb{R}^d\end{aligned}\tag{4-10}$$

其中  $\mathbf{x}_i$  是图上节点的特征向量， $\tilde{\mathbf{h}}_i^{(l)}(t)$  是节点  $v_i$  经过  $l$  层的动态图注意力网络之后的最终输出即节点在  $t$  时刻的动态节点嵌入， $\mathbf{W}_0^{(l)}, \mathbf{W}_1^{(l)}, \mathbf{b}_0^{(l)}, \mathbf{b}_1^{(l)}$  是神经网络中的线性映射矩阵以及偏移向量。如果进一步推广到多头注意力的情况，那么只需要产生多个不同的注意力特征  $\mathbf{h}^{(i)}(t) = \text{softmax} \left( \frac{\mathbf{Q} \mathbf{K}^\top}{\sqrt{d}} \right) \mathcal{V}, i = 1, \dots, k$ ， $k$  是动态图



17010226

注意力模型中的注意力头的个数。并把这些不同注意力头产生的注意力特征进行拼接在经过前馈神经网络就可以得到最终的注意力特征。

$$\tilde{\mathbf{h}}_0^{(l)}(t) = \text{FFN} \left( \mathbf{h}^{(1)}(t) \| \dots \| \mathbf{h}^{(k)}(t) \| \mathbf{x}_0 \right) \quad (4-11)$$

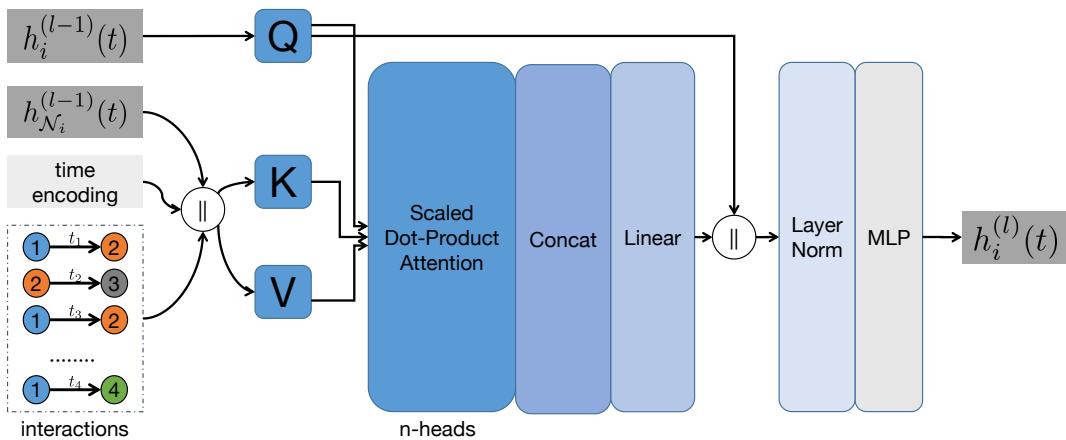


图 4-2 本章节提出的基于空间-时间联合分布建模的动态关系推理模型 TERRINE 的核心编码组件是一个基于多头注意力机制的编码器。

Figure 4-2 The encoder of TERRINE is a multi-head attention module, which would calculate the current node embedding  $h^{(l)}(t) \in \mathbb{R}^d$  according to the relativity between last undated embedding  $h^{(l)}(t) \in \mathbb{R}^d$  and interactions' features. Where the  $h_{N_i}^{(l-1)}(t)$  means the embeddings of target node  $v_i$  in last layer,  $\|$  means the concatenate operation, and  $n$  means the number of attention-heads.

本模型的整体架构如图4-2所示，对于某个节点  $v_i$  来说，如果要通过某一层动态图注意力网络计算该节点的特征表示，则该节点的特征  $\tilde{\mathbf{h}}_i^{(l-1)}(t)$  会作为“查询”，节点  $v_i$  的时态邻居节点  $N(v_i; t)$ （简写为  $N_i$ ）的特征  $\tilde{\mathbf{h}}_{N(v_i;t)}^{(l-1)}(t)$ ，以及节点  $v_i$  及其邻居之间的交互信息和时间戳会作为注意力结构中的“键”和“值”，一起被送入点积注意力网络中计算，其输出再经过一个线性的前馈型神经网络，与节点  $v_i$  的特征  $\tilde{\mathbf{h}}_i^{(l-1)}(t)$  拼接，起到一个即保留考虑邻居的特征，也保留自己的特征的目的。

**层次标准化 (Layer Normalization)** 由于不同节点的注意输出是不同的，我们需要一个归一化方案来限制输出的均值和方差，以便让注意力层得到更精准的训练。层次标准化<sup>[176]</sup>是注意模型中最常见的选择，因为复杂的注意机制可能会破坏每个数据批次内的统计分布。如果我们交替使用在机器学习领域中更常见的批量标准化 (Batch Normalization)，可能会导致次优结果。层次标准化通过计算用于标准化的平均值和方差来实现这一目标，这些平均值和方差来自一个层中神经



17010226

元的所有求和输入：

$$\begin{aligned}\mu &= \frac{1}{d} \sum_{i=1}^d \mathbf{a}_i, \quad \sigma = \sqrt{\frac{1}{d} \sum_{i=1}^d (a_i - \mu)^2} \\ \bar{a} &= f\left(\frac{g}{\sqrt{\sigma^2}} \odot (a - \mu) + \mathbf{b}\right)\end{aligned}\quad (4-12)$$

$d$  代表层次标准化层的输入维度， $\mu$  和  $\sigma$  代表均值和方差。 $\odot$  代表两个向量之间按元素相乘。两个可学习的参数  $\mathbf{b}$  和  $g$  定义为偏差和增益，以确保标准化操作对原始信息没有影响。之后，层次标准化的输出将被传送到多层感知机（MLP）网络中，以生成新的节点特征表示向量  $\mathbf{h}_i^{(l)}(t)$ 。

#### 4.2.3 带有时空约束关系的动态图嵌入

根据上一小节所描述的动态图注意力聚合模块，给定一个待查询的节点和查询时间戳  $t_0$ ，模块可以生成该节点在  $t_0$  时刻的动态节点嵌入。关系推理模块的功能就是根据节点嵌入，预测最有可能首先跟图中的哪个其他节点相连，以及这条连边的时间戳  $t$  是什么。可以用数学表达式来说明这个问题，动态关系推理就是让模型学习一个映射  $\Phi : (\mathcal{G}_{t_0-}, v^s) \Rightarrow (v^d, \Delta t)$ ，其中  $\Delta t = t - t_0$ 。

然而由于时间戳是无穷无尽的，可以认为预测时间戳可以类比于一个分类类别近似于无穷的分类问题，毫无疑问，这个训练目标是非常困难的。更何况我们要做的是空间-时间的联合训练，由于在大规模的图数据集中节点数量可能达到上亿，如果我们要直接完成上亿类的分类问题再加上无穷的未来时间戳的联合预测，这毫无疑问是相当困难的。为了达成这个目标，我们不采用直接端到端预测的方式直接让模型学习如此困难的映射，而是采用某种方法先让模型学会如何提取节点关联和时间关联的特征嵌入。

首先，我们将预测问题先转化为一个判断问题，即：在给定历史的图数据  $\mathcal{G}_{t_0-}$  的情况下，判断一个三元组  $(v^s, v^d, t)$  是否在未来具有存在性，意思是在未来的  $t$  时刻，节点  $v^s$  和  $v^d$  是否会发生交互，也就是估计概率  $p((v^s, v^d, t) | \mathcal{G}_{t_0-})$  的大小。由此，为了解决预测问题，我们先解决一个样本的判别分类的子问题。为了达到这个目的，我们必须为每个存在于数据集中的正样本三元组  $(v^s, v^d, t) \in \mathcal{E}$  引入数个负样本  $(v^s, v_{\text{neg}}^d, t_{\text{neg}}) \notin \mathcal{E}$ ，注意，在任意一对正负三元组中，其源节点  $v^s$  是相同的，我们仅仅只需不断地在训练过程的迭代中随机变换目标节点和时间戳，就可以让模型学习估计某个三元组存在的概率。

TERRINE 模型具体的训练形式受到文献 TransE 模型<sup>[177]</sup>的启发，该文献致力于在知识图谱中推理关系，即通过已有的知识图谱，推理图谱中一个实体与另一



17010226

个实体的潜在关系。比如加入我们已知“比尔盖茨-是总裁-微软”和“微软-总部在-西雅图”这两对关系，我们希望推出“比尔盖茨-居住-西雅图”这个潜在的关系。TransE 模型通过学习实体和关系在特征嵌入空间的位置关系来推理，即 TransE 的训练目标是令向量“比尔盖茨” + 向量“是总裁”约等于向量“微软”，通过这种事实依据的训练目标，TransE 模型可以将代表比尔盖茨、微软、是总裁等等在知识图谱中的实体和关系映射到特征嵌入空间，并在该空间中推理额外潜在的关系。

基于此，我们设计了一种用于动态关系推理的特征映射框架，在 TransE<sup>[177]</sup>中，作者认为知识图谱中的两个实体向量之间的关系可以由某个关系向量来表述，而在我们的 TERRINE 模型中，我们认为动态图上的两个节点的连接关系可以由一个时间关系向量来表示。具体来说在特征空间中，我们希望动态图上的节点向量表达和时间向量表达可以满足以下两种时空相关关系假设。首先是时空三角关系假设：

$$\overrightarrow{\phi^V(v^d; t_0)} \approx \overrightarrow{\phi^V(v^s; t_0)} + \overrightarrow{\phi^T(t - t_0)} \quad (4-13)$$

然后是时间戳与时间向量的单调性假设：

$$\left| \overrightarrow{\phi^T(t_i - t_0)} \right|_2 > \left| \overrightarrow{\phi^T(t_j - t_0)} \right|_2, \forall t_i > t_j \quad (4-14)$$

其中  $\phi^V$  和  $\phi^T$  分别代表节点和时间值在动态特征嵌入空间的映射， $|\cdot|_2$  代表一个向量的 2 范数， $\phi^V$  是我们在上一节中描述的动态图注意力聚合模块，而  $\phi^T(t) = \mathbf{w}(t) + \mathbf{b}$  是一个纯的线性映射层，并且跟上一节中公式4-5中描述的时间映射函数共享参数。

时空三角关系假设的含义是，对于未来的图上的交互事件  $(v^s, v^d, t) \in \mathcal{E}$  来说，在特征嵌入空间中，源节点的嵌入向量  $\overrightarrow{\phi^V(v^s; t_0)}$  加上当前时间戳和未来事件的时间戳的差值的嵌入  $\overrightarrow{\phi^T(t - t_0)}$  约等于目标节点的嵌入向量  $\overrightarrow{\phi^V(v^d; t_0)}$ 。通过这个假设，我们约束了时间和空间的关系，压缩了本来难以解决的动态关系预测问题  $v^s \rightarrow (v^d, t)$  的解空间。

而单调性假设的含义是，如果两个节点在嵌入空间的位置越近，首先代表两者的关系更近，意味着两者之间所夹的时间向量的范数越小。如果我们的时间戳映射函数  $\phi^T$  可以满足单调性假设的话，则这意味着两者更有可能在未来发生交互。借由单调性假设，我们可以令节点在嵌入空间中的位置具备时间-空间的二元语义信息。该假设在模型上的实现方法也非常简单，我们只需要约束  $\phi^T$ ，令这个神经网络的参数矩阵一个正定的约束，又因为时间戳总是正的，所以当时间戳越大的时候，其对应的时间向量的范数总是会正相关地变大。

如图4-3所示，我们希望源顶点、时间、目标顶点在特征嵌入空间中形成一个



17010226

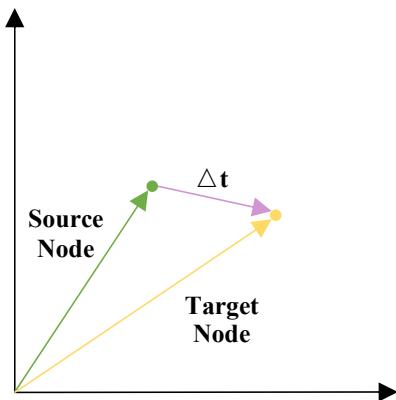


图 4-3 TERRINE 模型的训练目标示意图

Figure 4-3 Schematic diagram of training objectives of TERRINE model

三角闭环，同时，我们认为时间不是决定交互事件是否发生的主要因素，源顶点和目标顶点在嵌入空间中的相似度才是事件的决定性因素，而引入时间所产生的闭环只是为了推理这两个节点究竟何时会发生交互。同时，由时间映射函数  $\phi^T(t)$  的性质可以得出结论：如果两个向量的关系特别近，则意味着它们的关系仅仅需要一个范数较小的时间关系向量来描述。这个非常自然且合理，因为时间向量的范数较小意味着这两个节点下一次可能发生交互的时间较近；反之，如果两个向量的关系特别远，则需要更长时间它们才有可能发生交互，也就是需要范数更大的时间嵌入向量来描述。

由上述的灵感，我们可以得到 TERRINE 模型的损失函数：

$$\mathcal{L} = \sum_{(v^s, v^d, t) \in \mathcal{E}} \sum_{(v^s, v^{d'}, t') \in \mathcal{E}'} [\gamma + \|v^s + \Delta t - v^d\|_2 - \|v^s + \Delta t' - v^{d'}\|_2]_+ \quad (4-15)$$

其中，加粗的符号代表三元组中的元素在查询时间戳  $t_0$  是在特征嵌入空间中的表达向量，即  $v^s = \overrightarrow{\phi^V(v^s; t_0)}$ ,  $v^d = \overrightarrow{\phi^V(v^d; t_0)}$ ,  $\Delta t = \overrightarrow{\phi^T(t - t_0)}$ ;  $\|\mathbf{x}\|_2$  代表向量  $\mathbf{x}$  的二范数； $[x]_+$  代表  $x$  的最小值为 0，即  $[x]_+ = \max(0, x)$ ； $\gamma$  代表优化边界，其的作用相当于是一个正确的三元组与错误三元组之前的间隔修正， $\gamma$  越大，则两个三元组之之间被修正的间隔就越大，则对于节点和节点之间的嵌入关系映射的修正就越严格，但是太严格的修正也会导致模型难以训练，所以这是一个超参数； $\mathcal{E}'$  代表负样本的集合，它在 TERRINE 模型的训练中非常重要。之所以说负样本集合在模型训练中非常重要，其原因是数据是模型的养料，在我们的任务中，我们仅仅只有正样本，如果我们不能提供精巧的负样本来帮助模型描述正负样本的区别的话，那模型就难以学习到有用的节点和时间嵌入向量。



17010226

我们设计了一种空间-时间联合负采样，简单的来说，对于一个图中存在的正三元组  $(v^s, v^d, t) \in \mathcal{E}$ ，我们生成两个不同的负样本，一个负样本仅更改目标节点  $(v^s, v^{d'}, t), v^{d'} \in \mathcal{V}$  以及另一个负样本仅更改时间  $(v^s, v^d, t')$ 。 $v^{d'}$  是从动态图的节点集中均匀采样得到的，而  $t'$  是由  $t$  叠加了一个从  $(-10\%t, 10\%t)$  采样的均匀分布得来的。

此外，在训练的损失函数  $\mathcal{L}$  中理应存在对于模型以及时间向量的范数 L2 最小化正则，但计算这些正则项会引入额外的计算成本。在训练时，我们引入了权值衰减（Weight Decay）<sup>[178]</sup>策略，在梯度优化时引入权值衰减，就可以为模型加入类似 L2 正则化的作用，并且几乎不引入额外的计算量。

#### 4.2.4 模型分层推理机制

在上一小节中，我们提出暂时搁置让模型学习困难问题  $\Phi : (\mathcal{G}_{t_0-}, v^s) \Rightarrow (v^d, t)$ ，转而让模型先学会通过判断三元组表示  $(v^s, v^d, t)$  在图上的存在性这个子问题，来学习如何表示图上的节点以及事件发生的时间关系，即学会如何将三元组的三个元素分别映射在特征嵌入空间中。然而，仅仅解决一个正负三元组的分类子问题还是不能达到动态关系推理的任务目标，我们需要利用三元组在特征嵌入空间的位置来最终解决这个困难的预测问题，为此我们引入了第二阶段的问题转化。

在第一阶段，我们得到了一个概率模型  $p((v^s, v^d, t) | \mathcal{G}_{t_0-})$ ，而我们的动态关系预测的目标是要根据  $\mathcal{G}_{t_0-}$  和  $v^s$ ，联合预测  $v^s$  下一步会跟哪个节点  $v^d$  交互，以及交互的时间戳  $t$  是什么。幸运的是，经过第一阶段的模型训练，模型拥有了将节点以及时间映射到特征嵌入空间的能力，接下来，我们就根据特征嵌入空间的相对位置关系，来根据源节点  $v^s$  预测目标节点  $v^d$  和时间  $t$ 。

在特征嵌入空间中，给定一个代表源节点的向量  $\mathbf{v}^s$ ，我们可以根据计算  $\mathbf{v}^s$  与其他节点向量的内积  $\langle \mathbf{v}^s, \mathbf{v}^r \rangle, v_r \in \mathcal{V}, r \neq s$  的大小来找到最有可能跟该节点发生的交互的其他节点  $\mathbf{v}^d$ ，内积越大代表两个节点的属性越相关。紧接着，由 TERRINE 模型的训练目标有： $\Delta \mathbf{t} = \mathbf{v}^d - \mathbf{v}^s$ ，即可求出  $\mathbf{v}^s$  和  $\mathbf{v}^d$  之间所存在的时间向量  $\Delta \mathbf{t}$ 。

然而，仅仅知道时间向量  $\Delta \mathbf{t}$  的话我们无法确切地知道具体的时间戳数值  $t = t_0 + \Delta t$ 。幸运的是，我们可以用简单的岭回归<sup>[179]</sup>方法来解这个问题。岭回归是一种广义的线性回归方法，其可以通过一批可见的特征数据  $\{\mathbf{x}_i\}, \dots, \mathbf{x}_n$  来预测待回归的标签值  $\{y_1, \dots, y_n\}$ 。在我们 TERRINE 模型的训练过程中，我们积累了许多时间向量和时间值的对应关系： $(\mathcal{T}, \tau) = \{(\overrightarrow{\Delta \mathbf{t}_i}, \Delta t_i), i = 1, 2, \dots\}$ ，其中  $\mathcal{T}$  代表训练过程中累计的时间向量集合， $\tau$  代表时间向量对应的时间值集合。



17010226

---

**算法 4-1 TERRINE 模型推理过程**

---

**Input:** 动态图  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ , 查询时间戳  $t_0$ , 源节点  $v^s \in \mathcal{V}$ ,节点特征映射函数  $\phi^{\mathcal{V}}$ , 岭回归模型  $\theta$ **Output:** 源节点  $v^s$  发生的下一个动态图事件  $(v^s, v^d, t) \in \mathcal{E}$ 在时间戳  $t_0$ , 使用  $\phi^{\mathcal{V}}$  为图中所有需要更新的节点学习节点嵌入在特征嵌入空间寻找距离  $v^s$  最近的节点作为预测的目标节点  $v^d$ 根据闭合三角关系计算时间向量  $\vec{\Delta t} = \vec{v}^d - \vec{v}^s$ 使用岭回归模型  $\theta$  计算时间戳:  $t = t_0 + \vec{\Delta t}_i \theta$ **return**  $(v^s, v^d, t)$ 

根据岭回归<sup>[179]</sup>理论, 可以用参数  $\theta$  来逼近这个对应关系, 即  $\mathcal{T}\theta \approx \tau$ , 参数  $\theta$  的解是  $\theta = (\mathcal{T}^T \mathcal{T} + \mathcal{I})^{-1} \mathcal{T}^T \tau$ , 则对于任意一个时间向量  $\vec{\Delta t}_i$ , 由岭回归模型可以求得其对应的时间值为:  $\Delta t = \vec{\Delta t}_i \theta$ 。岭回归模型仅仅只会在每轮次 (Epoch) 训练结束后被更新, 而一个轮次中 TERRINE 模型可能会被训练数千次, 所以岭回归模型所带来的额外计算成本可以忽略不计, 其参数数量仅仅等于特征嵌入空间的维度, 一般来说岭回归的参数数量会小于 100。

我们使用算法流程4-1再次总结 TERRINE 模型的推理预测过程。在已知源节点  $v^s$  的情况下, 根据图上的各个节点在特征嵌入空间中的距离远近关系, 我们可以求得目标节点  $v^d$ , 然后根据  $\vec{t} = \vec{v}^d - \vec{v}^s$ , 得出节点  $v^s$  和  $v^d$  的交互时间向量, 再根据由岭回归模型所描述的时间向量和时间值的对应关系, 预测出最终该交互发生的时间值  $t$ , 组成图上的交互事件三元组  $(v^s, v^d, t)$ 。

以上是单步预测的情况, 本框架只需要很自然的改动就可以支持  $k$  步预测。根据源节点  $v^s$  跟其他节点在嵌入空间的位置关系, 我们可以按照内积由大到小的顺序依次选择  $k$  个其他节点作为目标节点, 然后再根据源节点  $v^s$  与这  $k$  个目标节点在空间中的三角关系一一对应求得  $k$  个时间向量, 再通过回归模型计算  $k$  个时间值就可以完成  $k$  个三元组的预测。

### 4.3 对比实验与结果分析

在这一小节中, 我们在四个公开的、现实世界动态图数据集上测试了所提出方法的性能, 这些数据集包括 Wikipedia、MOOC<sup>[21]</sup>、GitHub<sup>[26]</sup>和 SocialEvo<sup>[180]</sup>四个数据集。

**Wikipedia**<sup>①[21]</sup>, 中文名可译为“维基百科”数据集, 它被广泛用于动态的推

---

① <http://snap.stanford.edu/jodie>



17010226

荐系统的评估中，它是一个包含用户和维基百科页面作为节点类型，以用户对百科的编辑行为作为边的”二部图“（二部图的定义详见第二章中的定义2.7）。交互频率遵循长尾分布，一些节点具有相对较高的交互频率，而另一些节点只有少量交互。每个用户和每个项目的平均交互次数分别为和 157。我们将每次编辑的文本转换为表示其 LIWC 类别<sup>[164]</sup>的边特征向量。

**MOOC**<sup>①[21]</sup>数据集即慕客平台数据集。慕课平台即大型开放式网络课程（Massive Open Online Courses），大多数慕课平台针对高等教育，并且像真正的大学一样，为在线的学生提供系统性的学习和课程管理系统。MOOC 数据集收集自中文慕课类型的网站“学堂在线（XuetangX）”，学堂在线是清华大学联合教育部在线教育研究中心于 2013 年 10 月发起建立的慕课平台，该数据集中包括学生对于 MOOC 课程的操作，比如观看某节课的视频、提交一份作业等等。该数据集由 7047 名学生组成，他们与 98 个项目（视频、答案等）进行互动，互动超过 411749 次。每个交互都与文献<sup>[21]</sup>中提供的特征向量相关联。

**Github**<sup>①[26]</sup>数据集来源于一个面向开源及私有软件项目的托管平台“Github”。数以百万计的开发者和公司在全球最大、最先进的开发平台 GitHub 上构建、发布他们的软件并且不断维护它们，包括编写说明文档，修复和提交 bug 等等。Github 数据集是一个基于 GitHub 用户活动构建的社交网络，其中所有节点都是真实的 GitHub 用户，交互代表用户对另一方存储库的操作，如包括监视（Watch）、点赞（Star）、拷贝（Fork）、推送（Push）、创建问题（Creating Issue）、对问题进行评论（Commenting on Issue）、代码合并请求（Pull Request）、提交（Commit）等。

**SocialEvo**<sup>②[26,180]</sup>数据集是一个小型的社交网络数据，其由麻省理工学院（MIT）人类动态实验室（Human Dynamics Lab）收集。该数据集拥有 83 个个体和大约 6 万次交互，显而易见的是，该数据集历史交互次数更多，相比其他数据集有更丰富的历史信息。

由于公共数据集 SocialEvo 和 Github 没有边特征，我们使用以下属性生成 10 维边特征，包括边的两个事件节点的当前度数，以及两个事件节点的当前时间戳和最后更新的时间戳之间的时间差。请注意，时差分别以天数、小时数、分钟数和秒数表示。表格5-1显示了我们实验中使用的数据集的统计信息。

---

① <https://www.githubarchive.org>

② <http://realitycommons.media.mit.edu/socialevolution4.html>



17010226

统计信息	Wikipedia	MOOC	Github	SocialEvo
边数 (Edges)	157,474	411,749	20,726	62,009
节点数 (Nodes)	9,227	7,145	282	83
最大度数 (Max Degree)	1,937	19,474	4,790	15,356
平均度数 (Mean Degree)	34	115	147	1,310
边特征维数 (Edge Feature Dimension)	172	4	10	10
是否二部图 (Is Bipartite)	True	True	False	False
时间区间 (Timespan)	31days	30days	1years	74days
数据集分割方式 (Data Spilt)	70%-15%-15% by timestamp order			

表 4-1 四个数据集的统计信息

Table 4-1 Statistics of the datasets used in our experiments.

### 4.3.1 基线方法

虽然我们提出的任务是崭新的，并且意味着过去没有在类似的设置下的文献，但是我们还是试图寻找了一下领域相关的文献方法并想方设法在之上做一些非平凡的改动以达到我们动态关系推理的目标。我们使用的基线模型包括三大类，并且所有方法都在2.4.4小节中有过说明。

**1. 动态图嵌入模型**这类模型的主要思路是通过图神经网络提取动态图的信息，并学习节点的动态嵌入表示，但这类方法的任务中只考虑了静态的关系预测任务而没有考虑时间戳的联合预测，我们为其强行加入了时间戳预测的功能，这可能不够合理但是这是将这类方法应用在动态关系推理任务上的最简单方法。这类基线方法中，我们主要考虑两个最先进方法：

- **动态图注意力网络 TGAT<sup>[22]</sup>**: 如第二章第2.4.3.2小节所述，利用时间编码的注意力机制的 TGAT 嵌入方法是一种无记忆的节点嵌入方法。
- **动态图网络 TGN<sup>[24]</sup>**: TGN 是 TGAT 的修改版本，它将节点上的内存更新策略引入 TGAT 的时间聚合阶段，为每个节点维护了一个使用 RNN 更新的内存。据称，它在链路预测任务中具有更好的性能。

**2. 不使用图信息的点过程模型**这类模型是应用在传统的事件序列上的随机点过程模型，这类模型建模的是某个事件  $y$  在时间  $t$  到  $t + dt$  之间所发生的条件概率，考虑的是事件类型  $y_i$  与发生时间  $t_i$  之间的关系。而这类模型在动态图上的直接应用就是把动态图上所有交互节点对  $(v^s, v^d)$  视为一个独立的事件，则此种方法所建模的事件个数就等于动态图上的节点数的平方  $|\mathcal{V}|^2$ ，这平方复杂度的建模毫无疑问限制了其在大规模图上的应用，但是我们仍然试图了解其在动态关系预测



17010226

上的能力。这类基线方法中，我们主要考虑：

- **TPP-Poisson:** 我们假设在每个节点对  $(v^s, v^d)$  处发生的事件遵循泊松过程，具有恒定的强度值  $\lambda(v^s, v^d)$ ，通过最大似然估计（Maximum Likelihood Estimation, MLE）从数据中学习。
- **TPP-Hawkes<sup>[181]</sup>:** 我们假设在每个节点对  $(v^s, v^d)$  处发生的事件遵循霍克斯过程<sup>[130]</sup>，其基本强度值为  $\mu_{v^s, v^d}$  和激励参数  $\alpha_{v^s, v^d}$  通过 MLE 从数据中学习。
- **RMTPP<sup>[33]</sup>:** 我们直接考虑每个源节点和目标节点对作为唯一的标记。我们注意到，这将消耗非常巨量的内存和时间，因为 RMTPP 将为每个节点对分配一个可学习的嵌入，结果是 RMTPP 模型将会拥有  $|V|^2$  数量级的参数量，这对于计算系统来说太过于昂贵。
- **NeuralHawkes<sup>[34]</sup>:** 同 RMTPP 一样，我们将直接考虑每个源节点和目标节点对作为唯一的标记，这同样也会带来  $O(|V|^2)$  的复杂度。此外，NeuralHawkes 中使用的一种独热编码还会额外地进一步降低计算速度。

**3. 动态图上的点过程模型** 由于点过程在动态图上的应用效果还不明朗，目前这类文献较少，有开源代码的工作更加稀少，本章还是支持与这类工作中的最先进模型 DyRep 进行对比，这类工作结合了图信息与点过程的优势，既可以利用图上的空间关联，也可以利用图上的时间信息。该方法在理论上是我们 TERRINE 方法的最直接竞争者。

- **DyRep<sup>[26]</sup>:** 据我们所知，DyRep 是将时序点过程与图学习技术结合起来，同时对时间和空间依赖性进行建模的文献中最流行的。虽然 DyRep 原文并没有支持动态关系推理的任务，我们还是做了一些非平凡的修改来使之适应我们的任务。DyRep 的原文分别进行了静态关系预测任务  $v^s \rightarrow v^d$  以及时间戳预测任务  $(v^s, v^d) \rightarrow t_{sd}$ ，我们强迫 DyRep 在训练中同时学习这两个任务，然后在模型推理过程中，我们先执行  $v^s \rightarrow v^d$  再执行  $(v^s, v^d) \rightarrow t_{sd}$ ，就可以完成  $v^s \rightarrow (v^d, t_{sd})$  映射。经过修改后的 DyRep 方法与我们的方法的本质区别在于，DyRep 使用了图信息建模了  $v^s, v^d$  的在嵌入空间中的表达和强度函数，但对于时间戳的预测则是单独使用了简单的多层感知机模型结合蒙特卡洛采样。而 TERRINE 模型则是显式地在特征嵌入空间中描述了  $\vec{t}, \vec{v}^d, \vec{v}^s$  这三者之间的关系。此外，由于 DyRep 模型的主要模型框架由循环神经网络组成，这限制了其在大规模图数据中的并行训练，而我们的模型是基于注意力机制的，支持大规模的并行训练。



17010226

### 4.3.2 评估指标

为了检验模型预测遥远未来的能力，每个模型只能学习训练集中的事件，并不能观察到验证集和测试集（非训练集）中的未来事件。一旦模型训练完毕，则所有参数都会被固定，验证集和测试集中的未来事件不能对模型的参数有任何的影响。我们对数据集中所有的未来事件按照源节点进行聚合，就可以得到该源节点未来的交互序列  $\mathcal{I} = \{(v_i^d, t_i), i = 1, 2, \dots\}$ ，这个交互序列即可作为真实的标签。对模型方面，我们令每个基线模型对于待预测的源节点输出跟该真实交互序列  $\mathcal{I}$  长度相同的预测序列。对于这两个序列，我们希望两者在目标节点和时间的预测上都能尽可能准确，并且我们也希望这两个指标可以分开来进行估计，以便我们探究模型分别在空间和时间上的建模能力。

因此，对于每一对真实交互序列和预测交互序列，我们分别采用目标节点序列的命中准确率（Error）和时间序列的平均绝对误差（Mean Absolute Error, MAE）来分别刻画模型对于空间连接和时间关系的预测能力。这种预测模式同样也被经常使用在交通流预测领域中<sup>[47]</sup>。对于在非训练集中的每个源节点来说，我们都可以计算一个 Error 和 MAE，我们将所有非训练集中的源节点对应的 Error 和 MAE 分别做平均，就可以得到平均 Error 和平均 MAE 指标，本章就采用这两个指标来汇报各种模型的表现，

### 4.3.3 实验细节

在本章中，我们汇报了 TERRINE 模型和其他基线模型的实现细节，包括网络的详细结构、超参数的选择以及实验所运行的环境。在软件环境方面，我们使用 PyTorch<sup>[148]</sup> 学习框架以及 Deep Graph Library<sup>[43]</sup> 来实现我们所有的模型，在硬件条件方面，我们在 i7-9750H CPU、16GB 内存、Nvidia T40 显卡的 Linux 系统机器上训练我们的模型和基线模型。对于所有基线模型，我们采用 AdamW<sup>[178]</sup> 随机梯度优化器，学习率为 0.0001，训练、验证和测试时我们使用的批量大小均为 1024，Dropout 比率为 0.1，并且使用了早停策略<sup>[165]</sup>，早停耐心值为 5。对于基线模型的其他参数，我们采用跟它们原始论文相同的参数设置。对于 TERRINE 模型来说，网络节点和时间嵌入的特征维度设为 64，动态图注意力层中的注意力头的数量设置为 4，图神经网络的消息传递层数为 2。对于动态图注意力层和时间映射函数中的 MLP 网络，我们采用了隐藏大小为 128（两倍的嵌入特征维度）的两层前馈神经网络。我们还将所有的超参数在表格 4-2 中展示，方便读者参考和复现。



17010226

参数名	取值
节点 & 时间嵌入维度	64
所有 MLP 隐含层维度	128
所有 MLP 层数	3
所有 Dropout 层比率	0.1
邻居采样个数	15
图神经网络层数	2
模型学习率	0.0001
Optimizer	AdamW <sup>[178]</sup>
注意力头的个数	4
训练轮次	80
数据批量大小	1024

表 4-2 TERRINE 模型的参数配置

Table 4-2 Parameter configuration of TERRINE model

Datasets	Wikipedia		Github		MOOC		SocialEvo	
Metric	Error(%)	MAE	Error(%)	MAE	Error(%)	MAE	Error(%)	MAE
TPP-Hawkes	51.83(0.45)	0.401(0.029)	92.60(2.09)	0.261(0.016)	95.11(1.80)	0.279(0.004)	29.82(2.57)	0.364(0.013)
TPP-Poisson	74.07(1.25)	0.413(0.045)	96.02(1.62)	0.273(0.004)	60.68(2.30)	0.277(0.002)	11.77(0.16)	0.353(0.051)
RMTPP	42.59(0.51)	0.304(0.089)	18.25(2.15)	0.131(0.031)	25.74(1.97)	<u>0.194(0.016)</u>	<u>10.13(1.03)</u>	0.303(0.005)
NeuralHawkes	51.33(2.45)	0.299(0.105)	30.78(2.52)	0.260(0.016)	25.51(0.77)	0.201(0.009)	12.05(1.22)	0.343(0.013)
TGAT	46.85(1.02)	0.358(0.019)	17.59(0.58)	0.312(0.023)	20.81(0.65)	0.197(0.011)	20.01(1.09)	0.305(0.004)
TGN	43.81(0.42)	0.294(0.111)	<b>15.43(1.35)</b>	0.284(0.006)	<u>18.27(0.48)</u>	0.240(0.006)	10.82(0.44)	0.275(0.010)
DyRep	<u>41.58(1.72)</u>	<b>0.093(0.022)</b>	19.77(2.16)	<b>0.084(0.006)</b>	26.56(2.18)	0.211(0.016)	11.12(0.53)	<u>0.274(0.010)</u>
TERRINE	<b>26.89(0.64)</b>	<u>0.094(0.020)</u>	16.76(0.42)	<u>0.087(0.023)</u>	<b>17.64(0.21)</b>	<b>0.174(0.006)</b>	<b>7.87(0.08)</b>	<b>0.273(0.004)</b>

表 4-3 对比实验结果分析

Table 4-3 Analysis of comparative experimental results

#### 4.3.4 结果分析

表格4-3中展示了我们的 TERRINE 模型以及其他七种基线模型在四个数据集、两种评价指标上的表现对比分析，对于两种指标来说，我们汇报的是该指标在十次随机实验中的平均数和标准差，都是数值越小性能越优秀。最优秀的模型性能我们用**加粗字体**表示，第二名的模型我们用下划线来表示。我们可以得出结论，在这两种指标上，与不同数据集的其他基线模型相比，我们的 TERRINE 模型具有明显的优势。此外，我们分别对各类模型在目标节点预测（空间预测）准确率以及时间预测准确率上进行分析。



17010226

首先是空间预测方面,可以看出由于 TGAT、TGN、Dyrep 以及我们的 TERRINE 模型使用了图信息,这类模型在空间预测的性能上明显比不使用图信息的点过程模型好。并且基于图信息的模型比基于点过程模型更加节约内存,因为时序点过程的模型需要在图上的每条边都建立强度函数的估计,而其他模型仅仅需要一个统一且共享的模型来估计。

而在时间预测方面,基于点过程的模型则比没有点过程的模型要稍有优势,这一点在 Dyrep 和 TGN 的对比上可以看出,两者同样使用了图模型,但是由于点过程的引入, DyRep 在时间戳预测的准确率上明显好于 TGN。

TERRINE 模型和 Dyrep 相比,都是同时使用了图信息和时间的建模,不同的是, DyRep 根据节点的特征嵌入,使用了点过程来直接预测时间戳,而 TERRINE 将时间向量和节点嵌入向量放在同一个空间考虑,进一步地增强了节点和时间预测的一致性。

此外,值得一提的是在 SocialEvo 数据集中, RMTPP 在时间戳预测方面表现出人意料地好。一个可能的原因是 SocialEvo 数据集中的节点交互非常频繁,导致了一个非常密集的交互网络,其中来自邻居事件的影响变得不那么重要。

#### 4.3.5 参数敏感性分析

在本小节中,我们分析了 TERRINE 模型的参数敏感性,我们分别考虑了该模型对于邻居采样个数、GNN 层数、嵌入空间的维度以及注意力头的敏感性关系。再次提醒,汇报的指标越小,模型的性能越强,每个数据点上的封闭短线代表 10 次实验的标准差。如图4-4所示,随着每层 GNN 采样的邻居从 5 增加到 50,我

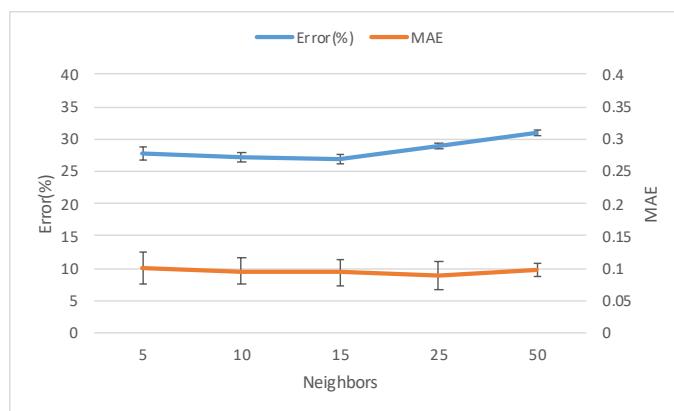


图 4-4 TERRINE 的性能与邻居采样个数的关系

Figure 4-4 Relationship between TERRINE's performance and sampled neighbors

们发现模型在采样个数等于 10~15 时达到最优性能,但是采样个数为 10 时模型



17010226

的性能同样也可以接受，而当采样个数等于 25 时，虽然时间预测的准确度依然在小幅提高，但是空间预测的准确率大幅下降了，这是因为过多的采样邻居会导致节点接收到的冗余信息过多，造成模型无法学习有表现力的节点嵌入。类似的情况在采样邻居数量等于 50 的时候更加严重，这也跟文献<sup>[77]</sup>所提到的结论类似。图4-5展示了 TERRINE 的性能与 GNN 层数的关系，可以看出来我们的 TERRINE

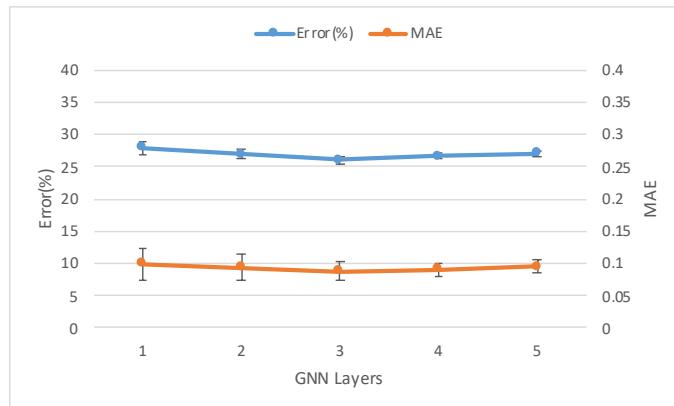


图 4-5 TERRINE 的性能与 GNN 层数的关系

Figure 4-5 Relationship between TERRINE’s performance and GNN layers

模型基本不受其层数的影响，仅仅只是在层数为 1 的时候性能出现了明显的下降。其原因是图中节点的信息关联类似于社交网络中的好友关系，一个个体的可能经常会跟其一度好友二度好友发生交互，但基本上不与三度好友交互，所以模型层数过多对于模型的表达能力并没有显著的提升<sup>[73]</sup>。图4-6代表的是图中的节点

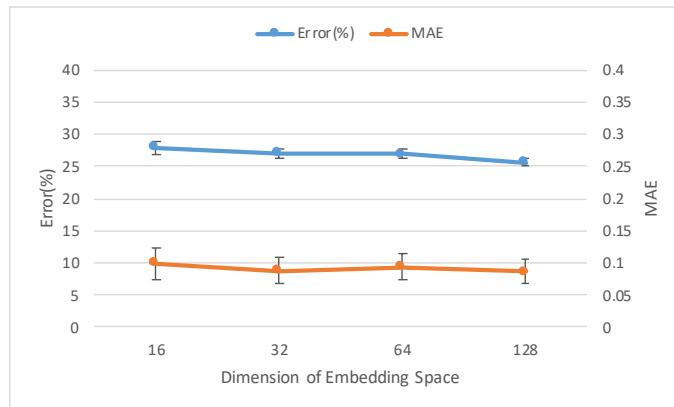


图 4-6 TERRINE 的性能与嵌入空间维度的关系

Figure 4-6 Relationship between TERRINE’s performance and embedding space dimension

和互相交互的时间在特征嵌入空间的映射维度给 TERRINE 模型带来的改变。从



17010226

图中我们似乎可以得出结论是随着特征嵌入空间的维度变大，其模型的性能也在变好，但我们仍然需要考虑维度变大所带来的计算成本与代价。我们认为，在模型的性能变动不明显的情况下，采用一个合理的嵌入维度就足够了。图4-7则是图

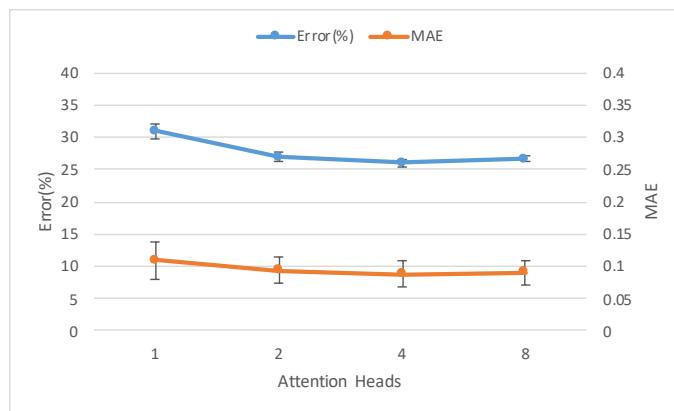


图 4-7 TERRINE 的性能与注意力头数的关系

Figure 4-7 Relationship between TERRINE’s performance and attention heads

注意力模型中注意力头数量所带来的影响。在基于注意力的模型中，不同的注意力头会学习关注不同方面的特征模式，通常来说，注意力头数增加会有助于模型的表现提升，但是该参数增加所带来的提升经常会在 4 以上的时候达到瓶颈，不但性能不会提升，而且会导致整体运行效率的下降。因此当遇到性能瓶颈时，我们不应该再增加注意力头数。

综上几个图表所带来的结论，我们认为 TERRINE 模型是一个对参数具有低敏感性的模型，这意味着该模型部署容易，不需要非常小心的调参就可以达到较为理想的效果。

#### 4.3.6 并行计算实验

为了验证我们的 TERRINE 模型在大规模数据集上的并行计算能力，我们监控了该模型在不同的并行进程数的情况下模型的训练过程。值得注意的是，CEP3 模型由于没有引入基于节点状态的编码器，所以使其能够按照5.2.2节所述，以小批量方式同时训练多个时间窗口的数据。这一特性使我们的模型能够享受小批量训练带来的好处，例如梯度稳定和更快的收敛速度。图5-9显示了我们在 GitHub 数据集上使用不同并行进程数的实验结果，很明显，即使我们只将批处理大小增加到 2，由困惑度度量的性能也显著提高。对于 CEP3 模型的训练来说，一个批次的数据由几个或十几个的社群数据组成，也就是说假如批次大小为 4，意味着模型在训练时一次性随机读取 4 个社群的历史数据。一次性读取的社群越多，则模



17010226

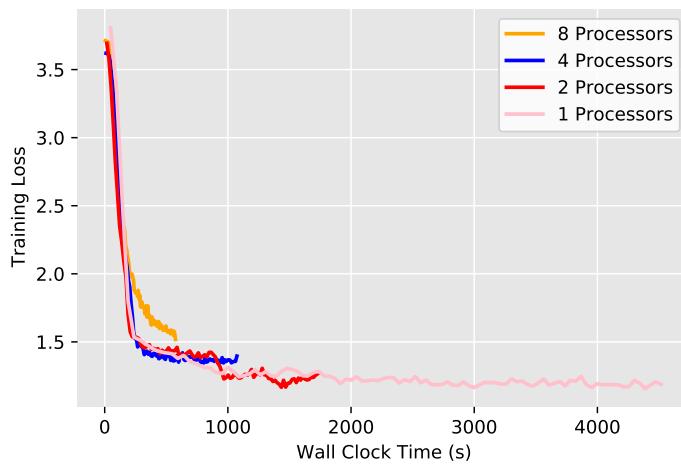


图 4-8 并行计算核心数目对于模型训练的影响

Figure 4-8 Effect of parallel processors on training performance

型的训练时间越快，但是，如果批量太大，收敛速度也会受到影响，因为优化程序没有足够的噪声级跳出坏的局部极小值。

#### 4.3.7 异常用户行为序列可视化分析

从上到下，图4-9分别在 Github、Wikipedia 和 MOOC 数据集的中以固定的圆形排布可视化某个异常用户的未来行为模式，可能由于异常用户是机器人进行有目的性的发帖和评论，这些异常用户通常与其他的用户间都存在着某种可预测可学习的高频连接关系。每同一行代表同一个用户在不同方法下的预测结果，同一列代表同一种方法在不同数据集下的结果。我们绘制了真实样本、TERRINE 模型预测和 DyRep 的预测以进行比较。绘制的节点大小表示其度数（Degree），而边的颜色表示在该时间窗口内的连接频率，颜色越深，频率越高。请注意，边颜色和节点大小只能在同一行中进行比较，不能在纵向比较。注意，在这里我们为了避免真实样本的交互序列数目与预测值差距过大，为了更清晰地为读者展示可视化的结果，我们在且仅在可视化的时候，只考虑与该用户真实交互的节点的预测结果。也就是说，这种情况下展示的是模型在框定真实候选节点之后的预测能力。我们希望同时在空间和时间维度展示模型的预测能力，即希望模型在一定的时间窗口内可以准确预测用户与其他节点所交互的频次。

下面我们分析一下用户行为可视化的结果。在第一行中，TERRINE 和 DyRep 都正确预测了该用户与其他三个节点有较高频次的连接，而 TERRINE 似乎增加了一条不存在的高频连接，Dyrep 则估计错了高频连接的位置关系。在第二行中，



17010226

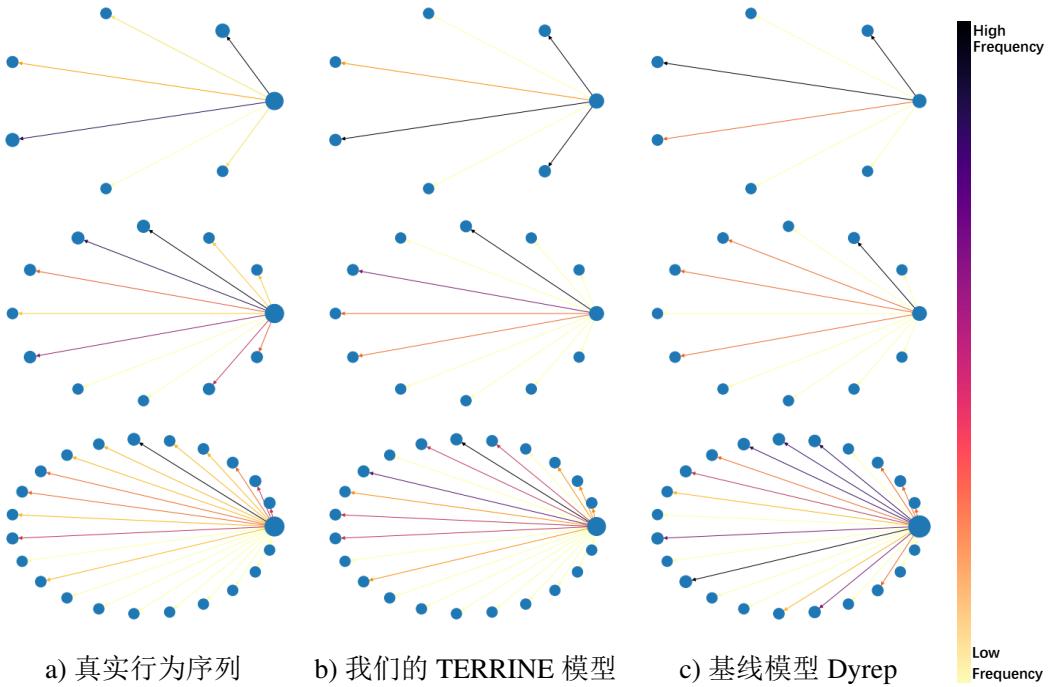


图 4-9 在测试集的整个时间窗口，针对几个特定用户在未来行为的预测可视化

Figure 4-9 The visualization of the predicted future behavior of several specific users in a selected time window

TERRINE 和 DyRep 都没能很好地预测正确的行为序列模式，而 TERRINE 的预测结果与真实值更相似，而 DyRep 生成了在原始图中不存在的高频黑色边。在第三行中，随着用户行为的逐渐复杂，TERRINE 和 DyRep 似乎都存在着过于平均预测的问题，很难明确地预报某种特有模式。TERRINE 成功地学习到了在图的上半部分中存在的一些高频连接模式，但 DyRep 在下半区却额外预测出了两条以上的较深颜色的连接。

我们可以看到，我们的方法成功地识别了具有高频交互模式的节点，捕获了一些用户动态行为模式。本质上，社区事件预测的基本目标不是关注单个局部的节点，而是从全局的角度预测社区中是否会出现某种高频模式。比如，探索金融交易网络中的洗钱模式<sup>[182]</sup>，或者社群中的疾病传播模式<sup>[183]</sup>。

#### 4.4 本章小结

在本章中，我们看到了目前的动态图相关文献存在不足之处，即仅仅只使用时间信息增加空间关系建模的能力，而没有统一考虑时间和空间的联合关系建模，这从根本任务上就限制了动态图模型的表达能力。我们提出了一个新的动态图建



17010226

模任务——动态关系预测——以尝试解决这个问题，并提出了一个新颖的动态图模型来适应这个建模任务。具体地，给定源节点，我们希望模型可以直接预测该节点下一步交互的目标节点以及这个交互所发生的未来时间戳。进一步地，为了建模这三者的关系，我们假设在特征嵌入空间中，源节点向量、目标节点向量以及时间向量三者之间呈现一个闭合的三角关系，并且我们设计了适合这个训练目标的负采样机制。给定源节点，利用与其他节点之间的相对位置关系即可确认目标节点，再利用三角关系求得时间向量，最后再对时间向量的岭回归技术预测出具体的交互形式。广泛的实验结果表明了，根据此目标设计的动态图模型在动态关系预测任务中取得了更优异的性能表现。



17010226

## 第五章 基于层次化随机点过程的关系演化预测

### 5.1 引言

随着社会中的各种系统越来越复杂，人们越来越倾向于使用图数据来建模社交网络、交通网络、生化分子等蕴含复杂特征的数据<sup>[12]</sup>。图神经网络<sup>[10]</sup>是一个新兴的深度神经网络家族，它通过消息传递机制将图数据中的结构信息嵌入到有限维度的特征空间中，它被广泛应用于这种复杂的图系统的建模。最近，不少文献（详见综述文献<sup>[170]</sup>）开始考虑时间相关的图结构数据建模，因为在社交网络、通信网络、交通网络等现实系统中，图中节点之间的交互自然是跟时间相关的。动态图神经网络（Temporal Graph Neural Networks, TGNNs）<sup>[170]</sup>通过将时间信号的处理合并到消息传递过程中，将图神经网络的应用范围扩展到了动态图中。其中，Jodie<sup>[21]</sup>、TGN<sup>[24]</sup>和 TGAT<sup>[22]</sup>模型被设计用来在动态图中执行关系预测任务，即在给定时间戳  $t$  和源节点  $v^s$  的情况下，预测目标节点  $v^d$ ：建模一个条件概率分布  $p(v^d | \mathcal{G}_{t-}, v^s, t)$ 。然而他们的工作大部分都致力于将图上的时间信息融入到传统的静态图学习任务中，即旨在使用循环时间网络<sup>[173]</sup>、注意力网络<sup>[174]</sup>等等来编码时间信息以更准确地执行传统的静态图空间关系学习任务，并不是真的发挥了动态图的全部数据潜力。因为学习任务与时间无关的话，机器学习模型就不会有强烈动力来编码时间有关的特征表示。

在第四章中，我们将时间戳预测任务引入了传统静态的关系推理任务中，提出了2.3.2小节所述的动态关系推理任务。静态的关系推理任务是给定源节点  $v^s$ ，预测目标节点  $v^d$ ；而时间关系推理任务则是给定源节点  $v^s$ ，预测一个由目标节点和时间戳组成的二元组  $(v^d, t)$ 。然而，动态关系推理任务仅仅只能在给定某个节点  $v^s$  的情况下，预测该节点下一步的连接对象  $v^d$  以及时间戳  $t$ 。动态关系推理任务更加聚焦于针对某个节点（用户）的未来行为进行预测性的建模。但在很多时候，这种预测并不能覆盖所有的应用需求。

比如在有些场景中，比如针对某个单独的个体进行预测，人们对于图整体的演化更加感兴趣。一个典型的场景就是新冠疫情的传播网络，在这个场景中，人们的社交近似于微观世界中混沌的粒子，有些人可能会选择闭门不出，有些人可能会继续参与广泛的社交。对于这种几乎无法准确预测的行为个体，以个体为单位去预测他们的交互几乎失去了意义。比起对每一个个体不可控的行为进行预测，政府更感兴趣的是针对整个群体进行感染风险的预测。对于这种不考虑某个具体个体的关系，而考虑图上整体的关系演化的任务，人们将其定义为动态关系演化



17010226

任务<sup>[35]</sup>: 在无需给定源节点  $v^s$  的情况下, 直接预测由源节点  $v^s$ , 目标节点  $v^d$  以及时间戳  $t$  组成的三元组  $(v^s, v^d, t)$ 。预测三元组  $(v^s, v^d, t)$  本质上是一种动态图上的事件预测任务, 从交互事件中建模和学习是一个具有广泛应用的重要课题, 它预测一系列未来事件, 不仅是关于它们将涉及哪些实体, 还包括它们将在何时发生。

显而易见, 直接预测三元组  $(v^s, v^d, t)$  比给定源节点  $u$  预测二元组  $(v^d, t)$  要困难得多。假设图中的总节点数量为  $N$ , 并且我们先不考虑时间戳  $t$ , 如果已知源节点  $v^s$ , 从除节点  $v^s$  以外的全部节点中选择一个作为目标节点  $v^d$ , 则预测的所有可能性有共  $N - 1$  种; 然而, 在不事先确定源节点  $v^s$  的情况下, 从图上的所有节点中选择一对交互节点  $(v^s, v)$ , 则共有  $N \times (N - 1)$  种情况。为了解决模型解空间庞大的问题, 已经有几篇文献提出了引入随机点过程模型的解决方案, 如果模型很难在庞大的搜索空间中直接给出预测结果, 那么人们就假设未来的事件发生服从某个随机过程分布。模型只要去学习某个随机过程就可以了。然而, 这些方案都是在折衷的条件下解决问题的, 在图5-1中, 我将本章的方案与过去主流方案的对比进行了展示与说明。

Know-Evolve<sup>[27]</sup>和 DyRep<sup>[26]</sup>将 GNN 与多变量连续时间序列的方法相结合, 例如用于时间戳预测任务的时序点过程 (Temporal Point Process, TPP)。这类工作致力于预测下一个交互事件的时间, 但需要知道交互事件的两个对象, 即建模另一个条件概率分布  $p(t | G, v^s, v^d)$ 。然而, 如图5-1a所示, 这些模型<sup>[26,35-36]</sup>试图仅仅只使用单独的一个随机过程网络, 从纯全局的视角来描述整个图的演化, 忽略了图上不同的区域可能具有不同的密度、数量、交互频次等属性, 也可能具有不同模态的演化模式, 而这导致模型仅仅只能隐式地学习不同演化的模式区别, 带来了模型对于不同演化模式的适应能力不足问题。这种建模方法过于粗糙, 只考虑了图整体的模式, 没有考虑图的局部具有不同的性质。

如图5-1b所示, 另外一些工作试图将用于传统时间序列上的 TPP 方法拓展, 以加入额外的信息。值得注意的是, Marked Temporal Point Process (MTPP) 将每个事件与一个标记相关联, 并联合预测该标记以及未来事件的时间戳。Recurrent Marked Temporal Point Process (RMTPP)<sup>[33]</sup>提出使用递归神经网络来学习条件强度和标记分布。MTPP 和 RMTPP 能够通过将节点对  $(v^s, v^d)$  作为事件标记来预测图上的新事件。但是, 这会创建一个  $O(|\mathcal{V}|^2)$  的高复杂度概率分布空间, 使其无法扩展到具有大量节点的图上, 而且这种建模过于精细, 只考虑了局部信息, 忽略了图上局部与整体之间的联系。

由于图上的节点规模经常可以达到数亿级别, 模型很难从数亿节点中预测出



17010226

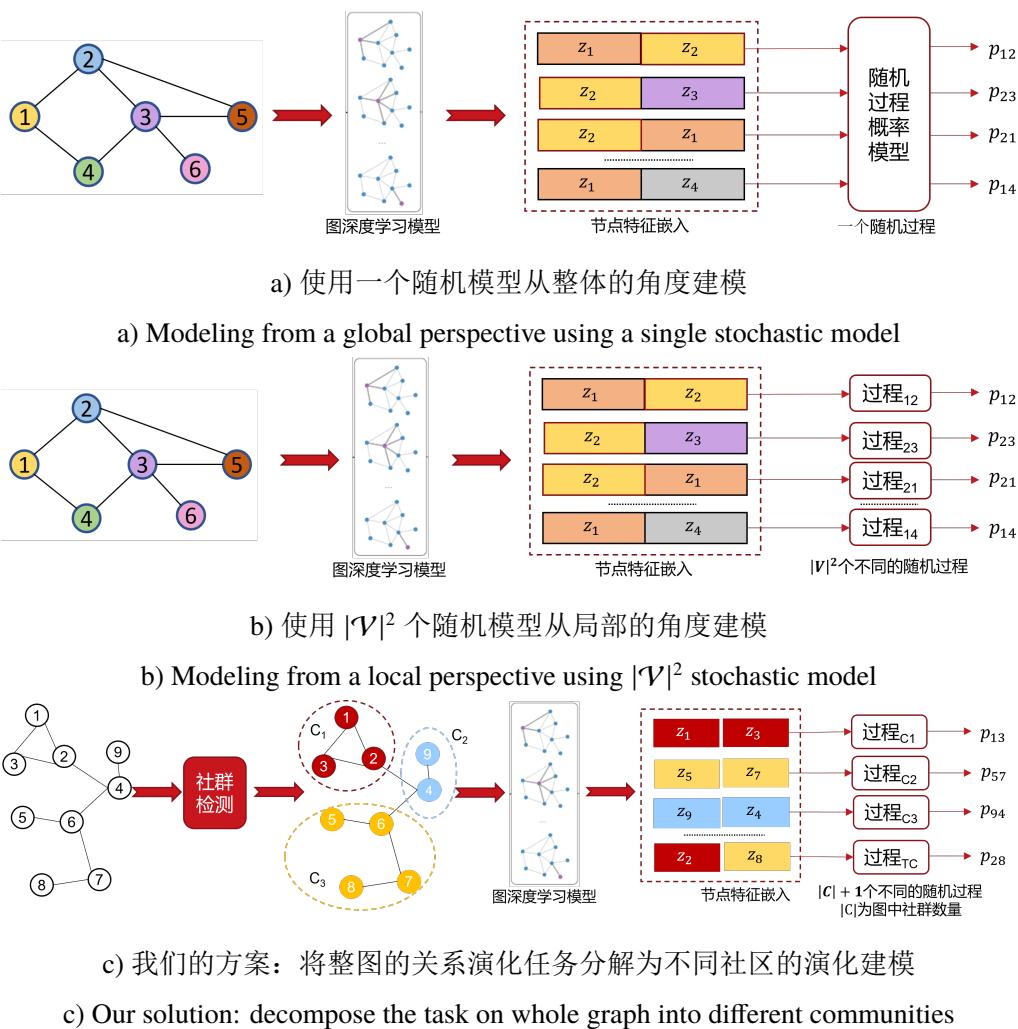


图 5-1 图 (c) 本章的方案与图 (a,b) 过去的方案的对比: 我们的方案同时考虑了局部和整体关系

Figure 5-1 Different schemes in the past literature

合理的三元组  $(v^s, v^d, t)$ , 因此对整体图的演化趋势进行预测非常困难, 而对局部进行建模却带来了较高的模型复杂度。在直接建模动态图上的关系演化非常困难的情况下, 我们给出的解决方案是“分群而治之”。在某些情况下, 一些实体, 例如具有密集连接的实体, 或具有类似特征的实体, 可能会存在社群的聚集特性或在演化中形成某些社群。我们认为, 图上的不同社群通常都会具有不同的演化模式, 比如消费水平不同的圈子在电商平台购买商品的行为通常是不同的; 属于不同行业圈内部的用户的通信行为可能也大相径庭。因此我们提出基于社群的预测, 在给定的动态图上, 我们先采用社群检测方法进行分群, 然后我们优先使用全局视角预测社群内部的交互行为演化, 再通过局部的视角考虑不同社群之间的交互。



17010226

即考虑了不同社群之间的差异性，也考虑了同一社群之间的相似性，同时融合了全局和局部的信息，增强了模型对于图上不同区域演化模式的适应能力。如图5-1c中所述，我们将一个困难的图关系演化问题转化成为数个易解的基于社群的演化预测子问题：针对不同的社群分别建立不同的演化预测模型。而对于社群与社群之间的交互关系，我们则使用一个随机过程 Transfer Community (TC) 来描述，通过这种方法，我们只需要使用  $|C| + 1$  个随机过程模型就可以完成对整个图的建模， $|C|$  是社群的数量。相比使用一个随机过程的全局方法和  $|\mathcal{V}|^2$  个随机过程的局部方法，我们基于社群的方法实现了更好的精细度与复杂度的折衷，因为  $|C| \ll |\mathcal{V}|$ 。总结来说，基于社群的动态图演化预测不但解决了原问题难以求解或者是复杂度过高的弊端，而且也可以增强模型对于图上不同数据演化模式的适应能力。

此外，研究某个社群内的未来事件演化对于研究动态图的演变至关重要，且更加符合实际需求。例如，社群关系演化预测技术可以为社区内部的流行病（比如新冠病毒）传播作提前干预<sup>[49]</sup>；也可以对某个交通拥堵区域的拥堵传播情况作监控和预测<sup>[47]</sup>，方便交警提前布置人手干预；如果模型预测某个社区的经济状况<sup>[50]</sup>或政治倾向<sup>[51]</sup>发生突然变化，则代表社区需要政府和相关部门更多的关注和帮助。已经有很多文献专注于在社群中挖掘某种模式或者预测社群的行为，比如在社群行为预测<sup>[184]</sup>、动态社群发现<sup>[185]</sup>以及社群异常检测<sup>[108]</sup>等应用场景中，人们只考虑在给定的候选节点集合中预测未来事件，即相比于整个图的所有节点来说，更加自然地，人们仅仅关注图上的一部分节点，这在经验上是合理的。

综合以上两点因素，我们决心研究如何解决动态图中的社群关系演化问题。类似于2.3小节的定义2.12中对于动态关系演化问题的定义，让我们将动态关系演化问题这个问题的子问题——动态社群关系演化进行数学上的规范化：给定动态图  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ ，边集  $\mathcal{E} = \{e_\tau : \tau = 1, \dots, T\}$  是有序集合，其中  $e_\tau = (v_\tau^s, v_\tau^d, t_\tau)$ 。在动态图  $\mathcal{G}$  中，给定待查询的节点集  $\mathcal{V}' \subset \mathcal{V}$  以及当前时间戳  $t$ ，则我们可以得到一个子图社群  $\mathcal{G}' = (\mathcal{V}', \mathcal{E}'), \mathcal{E}' \subset \mathcal{E}_{t-}$ 。社群关系演化预测的目的是，根据社群的历史信息  $\mathcal{G}'$  以及时间戳在  $t$  时刻之前的动态图  $\mathcal{G}_{t-} = (\mathcal{V}_{t-}, \mathcal{E}_{t-})$ ，综合局部的社群信息  $\mathcal{G}'$  以及全部的图信息  $\mathcal{G}_{t-}$ ，直接推理  $k$  个的未来事件  $\{e_i, i = 1, \dots, k\}$ ，其中  $e_i \notin \mathcal{E}_{t-}$ ，并且  $e_i$  是一个三元组  $(v_i^s, v_i^d, t_i)$ ，其中  $v_i^s, v_i^d \in \mathcal{V}'$  并且  $t_i > \max_{e_\tau \in \mathcal{E}'} t_\tau$ ，这个三元组代表的是社群内部的一次交互。

预测社群内  $k$  个未来的事件其实就相当于让模型学习一个条件概率分布：

$$p(e_i, i = 1, \dots, k | (\mathcal{G}, \mathcal{V}', t)) \quad (5-1)$$

其中每条边  $e_i$  的预测都是包含一个对于源节点、目标节点以及时间戳  $(v_i^s, v_i^d, t_i)$  的联合概率分布的估计。与传统的时间序列预测相比，动态社群事件预测必须联



17010226

合考虑由图表征的空间信息和以事件流为特征的时间信号，以便做出更准确的预测。此外，如图5-2所示，有颜色的节点和边代表用户所关心的社群部分，其他的灰色的节点和边代表动态图上的其他部分，是用户不关心的本任务希望模型不仅仅只根据社群子图  $\mathcal{G}'$  的信息预测未来的社群关系演化，而是更加鼓励模型也获取整个动态图  $\mathcal{G}$  上的信息来增强社群关系演化预测的精确度。

具体地，本文采用自回归预测（Autoregressive Prediction）的方式一步一步地预测社群的演进，以执行社群关系演化预测任务。自回归预测（Autoregressive Prediction），是统计学中一种处理时间序列的方法，用同一变量例如  $x$  的之前各个时间点，亦即  $x_1$  至  $x_{t-1}$  来预测本时间点  $x_t$  的表现。自回归预测是从机器学习的回归分析理论中发展而来，只是不用回归分析的范式“数据  $x$  预测标签  $y$ ”，而是用  $x$  预测  $x$ （自己），因此叫做自回归。自回归预测模型被广泛运用在经济学、资讯学、自然现象的预测上。

社群关系演化任务的目的是在给定历史数据（即图  $\mathcal{G}'$ ）以及感兴趣的节点集  $\mathcal{V}'$  的情况下，模型就自动生成新的边  $(v^s, v^d, t)$ ，并将这个  $(v^s, v^d, t)$  作为一条边添加到子图社群  $\mathcal{G}'$  中，迭代执行预测  $k$  次，这个演化预测的过程跟自回归预测非常相似，通过一步一步地生成新的边再更新社群的历史数据，模型就可以把社群上的演化模拟出来。然后再结合社群之间的连接建模，就可以进一步地进行全图上的演化事件预测。

虽然基于社群的关系演化预测方法解决了前人方法存在的全局-局部信息利用不全以及多模态适应能力不足的问题，但是它也带来了一些新的难点。由于社群之间存在关联（随机过程 TC 的引入），计算先后顺序会影响结果，所以我们的模型必须支持并行计算。然而过去几乎所有融合了时序点过程以及图结构学习的文献<sup>[26-32]</sup>都使用了基于时序点过程的 RNN 模型作为工具来建立随机过程。然而，RNN 模型因其循环结构（必须按顺序处理每个事件）无法进行并行训练，这给大规模图数据集上的训练带来了很大限制。我们的模型设计了基于时间核函数编码的纯粹注意力模型替代了 RNN 循环结构模型，破除了过去随机点过程图模型无法并行计算的限制。

本章所做出的贡献如下：

i) 我们将原本求解困难的图上的关系演化任务分解为更容易求解的考虑不同社群的关系演化预测任务，它联合预测社群中下一个事件的事件节点和时间戳，这比时态关系推理和时间戳预测都要困难得多。而且它解决了过去关系演化预测文献中对于全局-局部关系抽取能力不足以及对于不同模态适应能力不足的问题。

ii) 我们提出了基于层次化随机点过程的图神经网络模型（Community Event



17010226

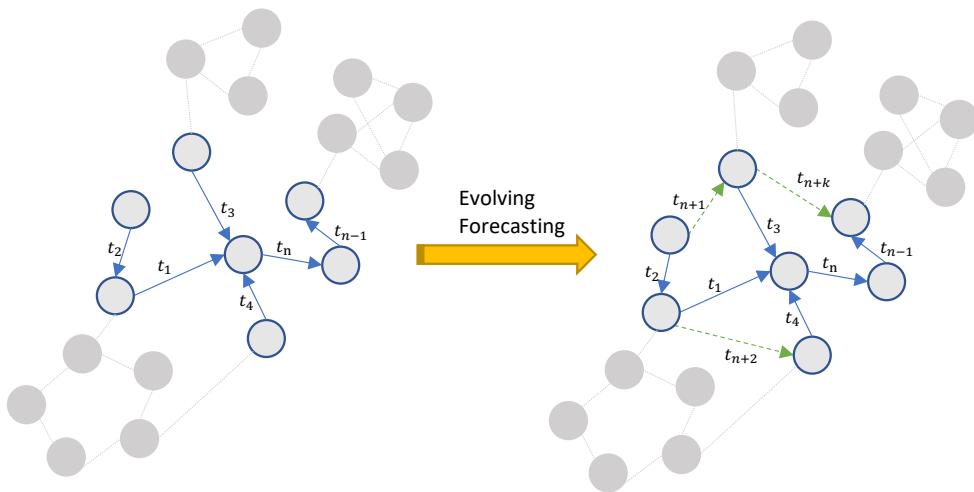


图 5-2 社群内部的关系演化预测

Figure 5-2 The proposed community evolution prediction task.

Predicting task with a graph Point Process model, **CEP3**)。它将 GNN 与 TPP 结合，聚合了空间和时间信息，可以联合预测某个事件相关的实体和发生的时间戳。为了扩展到大规模图，我们将图上事件的联合概率分布分解为三个条件概率分布的乘积，进一步降低了模型建模的复杂度。此外，我们提出了基于时间核编码的注意力模块来取代 TPP 模型的循环性神经网络结构，实现了在大规模数据上的并行随机小批量训练。

iii) 我们为社群关系演化预测任务提出了新的基准。具体来说，我们设计了新的评估指标来衡量实体和时间戳的预测质量。对于基线模型，我们对比了目前 5 种最先进模型。我们的评估表明，我们的 CEP3 模型在所有四个真实图数据集中都具有优越性。

## 5.2 层次化点过程图神经网络模型

为了解决在引言中提到的问题，我们提出了层次化点过程图神经网络 (Community Event Predicting task with a graph Point Process model, **CEP3**) 模型，如图5-3所示。给定历史的图信息  $\mathcal{G}_{t-}$  以及用户感兴趣的节点集  $\mathcal{V}'$ ，则我们可以得到感兴趣的图社群  $\mathcal{G}'$  为了预测接下来  $k$  个事件  $e_1, \dots, e_k$ ，我们首先利用作为编码器的动态图模型获得了每个节点的特征表示  $h_0^i$ ，从而获得了初始的图  $\tilde{\mathcal{G}}_0$ 。之后在第  $i$  步，我们预测  $e_i = (v_i^s, v_i^d, t_i)$ ，即源节点、目标节点以及时间戳。这个预测出的事件  $e_i$  之后会被加入到图  $\tilde{\mathcal{G}}_{i-1}$  并声称图  $\tilde{\mathcal{G}}_i$  跟踪我们到目前为止的预测。对于图社群中的每个节点来说，其隐空间状态表示  $h_i^{(i)}$  也会根据新的图  $\tilde{\mathcal{G}}_i$  而更新到



17010226

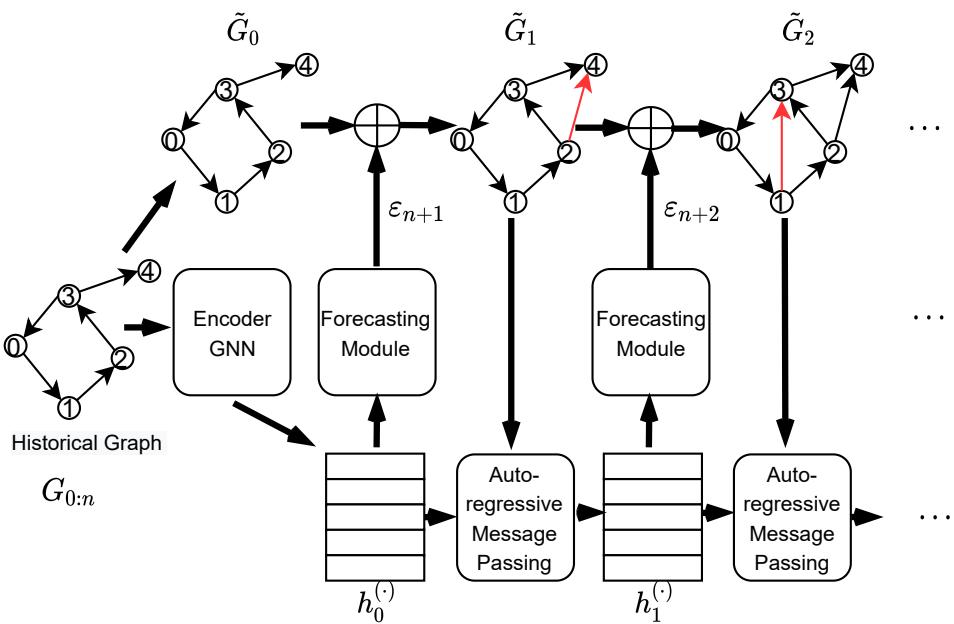
图 5-3 提出的 CEP3 模型的总体架构。红色箭头表示预测的未来事件  $e_i$ 。

Figure 5-3 The overall architecture of proposed CEP3 model. Red arrows represent the predicted future events  $e_i$ .

$h_{i-1}$ 。这个机制遵循这 RMTPP<sup>[33]</sup>的框架，除了有几个显著差异：**i)** 我们使用时间感知 GNN 初始化循环网络状态，这允许我们的循环模块在短得多的序列上遍历，而不会丢失历史信息，**ii)** 我们用 GNN 更新网络状态，以模拟由新事件引起的实体之间的拓扑依赖关系，**iii)** 我们通过分解联合概率分布来预测节点和时间戳。我们给出每个组件的具体细节如下。

### 5.2.1 时间核方法

作为一个动态图学习模型，有一个必须存在的重要模块就是如何将“时间”这个信息映射到可学习的目标向量，所以我们试图建立一个连续的映射函数  $\Phi : T \rightarrow \mathbb{R}^{d_T}$ ，将时间域的离散时间戳数值映射到连续且稠密的  $d_T$  维度的向量空间，其中  $T \subset \mathbb{R}^+$  是一个正实数值。而将一个低维的连续函数映射到高维的离散空间，最自然的想法就是引入泰勒级数展开或者傅里叶级数展开。然而泰勒级数要求函数必须处处可微，而时间戳数值是离散的，傅里叶级数则要求函数具有周期性而时间戳是单调增的。这导致了最自然的思路无法适用我们的问题。

关于如何定义时间核方法来生成有关于时间的向量表示，我们有两个基本考虑。第一，由于模型的目的是寻找事件与事件之间在时间上的关联性，所以我们



17010226

选用两个时间表示向量的内积  $\langle \Phi(t_1), \Phi(t_2) \rangle$ <sup>①</sup> 作为核函数的表示形式。第二，通常来说，两个事件的相对时间跨度  $|t_2 - t_1|$  才揭示了关键的、事件之间的时间信息关联，而不是两个事件的时间的绝对值  $t_2$  和  $t_1$ ，因此我们对于如何学习时间跨度的模式更加感兴趣。

仿照文献<sup>[186]</sup>，我们定义了一个时间核函数  $\mathcal{K} : T \times T \Rightarrow \mathcal{R}$ ，其中  $\mathcal{K}(t_1, t_2) := \langle \Phi(t_1), \Phi(t_2) \rangle$ ，并且  $\mathcal{K}(t_1, t_2) = \psi(t_1 - t_2)$ ,  $\forall t_1, t_2 \in T$ ,  $\psi$  是一个将时间域映射到实数域的可训练的映射函数。

还由于  $\mathcal{K}(t_1 + c, t_2 + c) = \psi(t_1 - t_2) = \mathcal{K}(t_1, t_2)$ ，这个时间核函数具有平移不变的性质，这对于模型学习并在大规模数据中泛化具有巨大的意义。

综上所述，我们需要做的就是定义好  $\Phi$  的参数化形式，以便进行有效的基于梯度的优化。最终解决方案由经典的谐波分析理论——博赫纳定理给出，因为时间核函数  $\mathcal{K}$  是半正定的且连续的，因为它是通过 Gram 矩阵定义的，并且映射  $\Phi$  是连续的。因此，上面定义的时间核函数  $\mathcal{K}$  满足博赫纳定理的假设，我们在下面陈述。

**定理 5.1 (博赫纳定理 (Bochner's Theorem)<sup>[53]</sup>)** 一个  $\mathcal{R}^d$  上定义的在连续的平移不变核函数  $\mathcal{K}(\mathbf{x}, \mathbf{y}) = \psi(\mathbf{x} - \mathbf{y})$  是正定的，当且仅当  $\mathcal{R}$  上存在一个非负概率测度  $p(\omega)$ ，使得  $\psi$  是该测度的傅里叶变换。

因此，如果对核函数进行适当的缩放，核函数  $\mathcal{K}$  也可以写成：

$$\mathcal{K}(t_1, t_2) = \psi(t_1 - t_2) = \int_{\mathbb{R}} e^{i\omega(t_1 - t_2)} p(\omega) d\omega = \mathbb{E}_{\omega} [\xi_{\omega}(t_1) \xi_{\omega}(t_2)^*] \quad (5-2)$$

其中  $\xi_{\omega}(t) = e^{i\omega t}$ 。由于核函数  $\mathcal{K}$  以及概率测度  $p(\omega)$  是实数值，我们把公式 5-2 中的实数部分提取出来，可以获得如下的表达形式：

$$\begin{aligned} \mathcal{K}(t_1, t_2) &= \mathbb{E}_{\omega} [\cos(\omega(t_1 - t_2))] \\ &= \mathbb{E}_{\omega} [\cos(\omega t_1) \cos(\omega t_2) + \sin(\omega t_1) \sin(\omega t_2)] \end{aligned} \quad (5-3)$$

上面的公式中的期望  $\mathbb{E}_{\omega}$  部分可以使用蒙特卡洛积分<sup>[187]</sup>近似估计，即：

$$\mathcal{K}(t_1, t_2) \approx \frac{1}{d} \sum_{i=1}^d \cos(\omega_i t_1) \cos(\omega_i t_2) + \sin(\omega_i t_1) \sin(\omega_i t_2) \quad (5-4)$$

其中， $\omega_1, \dots, \omega_d \stackrel{i.i.d.}{\sim} p(\omega)$ 。经过数学推导，最终核函数映射的表达式可以写成：

$$t \mapsto \Phi_d(t) := \sqrt{\frac{1}{d}} [\cos(\omega_1 t), \sin(\omega_1 t), \dots, \cos(\omega_d t), \sin(\omega_d t)] \quad (5-5)$$

① 向量的内积是判断两个向量相似度的最常用方法之一



17010226

而由于  $\cos$  和  $\sin$  函数在表达能力上相同，则我们考虑在公式中省略  $\sin$  的部分并且加入偏置项  $\mathbf{b}$  以进一步增加模型的表达能力，时间核函数的形式由此可变成：

$$\Phi(t) = \sqrt{\frac{1}{d}} \cos(\mathbf{w}t + \mathbf{b}) \quad (5-6)$$

### 5.2.2 基于动态图神经网络的特征编码器

动态图神经网络编码器是一个层数为  $L$  的神经网络编码器，其应该能够同时处理关系依赖项、时间戳和可选的边特征。因此，对于其任意一层  $l$ ，都有以下形式：

$$z_{(v)}^{(l)} = \mathcal{F} \left( \left\{ \left( z_{v^s}^{(l-1)}, z_{v^d}^{(l-1)}, \mathbf{f}^e, \Phi(\tau - t) \right) : (v^s, v^d, t, \mathbf{f}^e) \in \mathcal{G}^{(k)}(v_i; \tau) \right\} \right) \quad (5-7)$$

其中， $\mathcal{G}^{(k)}(v_i; \tau)$  代表节点  $v_i$  在时刻  $\tau$  之前的事件所形成的  $k$  阶子图， $\Phi(\cdot)$  是在5.2.1小节中描述的可训练的时间核映射函数，其将连续的时间戳映射到一个固定维度的向量空间。 $z_{(v)}^{(0)}$  是节点  $v$  的特征向量，而  $\mathcal{F}$  代表某种神经网络参数结构，它根据某个节点的时态子图来学习并更新节点特征表示。

在本文中，我们使用我们使用动态图神经网络将拓扑局部邻域内的所有历史事件，包括事件节点、时间戳和事件特征一起作为输入，同时使我们能够并行地在多个历史图上进行训练。特别是，我们使用了4.2小节中已经有过详细介绍的动态图注意层<sup>[22]</sup>（TGAT）进行编码，时态图注意模块是一种基于自注意的节点嵌入方法，具体公式如下：

$$\begin{aligned} \mathbf{z}_i^{(l)} &= MLP(\mathbf{z}_i^{(l-1)} || \tilde{\mathbf{z}}_i^{(l)}) \\ \tilde{\mathbf{z}}_i^{(l)} &= \text{MultiHeadAttn}^{(l)}(\mathbf{q}_i^{(l)}, \mathbf{K}_i^{(l)}, \mathbf{V}_i^{(l)}) \\ \mathbf{q}_i^{(l)} &= [\mathbf{z}_i^{(l-1)} || \Phi(0)] \\ \mathbf{K}_i^{(l)} &= \mathbf{V}_i^{(l)} = \mathbf{C}_i^{(l)} \\ \mathbf{C}_i^{(l)} &= [\mathbf{z}_j^{(l-1)} || \mathbf{f}_{i,j,t_j}^e || \Phi(t - t_j), j \in \mathcal{N}^{(k)}(v_i; t)] \\ \Phi(t) &= \sqrt{\frac{1}{d}} \cos(\mathbf{w}t + \mathbf{b}) \end{aligned} \quad (5-8)$$

其中，公式第二行的多头注意力的计算过程如下：

$$\tilde{\mathbf{z}}_i^{(l)} = \sum_{a=0}^n \text{SoftMax}\left(\frac{(W_Q^{(l)} \mathbf{q}_i^{(l)})(W_K^{(l)} \mathbf{K}_i^{(l)})}{\sqrt{d_K}}\right)(W_V^{(l)} \mathbf{V}_i^{(l)}) \quad (5-9)$$

这种基于图神经网络的编码器有多种选择。我们在实验中使用动态图注意力网络 TGAT<sup>[22]</sup>，因为我们相信节点的局部邻域足以预测其未来事件，并且可以对



17010226

其进行小批量训练。为了验证我们的选择，我们还探索了我们方法的一种变体，即通过在训练阶段同时使用注意力和基于 RNN 的记忆模块，组成 CEP3 的变体（命名为 **CEP3 w RNN**）来进行比较，该变体可以分别从拓扑局部性和递归的角度合并历史事件和时间感知信息。然而，RNN 在训练中占用了大量的内存和时间，这一点我们在本章的引言中描述过，同时我们会在 5.3.7 小节中用实验证明。

### 5.2.3 基于层次化概率链的点过程事件预测模块

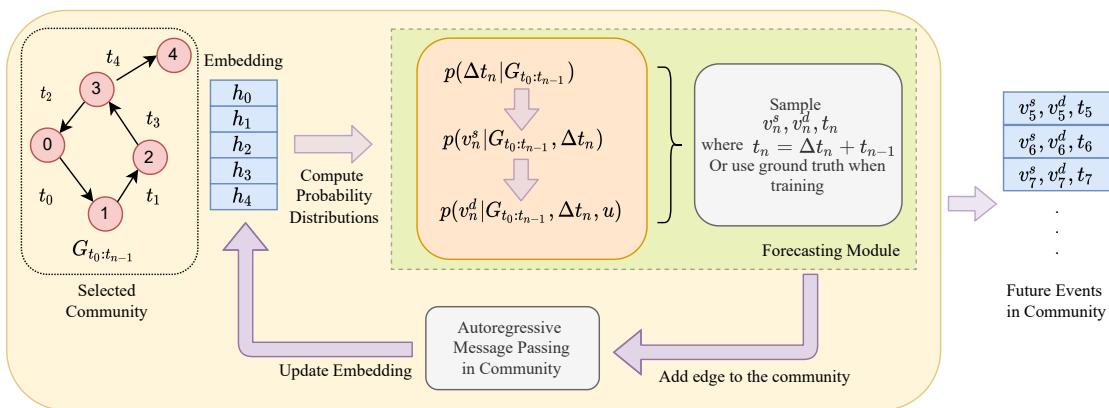


图 5-4 CEP3 模型的层次化预测模块及其与自回归消息传播模块的关系

Figure 5-4 HThe hierarchical probability-chain forecaster and its workflow relationship with the auto-regressive message passing module

图 5-4 展示了我们层次化概率化事件预报模块的详细信息。我们可以看到，预报模块仅根据所选候选节点对象（或社区）中的节点嵌入和历史连接来预测未来事件。这意味着我们不必担心大量的社区可能会减慢进程，因为 CEP3 模型可以在训练和推理过程中同时处理多个社区。分层概率链预测器及其与自回归消息传递模块的工作流关系。节点嵌入从第节 5.2.2 中描述的 GNN 编码器中学习。请注意，“所选社区”是指用于查询的候选节点的依赖于应用程序的集合。

根据 2.4.4.1 小节中描述的 MTPP 的叠加特性，我们通过首先预测事件发生的时间戳，然后预测源节点和目标节点，从而预测下一个事件  $e_i = (v_i^s, v_i^d, t_i)$ 。

利用 MTPP 的  $\lambda_{\text{total}} = \sum_{m \in \text{marks}} \lambda_m$  的叠加特性，MTPP 过程可以将所有标记合并到一个单独的过程中进行时间预测  $t \sim \text{TPP}(\lambda_{\text{total}})$ ，然后根据时间进行调整，标记预测问题将变成从以强度函数  $m \sim \text{Categorical}(\lambda_m)$  的值为特征的分类分布预测事件。这种特性使我们能够有效地将问题分解为数个单次过程，首先预测时间，然后选择最有可能发生的标记，而不是同时使用细化算法从复杂的联合分布中预测



17010226

时间和标记。由于我们可以很容易地从节点嵌入中获得候选集的总强度，如果我们需要预测下一个事件  $e_i = (v_i^s, v_i^d, t_i)$ ，可以通过将概率分布分解为数个条件概率的累乘：

$$p(u_i, v_i, t_i) = p(t_i) \times p(u_i | t_i) \times p(v_i | t_i, u_i) \quad (5-10)$$

这这意味着我们可以先预测时间戳，然后是源节点，最后是目标节点。这个类似于概率链条的建模过程如图5-4所示，其中，蓝色方框代表的节点特征嵌入由5.2.2小节中描述的特征编码器生成。

通过建模时间差值  $\Delta t_i = t_i - t_{i-1}$  的分布，我们首先预测  $t_i$ ：

$$\begin{aligned} \lambda_i^{(v)} &= \text{Softplus}(\text{MLP}_t(h_{i-1}^{(v)})) \\ \lambda_i &= \sum_v \lambda_i^{(v)} \\ \Delta t_{n+i} &\sim \text{Exponential}(\lambda_i) \\ t_{n+i} &= t_n + \Delta t_{n+i} \end{aligned} \quad (5-11)$$

其中 Softplus 是为了确保  $\lambda_i^{(v)}$  的非负性，并且同时保证了在  $\lambda_i^{(v)} < 0$  的情况下仍然存在可学习的梯度。 $\text{MLP}_t$  代表了用于学习时间相关的强度函数的多层感知机模型。由于我们假设  $\Delta t_{n+i}$  服从以  $\lambda_i$  为参数的指数分布，则我们使用该指数分布的均值  $\frac{1}{\lambda_i}$  作为训练时从概率模型采样出来的时间，这样可以保持训练时候的梯度更加稳定。而在预测时，时间戳会从该指数分布随机采样，而不是简单地使用均值。我们设置节点  $v$  上发生的下一个事件具有恒定的条件强度  $\lambda_i^{(v)}$ ，由节点  $v$  的当前循环隐藏状态  $\mathbf{h}_{i-1}^{(v)}$  输入神经网络计算得到。因此，在社群  $\mathcal{G}'$  中发生下一个事件的条件强度是社群中所有节点发生交互的条件强度函数  $\lambda_i^{(v)}, v \in \mathcal{V}'^{[188]}$  的总和，当然，其他假设也是可能的。

之后，我们需要估计下一个事件  $e_i$  中的源节点  $v_i^s \in \mathcal{V}'$ 。在给定时间戳  $t_i$  的情况下，选择社群中的某一个节点  $v$  作为源节点的条件类别概率分布  $p_{\text{src}}(v)$  由另外一个 MLP 模型进行参数化：

$$p_{\text{src}}(v) = \text{Softmax}(\text{MLP}_{\text{src}}(\mathbf{h}_{i-1}^v \| \phi(\Delta t_i))) \quad (5-12)$$

其中  $\|$  代表向量拼接操作， $\Delta t_i$  是公式5-11中预测的时间戳， $\phi$  函数跟我们在公式5-7中的形式一样。

之后，我们需要根据源节点和时间戳生成目标节点  $v_i^s$ ，我们采用了跟上面相似的结构。在给定时间戳  $t_i$  以及源节点  $v_i^s$  的情况下，任意一个节点  $v$  作为目标节点的条件类别概率分布  $p_{\text{dst}}(v)$  为：

$$p_{\text{dst}}(v) = \text{Softmax}(\text{MLP}_{\text{dst}}(\mathbf{h}_{i-1}^v \| \mathbf{h}_{i-1}^{v_i^s} \| \phi(\Delta t_i))) \quad (5-13)$$



17010226

这意味着，层次化点过程的推理过程遵循着一种贪婪（greedy）策略：先根据整体社群的状态决定下一个事件所发生的时间，我们首先选择概率最大的节点作为源节点，然后选择以所选源节点为条件的目标节点。值得注意的是，经过我们层次化点过程的设计，上面的两个预测公式只会生成具有  $|\mathcal{V}'|$  个可能元素的分布，而如果使用传统的点过程例如 RMTPP<sup>[33]</sup>的形式建模的话，则会生成  $|\mathcal{V}'|^2$  个元素复杂度的建模，其成本在大规模图社群的情况下是不可接受的。而为了验证这种贪婪设计的影响，我们还探索了我们方法的一种变体，我们直接估计源节点和目标节点的联合分布，生成一对双元组  $(u_i, v_i)$ ，这种策略会造成  $O(|\mathcal{V}'|^2)$  复杂度，但是有可能带来更准确的性能，我们期待在下一节的实验部分中验证它，我们将其命名为 **CEP3 w/o HRCHY**，其中 HRCHY 是英文单词 Hierarchical 的简写。

#### 5.2.4 基于信息传播的自回归模块

我们假设事件的发生将直接影响其事件节点的隐藏状态。此外，这种影响将沿着历史互动产生的链接传播到其他节点。因此，在生成新事件  $e_i$  之后，我们希望通过在图上传递新事件的消息来更新社群中其他节点的隐藏状态。我们通过维护另一个图  $\tilde{\mathcal{G}}_i$  来实现这一点，该图通过跟踪历史交互  $\mathcal{G}_{t-}$  和最新的预测事件  $e_i$  来实现更新。

具体来说，我们使用社群所有节点的特征来初始化  $\tilde{\mathcal{G}}_0$ 。每次预测一个新事件  $e_i$  时，我们将该事件添加回： $\tilde{\mathcal{G}}_i = \tilde{\mathcal{G}}_{i-1} \cup \{e_i\}$

之后，我们使用一个  $L$  层的信息传播网络更新社群中某个节点的隐藏状态，本章使的是用于空间传播的 GCN<sup>[73]</sup>和用于时间迭代更新的 GRU<sup>[189]</sup>组成信息传播模块：

$$\begin{aligned}\mathbf{w}_{i,0}^{(v)} &= \mathbf{h}_{i-1}^v \\ \mathbf{w}_{i,l}^{(v)} &= \text{MLP} \left( (1 + \epsilon) \mathbf{w}_{i,l-1}^{(v)} + \text{MAX} \left\{ \mathbf{w}_{i,l-1}^{(u)}, u \in \mathcal{N}_{\tilde{\mathcal{G}}}^{(1)}(v) \right\} \right) \\ \mathbf{h}_i^{(v)} &= \text{GRU} \left( \left[ \mathbf{w}_{i,L}^{(v)} \| \phi(\Delta t_{n+1}) \right], \mathbf{h}_{i-1}^{(v)} \right)\end{aligned}\quad (5-14)$$

其中， $\mathcal{N}_{\tilde{\mathcal{G}}}^{(1)}(v)$  是节点  $v$  在社群  $\tilde{\mathcal{G}}_i$  上的 1 阶邻居，GRU 是一种常见的循环神经网络架构，它最突出的特点是具有一个记忆存储向量用于存储过去的状态，GRU 的最终输出由记忆存储器与输入共同决定。

为了验证信息传播机制的必要性，我们还探索了一种变体，我们不使用自回归（Auto-Regression）模块更新社群中节点的隐藏状态，我们将此变体命名为 **CEP3 w/o AR**。



17010226

### 5.2.5 损失函数

上一小节中描述的预测模块输出的是社群中下一个事件  $e_i$  所发生的时间戳  $t_i$  以及所关联的节点  $v_i^s$  和  $v_i^d$ 。基于预测模块的输出，我们在训练中采用最大化对数似然（log likelihood）函数的方式进行优化，其损失函数的形式如下：

$$\mathcal{L}_{\text{time}} = \sum_{i=1}^K \underbrace{[-\log(\lambda_i) + \Delta t_{n+i} \lambda_i]}_{\text{time loss}} - \underbrace{\log p(u_{n+i}) - \log p(v_{n+i})}_{\text{entity loss}} \quad (5-15)$$

其中，损失函数求和中的前两项是来自公式5-11的对数生存概率，后两项是源节点和目标节点预测的对数概率。

## 5.3 对比实验与分析

### 5.3.1 数据集

在这一小节中，我们在四个公开的、现实世界动态图数据集上测试了所提出方法的性能和效率，这些数据集包括 Wikipedia、MOOC<sup>[21]</sup>、GitHub<sup>[26]</sup> 和 SocialEvo<sup>[180]</sup> 四个数据集。

**Wikipedia**<sup>①[21]</sup>，中文名可译为“维基百科”数据集，它被广泛用于动态的推荐系统的评估中，它是一个包含用户和维基百科页面作为节点类型，以用户对百科的编辑行为作为边的二部图（二部图的定义详见第二章中的定义2.7）。交互频率遵循长尾分布，一些节点具有相对较高的交互频率，而另一些节点只有少量交互。每个用户和每个项目的平均交互次数分别为 157 和 157。我们将每次编辑的文本转换为表示其 LIWC 类别<sup>[164]</sup>的边特征向量。

**MOOC**<sup>①[21]</sup>数据集即慕客平台数据集。慕课平台即大型开放式网络课程（Massive Open Online Courses），大多数慕课平台针对高等教育，并且像真正的大学一样，为在线的学生提供系统性的学习和课程管理系统。MOOC 数据集收集自中文慕课类型的网站“学堂在线（XuetangX）”，学堂在线是清华大学联合教育部在线教育研究中心于 2013 年 10 月发起建立的慕课平台，该数据集中包括学生对于 MOOC 课程的操作，比如观看某节课的视频、提交一份作业等等。该数据集由 7047 名学生组成，他们与 98 个项目（视频、答案等）进行互动，互动超过 411749 次。每个交互都与文献<sup>[21]</sup>中提供的特征向量相关联。

**Github**<sup>②[26]</sup>数据集来源于一个面向开源及私有软件项目的托管平台“Github”。数以百万计的开发者和公司在全球最大、最先进的开发平台 GitHub 上

① <http://snap.stanford.edu/jodie>

② <https://www.githubarchive.org>



17010226

构建、发布他们的软件并且不断维护它们，包括编写说明文档，修复和提交 bug 等等。Github 数据集是一个基于 GitHub 用户活动构建的社交网络，其中所有节点都是真实的 GitHub 用户，交互代表用户对另一方存储库的操作，如包括监视（Watch）、点赞（Star）、拷贝（Fork）、推送（Push）、创建问题（Creating Issue）、对问题进行评论（Commenting on Issue）、代码合并请求（Pull Request）、提交（Commit）等。

**SocialEvo**<sup>①</sup>[26,180]数据集是一个小型的社交网络数据，其由麻省理工学院（MIT）人类动态实验室（Human Dynamics Lab）收集。该数据集拥有 83 个个体和大约 6 万次交互，显而易见的是，该数据集历史交互次数更多，相比其他数据集有更丰富的历史信息。

由于公共数据集 SocialEvo 和 Github 没有边特征，我们使用以下属性生成 10 维边特征，包括边的两个事件节点的当前度数，以及两个事件节点的当前时间戳和最后更新的时间戳之间的时间差。请注意，时差分别以天数、小时数、分钟数和秒数表示。表格 5-1 显示了我们实验中使用的数据集的统计信息。

### 5.3.2 基线方法

在这一小节中，我们介绍了跟我们 CEP3 方法相对比的基线方法。除了在第 5.2 小节中所述的 CEP3、CEP3 w RNN、CEP3 w/o HRCHY 和 CEP3 w/o AR 方法以外，我们也跟其他常用方法进行了对比：时间序列模型 GRU<sup>[189]</sup>、泊松时序点过程（TPP-Poisson）、霍克斯时序点过程（TPP-Hawkes）<sup>[181]</sup>、循环标记时序点过程（RMTPP）<sup>[33]</sup>以及它在我们的任务中应用所产生的变体 RMTPP w HRCHY，该变体具有跟本章公式 5-12 和 5-13 相同的两级层次化概率分解。最后，我们还对比了 DyRep 模型<sup>[26]</sup>以及 DyRep 的自回归（Auto-Regression）更新版本 DyRep w AR。值得注意的是，RMTPP 通常是 MTPP 的最先进模型，而 DyRep 在动态图学习方面是最先进的。

我们在表格 5-2 中展示了我们介绍的所有基线模型所提供的能力，有些模型无法达到我们所期望的能力，有些模型可以通过一些非平凡的修改（用 \* 来表示）达到其中的某些要求。同时我们也渴望我们提出的模型能够实现小批量培训。如 5.2 一节开头所述，我们的 CEP3 分别利用层次 TPP 和基于 GNN 的更新模块实现大规模和并行训练。而正如 2.4.4.2 所提到的那样，RNN 的使用阻止了模型的并行训练。我们提出的模型 CEP3 满足所有期望的特性，并且将大部分算法所需的复杂度从  $O(|\mathcal{V}^2|)$  降至  $O(|\mathcal{V}|)$ 。接下来，我们将详细描述基线方法的选择。

① <http://realitycommons.media.mit.edu/socialevolution4.html>



17010226

Level	Statistics	Wikipedia	MOOC	Github	SocialEvo
整图级别 Graph level	边数 (Edges)	157,474	411,749	20,726	62,009
	节点数 (Nodes)	9,227	7,145	282	83
	最大度数 (Max Degree)	1,937	19,474	4,790	15,356
	平均度数 (Aver. Degree)	34	115	147	1,310
	边特征维数 (Edge Feat. Dim.)	172	4	10	10
	是否二部图 (Is Bipartite)	True	True	False	False
	时间区间 (Timespan)	31days	30days	1years	74days
	事件频率 (Edges/hour)	211.66	576.30	2.36	7.79
数据集分割方式 (Data Spilt)		70%-15%-15% by timestamp order			
社群级别 Community level	社群数量 (Communities)	142	25	17	10
	最大节点数 (Max Nodes)	396	990	46	18
	平均节点数 (Aver. Nodes)	50.27	264.96	15.94	7.7
	最大边数 (Max Edges)	4799	11686	3221	12199
	平均边数 (Aver. Edges)	778.28	2560.00	534.71	3420.90
	最小边数 (Min Edges)	77	16	34	863
	最大事件频率 (Max Edges/hour)	6.47	33.33	0.36	1.99
	平均事件频率 (Aver. Edges/hour)	1.11	5.36	0.06	0.56
最小事件频率 (Min Edges/hour)		0.14	0.34	0.01	0.15

表 5-1 四个数据集的统计信息

Table 5-1 Statistics of the datasets used in our experiments.

Taxonomy	GNN+TPP		RNN+TPP	GNN		TPP
Methods	CEP3	DyRep	RMTPP	TGAT	Poisson	Hawkes
推理关系 ( $v^s, v^d$ )	√	√	√	√	√	√
推理时间 $t$	√	√	√		√	√
关系-时间联合推理 ( $v^s, v^d, t$ )	√		√		√	√
显式建模图结构	√	√		√		
复杂度	$O( V )$	$O( V ^2)$	$O( V ^2)$	$O( V ^2)$	$O( V ^2)$	$O( V ^2)$
支持并行计算	√			√	√	
序列建模所用的模型	Attention	RNN+Attention	RNN	Attention	Poisson Process	Hawkes Process

表 5-2 各类模型的能力比较

Table 5-2 Comparison of model capabilities.

**时间序列方法:** 对于序列预测的基线模型, 我们建立了一个 RNN 模型, 即门控循环单元 (Gated Recurrent Unit, GRU)。模型给每个源节点和目标节点都设有一



17010226

个隐藏状态，模型的输出将被预测为时间均值和方差以及每个类相互作用的概率，时间将被表示为高斯分布，信源和目的地节点将被表示为分类分布。GRU 模型以 Seq2seq<sup>[190]</sup>的方式接收社群中的历史事件序列并输出该社群的未来  $K$  个事件。时间预测的损失项为均方误差，源和目的地预测的损失项为负对数似然。该公式迫使 GRU 仅根据 RNN 中的隐藏状态来预测即将发生的事件的时间戳，而其他基线将 TPP 函数作为随机概率过程进行调整，从而获得更好的建模能力。

**随机时序点过程方法:** 跟进文献<sup>[33]</sup>的实验基线模型设置，我们使用传统的时序点过程模型和基于深度学习的时序点过程模型作为我们 CEP3 模型的基线。这类模型是应用在传统的事件序列上的随机点过程模型，这类模型建模的是某个事件  $y$  在时间  $t$  到  $t + dt$  之间所发生的条件概率，考虑的是事件类型  $y_i$  与发生时间  $t_i$  之间的关系。而这类模型在动态图上的直接应用就是把动态图上所有交互节点对  $(v^s, v^d)$  视为一个独立的事件，则此种方法所建模的事件个数就等于动态图上的节点数的平方  $|\mathcal{V}|^2$ ，这平方复杂度的建模毫无疑问限制了其在大规模图上的应用，但是我们仍然试图了解其在动态关系预测上的能力。这类基线方法中，我们主要考虑：

- **TPP-Poisson:** 我们假设在每个节点对  $(v^s, v^d)$  处发生的事件遵循泊松过程，具有恒定的强度值  $\lambda(v^s, v^d)$ ，通过最大似然估计（Maximum Likelihood Estimation, MLE）从数据中学习。
- **TPP-Hawkes<sup>[181]</sup>:** 我们假设在每个节点对  $(v^s, v^d)$  处发生的事件遵循霍克斯过程<sup>[130]</sup>，其基本强度值为  $\mu_{v^s, v^d}$  和激励参数  $\alpha_{v^s, v^d}$  通过 MLE 从数据中学习。
- **RMTPP<sup>[33]</sup>:** 我们直接考虑每个源节点和目标节点对作为唯一的标记。我们注意到，这将消耗非常巨量的内存和时间，因为 RMTPP 将为每个节点对分配一个可学习的嵌入，结果是 RMTPP 模型将会拥有  $|\mathcal{V}|^2$  数量级的参数量，这对于计算系统来说太过于昂贵。
- **RMTPP w HRCHY:** 我们也考虑了 RMTPP 的变体，在这里我们用跟我们 CEP3 模型类似的层次化预测方式替换 RMTPP 基线中的（源节点-目标节点）预测：首先选择源节点，然后在源节点上选择目的节点。后一个公式也可以用作消融研究，以证明考虑图结构所带来的优势。
- **DyRep<sup>[26]</sup>:** 据我们所知，DyRep 是将时序点过程与图学习技术结合起来，同时对时间和空间依赖性进行建模的文献中最流行的。由于原始的 DyRep 公式只处理静态链路预测和时间预测，而不处理自回归的社群关系演化预测，因此我们使用 DyRep 为每个节点对计算强度值  $\lambda_{v^s, v^d}$ ，然后假设每个节点



17010226

对未来发生的概率满足泊松过程。

- **DyRep w AR:** 我们对 DyRep 的原始形式进行了一次简单的修改，即在向社群图中添加新边后更新所涉及的源节点和目标节点，该更新所使用的模型跟 DyRep 本身的更新模型共用参数。该基线模型旨在证明自回归的预测方式能够更好地将一个新的预测事件信息传播给图社群，从而影响邻居节点。

### 5.3.3 实验细节

在本章中，我们汇报了 CEP3 模型和其他基线模型的实现细节，包括网络的详细结构、超参数的选择以及实验所运行的环境。在软件环境方面，我们使用 PyTorch<sup>[148]</sup>学习框架以及 Deep Graph Library<sup>[43]</sup>来实现我们所有的模型，在硬件条件方面，我们在 Intel(R) Xeon(R) Platinum 8375C CPU @ 2.90GHz、96GB 内存的 Linux 系统机器上训练我们的模型和基线模型。对于所有基线模型，我们采用 AdamW<sup>[178]</sup>随机梯度优化器，学习率为 0.0001，训练、验证和测试时我们使用的批量大小均为 200，Dropout 比率为 0.1，并且使用了早停策略<sup>[165]</sup>，早停耐心值为 5。对于基线模型的其他参数，我们采用跟它们原始论文相同的参数设置。对于 TERRINE 模型来说，网络节点和时间嵌入的特征维度设为 64，动态图注意力层中的注意力头的数量设置为 4，图神经网络的消息传递层数为 2。对于动态图注意力层和时间映射函数中的 MLP 网络，我们采用了隐藏大小为 128（两倍的嵌入特征维度）的两层前馈神经网络。我们还将所有的超参数在表格 5-3 中展示，方便读者参考和复现。

### 5.3.4 评估指标

我们工作的主要贡献之一是提出了社群的演化预测，这是一项与动态图相关的新任务。据我们所知，我们是第一个能够同时对时间和结构信息进行事件预测的工作。给定历史社群图的节点嵌入以及用户关心的节点候选集，我们从时序点过程模型中自动回归采样，在候选节点集内构造链接，候选节点集可以任意选取，但是为了评估方便，我们使用经典社群检测算法 Louvain<sup>[52]</sup>分割产生的社群作为候选节点集。

Louvain 社群检测算法是一种提取网络社群结构的简单方法，这是一种基于模块化优化的启发式方法。该算法分两步工作。在第一步中，它将每个节点分配到自己的独立社群中（即每个节点单独初始化为一个社群），然后通过将每个节点移动到其所有邻居社群中，尝试为每个节点找到最大的正模块化增益。如果没有获得正增益，节点将保留在其原始社群中。给定一个图  $G = (\mathcal{V}, \mathcal{E})$ ，通过将孤立节



17010226

参数名	取值
编码器中的向量维度	32
预测模块中的向量维度	32
时间编码的向量维度	32
MLP 层数	3
采样邻居个数	15
图神经网络层数	2
学习率	0.001
优化器	AdamW
注意力头个数	4
循环神经网络模型	GRU
训练轮次	100
预测步长	200

表 5-3 CEP3 模型的参数配置

Table 5-3 Parameter configuration of CEP3 model

点  $v_i$  移动到社群  $C$  中获得的模块化增益可以通过以下公式（结合<sup>[52,191]</sup>）轻松计算：

$$\Delta Q = \frac{k_{i,in}}{2|\mathcal{V}|} - \gamma \frac{\Sigma_{tot} \cdot k_i}{2|\mathcal{V}|^2} \quad (5-16)$$

其中  $k_{i,in}$  是从节点  $v_i$  到  $C$  中节点的连接的权重之和， $k_i$  是所有连接到节点  $v_i$  的连边的权重之和， $\Sigma_{tot}$  是所有连接到  $C$  中节点的连边权重之和， $\gamma$  是分辨率参数。第一阶段会一直不断进行，直到没有任何单独的移动可以提供正模块化增益。

在第二步中，算法抛弃原来的图，建立一个新的图，其节点是第一阶段中发现的社群。为此，新节点之间链路的权重由相应两个社群中节点之间链路的权重之和给出。一旦这个阶段完成，就可以重新应用第一阶段，创建更大的社群，增加模块化增益。交替执行上述两个阶段，直到没有正模块化增益（或正模块化增益小于阈值）则停止。

根据社群检测算法 Louvain 确定图数据集上的社群之后，为了检验模型预测遥远未来的能力，我们对每个模型社群中预测未来  $K$  个事件跨度的性能， $K$  的取值由该社群在图中的未来事件数量决定。我们提出了专门适用于社群预测的评价指标，对于每个社群，在源节点-目标节点的预测方面，我们测量困惑度 (Perplexity,  $PPL$ )<sup>[192]</sup>；在时间戳预测方面，我们用平均绝对误差 (Mean Absolute Error,  $MAE$ ) 汇报结果。在交通流预测问题中<sup>[47]</sup>中也可以看到类似于我们这类对多个时间步跨



17010226

度的预测模型进行评估的想法。接下来我们，分别对 *PPL* 和 *MAE* 这两个指标进行解释。

首先是困惑度指标<sup>[192]</sup>，困惑度是信息论中的一种概念，困惑度可以度量某个概率分布或概率模型的预测结果与真实样本的契合程度，困惑度越低则契合越准确。在自然语言处理领域中，建立人类的语言模型是最主要的研究目标之一，语言模型通俗易懂的讲就是判断一句话是否符合自然语言规律的模型。困惑度指标被广泛用于在自然语言处理任务中来评价一个语言模型的好坏，即评价由该语言模型生成的句子有多满足真实的人类语言样本。我们把困惑度的概念引入到图事件预测中，将图上的节点序列先后相关性理解成为文本中的上下文。对比而言，语言模型的训练目的是预测下一个词语，而图预测模型的目标是预测下一个发生交互的节点，两者本来就有异曲同工之妙。其唯一的区别是，语言模型是从人类所有的词库中选择一个词，而图预测模型是从历史的候选节点集中挑选一个节点作为输出。

具体地，假设我们有来自真实样本的图事件序列  $(v_i^s, v_i^d, t_i)$ ，以及模型预测的事件序列  $(\hat{v}_i^s, \hat{v}_i^d, \hat{t}_i)$ ，其中  $i = 1, \dots, T$ 。对于这个长度为  $K$  的序列，其 *PPL* 的计算方法如下：

$$PPL = \exp \left( - \sum_{i=1}^K [\log p(u_i) + \log p(v_i | u_i)] \right) \quad (5-17)$$

类似于文献<sup>[135]</sup>，我们根据每个社群序列的长度和时间跨度对 *MAE* 进行了归一化处理，其的计算方法如下所述，：

$$MAE = \frac{1}{(t_T - t_0)K} \sum_{i=1}^K [|t_i - \min(t_K, \hat{t}_i)|] \quad (5-18)$$

*PPL* 和 *MAE* 这两个指标的值越小，表明模型性能越好。为了使 *MAE* 指标在不同的数据集中具有可比性，*MAE* 除以最大时间（max timespan） $t_K - t_0$  和序列长度  $K$ 。我们将所有社区的平均 *PPL* 报告为某个数据集的最终 *PPL*，*MAE* 的也同样计算其在每个社群内部的平均。两个指标的值越小，表示模型性能越好。

### 5.3.5 结果分析

表格5-4展示了比较社群关系演化预测任务中 CEP3 以及其他基线模型的 *PPL* 和 *MAE* 的平均和标准差。*MAE* 和 *PPL* 越小，模型的性能越好。最优秀的模型性能我们用加粗字体表示，第二名的模型我们用下划线来表示。

从结果分析中，我们可以看到，在 *MAE* 和 *PPL* 两种指标下，与不同数据集中的其他基线相比，CEP3 具有明显的优势。GRU 和 RMTPP 之间的 *MAE* 差异表



17010226

Datasets	Wikipedia			Github			MOOC			SocialEvo		Rank
Metric	Perplexity	MAE	Perplexity	MAE	Perplexity	MAE	Perplexity	MAE				Rank
GRU+Gaussian	131.06 ± 11.27	54.54 ± 1.19	68.53 ± 1.18	59.05 ± 1.72	457.40 ± 6.25	36.49 ± 2.01	33.85 ± 0.27	131.71 ± 7.09	8.00			
Hawkes	108.00 ± 3.73	56.84 ± 0.31	74.40 ± 2.47	55.21 ± 0.12	502.31 ± 12.30	36.67 ± 0.29	45.33 ± 5.35	139.35 ± 0.17	9.50			
Poisson	119.19 ± 1.11	56.70 ± 0.11	61.49 ± 0.96	55.21 ± 0.31	438.61 ± 7.05	36.61 ± 0.78	40.48 ± 1.99	139.3 ± 1.15	8.25			
RMTPP w HRCHY	133.68 ± 2.31	34.15 ± 0.89	62.19 ± 0.88	55.05 ± 1.02	616.79 ± 25.74	32.29 ± 1.59	41.37 ± 6.55	140.02 ± 2.06	8.88			
RMTPP	121.67 ± 1.01	32.91 ± 1.90	67.97 ± 1.02	54.79 ± 0.47	664.07 ± 11.05	32.83 ± 2.40	37.05 ± 0.77	138.9 ± 2.30	8.00			
DyRep w AR	116.07 ± 4.98	<u>28.74 ± 0.37</u>	54.57 ± 1.82	<u>28.46 ± 0.65</u>	431.18 ± 1.18	29.92 ± 1.48	29.6 ± 1.93	99.96 ± 6.18	3.38			
DyRep	119.13 ± 1.02	30.04 ± 0.14	64.05 ± 0.78	36.97 ± 1.74	438.61 ± 9.28	<b>13.41 ± 1.42</b>	36.59 ± 3.02	103.01 ± 3.49	4.75			
CEP3 w RNN	<u>104.87 ± 8.70</u>	41.94 ± 1.89	60.18 ± 1.04	39.22 ± 2.93	<u>374.77 ± 24.59</u>	20.09 ± 0.33	<u>30.37 ± 4.56</u>	95.12 ± 2.25	3.88			
CEP3 w/o HRCHY	<b>98.98 ± 7.61</b>	<b>28.69 ± 0.70</b>	<u>52.04 ± 3.33</u>	<b>26.8 ± 0.89</b>	<b>365.68 ± 28.01</b>	31.87 ± 0.18	<b>28.66 ± 2.74</b>	<b>79.58 ± 5.39</b>	<b>1.75</b>			
CEP3 w/o AR	125.51 ± 7.64	39.31 ± 2.59	61.03 ± 1.03	34.03 ± 0.37	448.37 ± 4.34	21.4 ± 0.47	38.59 ± 1.02	95.21 ± 4.44	6.13			
CEP3	118.82 ± 4.30	32.41 ± 0.58	<b>50.42 ± 0.70</b>	30.93 ± 1.67	401.64 ± 7.06	<u>17.69 ± 2.68</u>	36.8 ± 1.00	<u>94.54 ± 7.31</u>	<u>3.25</u>			

表 5–4 CEP3 以及其他基线模型在社群关系演化预测任务上的对比结果分析

Table 5–4 Comparative analysis of CEP3 and other baseline models in community evolution prediction task

明了时序点过程的引入在预测时间戳方面的有效性。将 CEP3 与基于序列的 TPP 模型 RMTPP 进行比较，我们可以看到，使用动态图神经网络捕捉历史交互信息可以提高预测性能。此外，当比较 DyRep w AR 和 DyRep 以及 CEP3 w/o AR 与纯 CEP3 时，我们可以得出结论，自回归更新可以更好地捕捉新预测事件对于整个待预测社群的影响。此外，我们的模型比 DyRep w AR 性能更好，因为在我们的 CEP3 中，在自回归更新期间，新预测的事件不仅影响事件中涉及的节点，而且还通过消息传递传播到其他节点。

### 5.3.6 异常社群的可视化分析

从上到下，图5–5分别在 Github、Wikipedia 和 MOOC 数据集的中以固定的圆形排布可视化某个存在异常的社区的连接模式。存在异常的社群通常的一大特点就是社群内部有着明显的高频用户以及低频用户的区分，而这些高频用户通常都是伪装成普通用户的机器人。我们绘制了真实样本、CEP3 模型预测和 DyRep 的预测以进行比较。绘制的节点大小表示其度数 (Degree)，而边的颜色表示连接频率，颜色越深，频率越高。我们希望同时在空间和时间维度展示模型的预测能力，即希望模型在一定的时间窗口内可以准确预测社群内节点的交互模式。请注意，边颜色和节点大小只能在同一行中进行比较，不能在纵向比较。

生成可视化图的步骤如下：我们首先应用在5.2.3小节中训练好的概率化预测模块，使用蒙特卡罗抽样方法预测社区中的边，此抽样过程重复三次。然后，我们按照边的概率大小排序，只保留最高的 33% 的概率边，并获得最终生成的图。多次生成然后丢弃可能性较小的边可以减少每次生成的图中的不确定性。



17010226

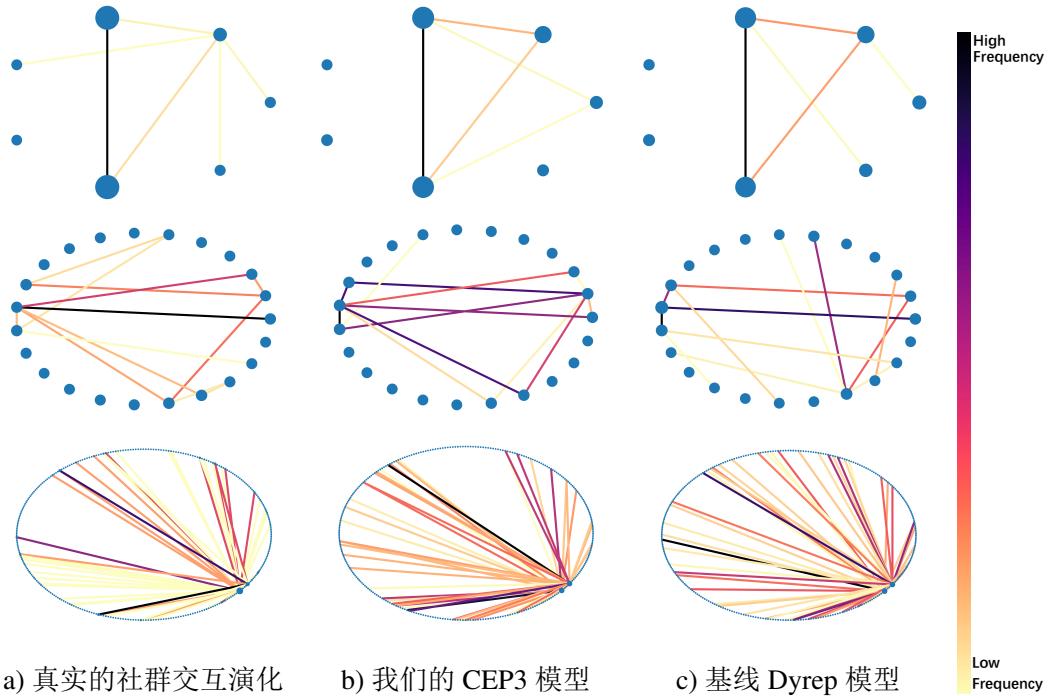


图 5-5 在测试阶段的整个时间跨度内，几个特定社区的预测可视化。

Figure 5-5 The prediction visualizations of certain communities in the whole timespan of the test phase.

下面我们分析一下可视化的结果。在第一行中，CEP3 和 DyRep 都捕获了这个小社区中的三角形连接。然而，真实样本中三角形颜色较浅的，这意味着 DyRep 在其预测中加强了这种联系。在第二行中，CEP3 的预测结果与真实值更相似，而 DyRep 生成了在原始图中不存在的高频紫色边。在第三行中，CEP3 成功地学习了真实样本中的两条黑色边，但 DyRep 预测出了两条以上的较深颜色的链接。

我们可以看到，我们的方法成功地识别了具有高频交互模式的节点，捕获了一些动态交互模式以及社区的演化动力学模式。本质上，社区事件预测的基本目标不是关注单个局部的节点，而是从全局的角度预测社区中是否会出现某种高频模式。比如，探索金融交易网络中的洗钱模式<sup>[182]</sup>，或者社群中的疾病传播模式<sup>[183]</sup>。

### 5.3.7 消融实验

在第5.2节中，我们提到了三种变体：消融模型 **CEP3 w RNN** 用牺牲了并行训练特性的代价来使用内存模块建模长期依赖性，消融模型 **CEP3 w/o HRCHY** 的提出是为了比较层次化预测与非层次化预测的性能区别，以及消融模型 **CEP3 w/o AR** 是为了比较在预测时是否加入自回归的节点嵌入更新所带来的性能差异。表



17010226

格5–4中也列出了这些变体之间的性能比较。

**CEP3 w/o HRCHY**。由于是我们最先提出了层次化的预测流程，那么其实如果需要比较模型的运行效率，只需要比较层次化和非层次化预测模型的运行时间和内存占用即可。为了证明所提出的 CEP3 层次化（Hierarchy, HRCHY）概率链的有效性，我们在图5–6中展示了 1000 步的推理时间开销与节点规模的关系。在这个散点图中，每个数据点代表 Wikipedia 数据集中的一个社区。我们通过实验证明了，与不含层次化的 CEP3 相比，CEP3 在大型社区更有效，没有层次化结构的 CEP3 模型具有较慢的训练和推理速度。这是因为计算每个可能的点对的强度函数将花费更多的时间。我们通过将节点对的预测问题分解为两个节点预测子问题来缓解这一问题，特别是在大规模社区中，CEP3 提出的层次化预测机制可以更快地解决进行预测。在理论上，传统的基于非层次化的点过程，是直接讲两个节点对的历史数据进行预测，这会导致  $O(|V|^2)$  的复杂度，而我们的层次化预测则是先预测源节点再预测目标节点，成功地将复杂度降低到了  $O(|V|)$ 。这证明了我们提出了层次化预测功能可以在大规模数据上很好地拓展。

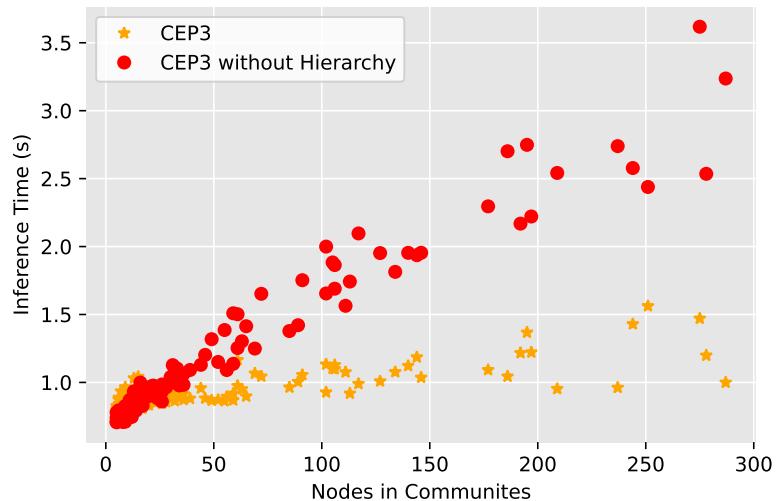


图 5–6 层次化 CEP3 与非层次化 CEP3 在预测阶段的运行时间实验

Figure 5–6 Running time experiment of hierarchical CEP3 and non hierarchical CEP3 in prediction stage

**CEP3 w/o AR**。如果在预测社群的未来事件时不在社群内部广播信息并更新各节点的特征嵌入，则 CEP3 模型性能会显著下降，并产生了与 DyRep w AR 相似结论果。在几乎所有的预测模型中，预测步长都是一个重要的参数。为了进一步研究预测步数和 AR 模块对 CEP3 模型的影响，我们对不同的预测步数进行了实验。MAE 越小，模型越好。从图5–7中，我们可以看到 AR 预测在不同数量的预



17010226

测步长中并不是都有帮助。当“预测步长”的数量很小时（例如 10 个），在理论上 CEP3 可以只使用时间为  $t_-$ ，即上个时刻的初始节点嵌入来预测事件。当预测步长的设置变大时，AR 的系统累积误差逐渐累积，导致 MAE 精度较低。随着步骤数的增加，初始节点嵌入在遥远的未来事件中几乎没有影响。这表明，如果没有 AR，CEP3 模型可能难以准确预测长期事件。

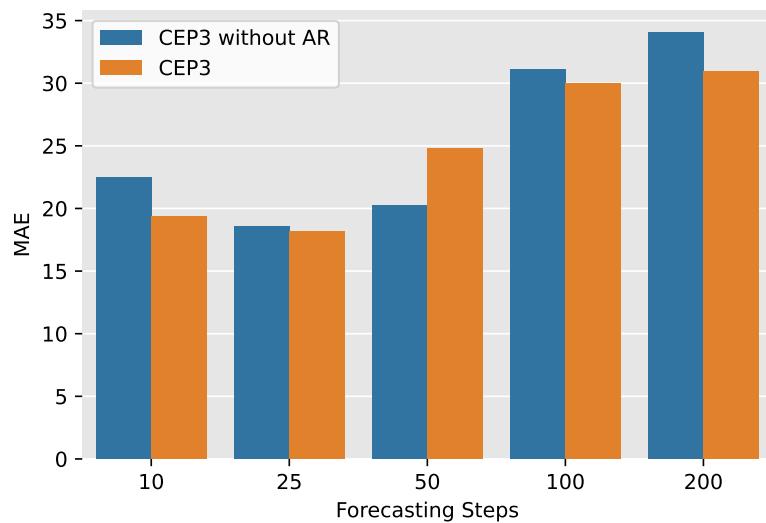


图 5-7 不同预测步长的模型在时间戳预测方面的表现

Figure 5-7 Performance of models with different prediction steps in time stamp prediction

**CEP3 w RNN。**尽管我们的模型在 Wikipedia 等大型交互网络数据集上显示了令人满意的结果，但当数据本身的结构模式化不足或其事件具有长期依赖性时，我们的模型的无内存机制和基于图聚合的性质可能会有性能不足缺点。例如，在 Github 数据集上，与纯 CEP3 相比，基于 RNN 的节点状态更新模块带来了性能改进。原因是 Github 数据集是一个具有少量节点和边的小数据集，节点和节点之间很难形成特定的结构化模式，同时它比其他数据集具有更长的时间跨度。这意味着在这种情况下，交互频率较低，导致节点状态更新的频率不足，进一步导致内存存在过期的情况。

在大多数情况下，引入 RNN 模块带来的节点状态更新机制似乎可以提高模型的预测能力。但我们认为，这种模型性能的提升可以通过在编码器 GNN 中采样更多的邻居来逼近。通过选择更多的邻居，CEP3 模型可以更可靠地捕获时间和空间上的依赖关系。

为了验证我们的假设，我们设置了对比实验来证明大量时态邻居在动态注意力机制中的有效性。我们将不同采样邻居数量的 CEP3 模型与 CEP3 w RNN 模型



17010226

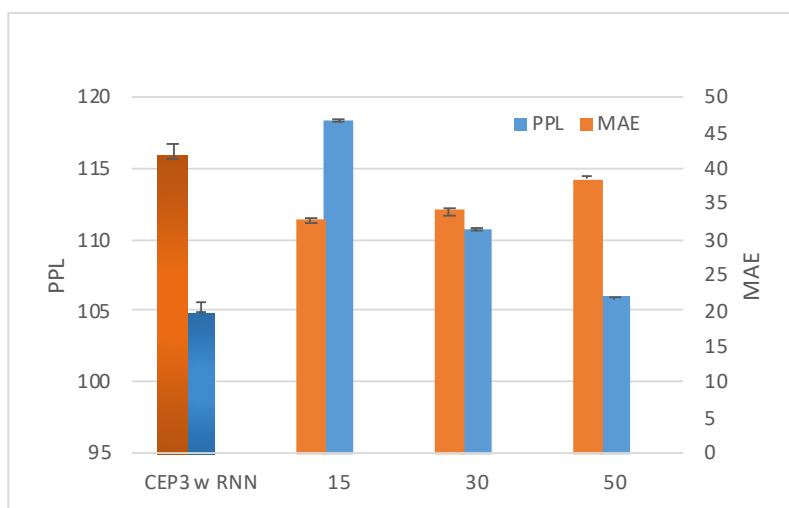


图 5-8 模型的性能与采样邻居个数之间的关系

Figure 5-8 Performance of the CEP3 w RNN versus pure CEP3 with different neighbors number.

一起进行性能对比，纯 CEP3 模型的邻居采样数量从 15 增加到 30,50。图5-8中的条形图显示了我们的实验结果，最左边一列是带有节点状态机制的 CEP3 模型，而右边的三列则是不同邻居数量的 CEP3 模型。它验证了我们的假设，即当采样 50 个邻居时，即使是无节点状态更新机制的普通 CEP3 模型可以提供近似于带节点状态更新模型 CEP3 w RNN 的预测能力。

### 5.3.8 并行训练能力

受自然语言处理任务中 Transformer<sup>[175]</sup>模型的启发，我们认为在建立时态图编码器时只使用纯粹的基于注意的模型来学习节点的特征是必要的，即抛弃掉使用 RNN 更新节点特征的方案。这是因为使用纯粹基于注意力机制的 GNN 作为编码器使我们能够按照第5.2.2节所述，以小批量方式并行训练多个时间窗口，而 Dyrep 和 RMTPP 等模型由于其 RNN 结构而无法采取并行的训练方式。

该特性使我们的 CEP3 模型能够受益于小批量训练所带来的好处，如稳定的梯度和更快的收敛速度。图5-9显示了我们在维基百科数据集上进行训练的过程中，关于不同数量的并行进程的损失函数的下降结果。损失函数曲线的下降情况表明使用并行训练可以显著提高速度，而不会损失准确性。

## 5.4 本章小结

在过去的动态图相关工作中，大部分人都遗漏了在动态图中估计社群关系演化的应用。而社群关系演化预测具有非常巨大的现实意义，比如疾病在社群传播



17010226

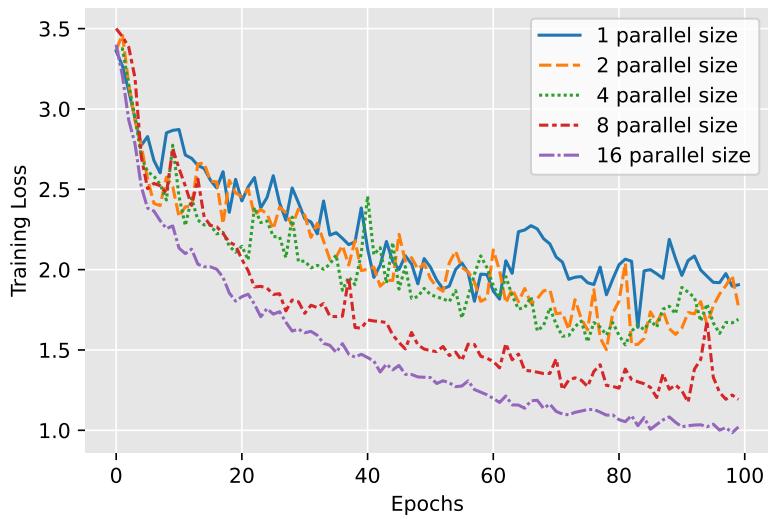


图 5-9 不同并行数量设置下的模型训练损失值曲线

Figure 5-9 Training loss curve of different parallel sizes.

的防控、城市拥堵的演化估计以及金融诈骗团伙的发展壮大。因此我们在动态图上提出了一种新的社群关系演化预测任务，并利用之前的动态图相关工作建立了该任务的基准。我们进一步提出了一个利用图结构解决这个问题的新模型——层次化随机点过程图神经网络（Community Event Predicting task with a graph Point Process model, **CEP3**）。通过这个模型的提出，我们解决了在图上利用时序点过程存在的可拓展性问题，并通过层次化建模降低了预测模型的复杂度。经过大量且广泛的实验，我们证明了我们提出的 CEP3 模型不但在性能上远超其他基线模型，而且在处理速度和对并行化训练的支持上都更加友好。



17010226

## 第六章 基于异步信息传播的实时动态图推理策略

### 6.1 引言

图是一种描述复杂网络的通用数学语言，它可以延续网络科学应用的各个领域，如经济网络、通信网络、交通网络、社会网络、交易网络、生物网络等。图  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  包含一个节点集  $\mathcal{V}$  以及一个边集  $\mathcal{E}$ 。每个节点  $v \in \mathcal{V}$  和边  $e \in \mathcal{E}$  可以分别具有其节点和边属性。随着图数据的应用越来越广泛，如何对图数据进行建模并将节点表示为下游任务的低维嵌入向量，已成为研究人员关注的关键问题。在这个问题上，图神经网络（GNN）有着广泛的应用，它已经成为实现这一目标的一种很有前途的方法。而到目前为止，大多数图学习工作，如 DeepWalk<sup>[64]</sup>和 SAGE<sup>[77]</sup>，都假定图是静态的，这意味着图是固定的和时不变的。

然而，大多数现实生活中的图系统是动态的：随着时间的推移，图上的节点和边可能会出现或消失，甚至节点属性也可能会改变。例如，在社交网络中，由于热点事件，用户通常会在短时间内将兴趣转移到其他实体；在经济网络中，欺诈者往往会突然实施一系列犯罪，然后在最短的时间内提取非法资金。假设我们采用静态图方法对这些动态网络进行建模，这将更简单、更省时，但我们无法捕捉拓扑结构的演化模式。此外，当我们学习动态网络上的节点表示时，我们需要考虑历史事件的影响。可以想象，动态图的研究难度远高于静态图。

近年来，基于动态图的图深度学习算法（动态 GNN）在图挖掘领域引起了越来越多的关注。这些算法旨在处理节点之间的时间交互  $(v_i, v_j, \mathbf{f}_{ij}^e, t)$ ，其中  $v_i, v_j$  代表发生交互的两个节点， $\mathbf{f}_{ij}^e$  代表本次交互附带的边特征， $t$  代表交互所发生的时间。这将比基于静态图的图挖掘算法在时间建模的能力上带来更大的弹性。大多数动态图算法，如 TGAT<sup>[22]</sup> 和 TGN<sup>[24]</sup>，通过由事件触发的动态子图聚合来建模动态节点状态。这些模型通常执行三个串行操作：图查询、图计算和模型推理。当一个交互  $v_i, v_j, \mathbf{f}_{ij}^e, t$  到来时，动态图算法首先需要通过图数据库访问它们的  $k$  阶动态邻居  $\mathcal{N}^{(k)}(v_i; t)$  和  $\mathcal{N}^{(k)}(v_j; t)$ 。然后，该图模型通过某种聚合机制汇总这些邻居的信息，该步骤称之为图计算。最后，模型根据聚合生成的消息生成节点  $v_i, v_j$  的节点嵌入，并执行最终的推理。

然而，虽然动态 GNN 算法在学习动态图嵌入方面取得了优异的性能，但几乎所有的图深度学习算法在实时计算时都面临着严峻的挑战。由于图深度学习方法对于数据规模的扩大更加敏感以及对于稀疏计算的需求非常庞大，这导致图模型在推理过程中存在较大的延迟。而这些延迟的瓶颈，如 6.2.1 的复杂度分析小节中



17010226

所述，主要来源于上述的图查询和图计算步骤：图深度学习在图查询的时间复杂度为  $O(|V| \cdot |E|)$ ，而图计算的复杂度则是最低  $O(|V|)^*$ 。这导致 GNN 在边数和节点数动辄上亿的大规模图中几乎无法进行实时部署。然而从目前来说，关于图神经网络模型的推理加速技术的研究仍然没有解决这个痛点。首先，图模型对于数据规模的敏感性导致过去的基于欧氏神经网络模型的网络量化模型失去效用。此外，尽管目前已经存在许多针对图计算模型的稀疏计算需求提出了软硬件加速框架，但这些框架普遍存在着适用范围过窄以及优化不完全的问题。有关这两部分的详细分析我们已经在2.4.5小节进行了梳理。

为了形象地说明图深度学习存在的速度瓶颈，我们给出一个具体的用例来展示动态 GNN 方法的应用存在的问题。支付宝是世界上最大的在线支付应用，用户和用户之间将进行各种转账交易，形成庞大的金融交易网络。与此同时，该应用每天都会发生数千起贪污、欺诈、洗钱和赌博案件。为了建模金融交易网中的拓扑关系，基于图的算法在这些金融欺诈检测任务中至关重要。然而，如果欺诈检测系统无法立即禁止欺诈交易，欺诈者可能会在系统响应之前提取非法资金，从而逃避平台的监控，给金融平台和用户造成无法估量的经济和声誉损失。除了对金融犯罪的快速反应外，在其他优势领域应用动态图也面临各种技术问题。首先，当交互频率在短时间内突然增加时，例如黑色星期五或双十一购物节，图数据库将过载，整个网络平台可能会变得不稳定。其次，虽然动态图算法有潜力捕获节点和图的快速变化，但如果由于效率限制，我们无法在在线平台上部署它们，这种潜力就无法最大化。

总结来说，动态 GNN 方法为利用大规模时态事件学习节点表示提供了一种很有前途的方法，但模型推理时间面临着最严峻的时效性挑战。

在这项工作中，我们重新设计了 GNN 框架，使模型推理和图查询步骤解耦，这样繁重的图查询操作不会影响模型推理的速度。为了便于解释说明，我们将类似 TGAT 的算法称为同步 GNN，将本章提出的改进算法称为异步 GNN。在图6-1中，我们解释了同步动态 GNN 和异步动态 GNN 算法之间的差异。图6-1a代表了同步 GNN 的运算逻辑和流程。由于用户无法忍受巨大的图数据库中邻居查询的高延迟，在在线支付平台上部署同步动态图模型几乎毫无价值。如果我们将其部署在离线系统中，我们可能无法在欺诈者提取非法资金之前禁止账户，从而给平台造成经济和声誉损失。而图6-1b则代表我们提出的异步动态 GNN 的推理逻辑，异步信息传播注意力网络（**Asynchronous Propagation Attention Network, APAN**，从根本上重新设计了动态图算法的工作流程。它将图查询和图数据更新阶段转移到模型推理的后面，将繁重的查询和计算操作放到异步链路中，可以将复杂的算法



17010226

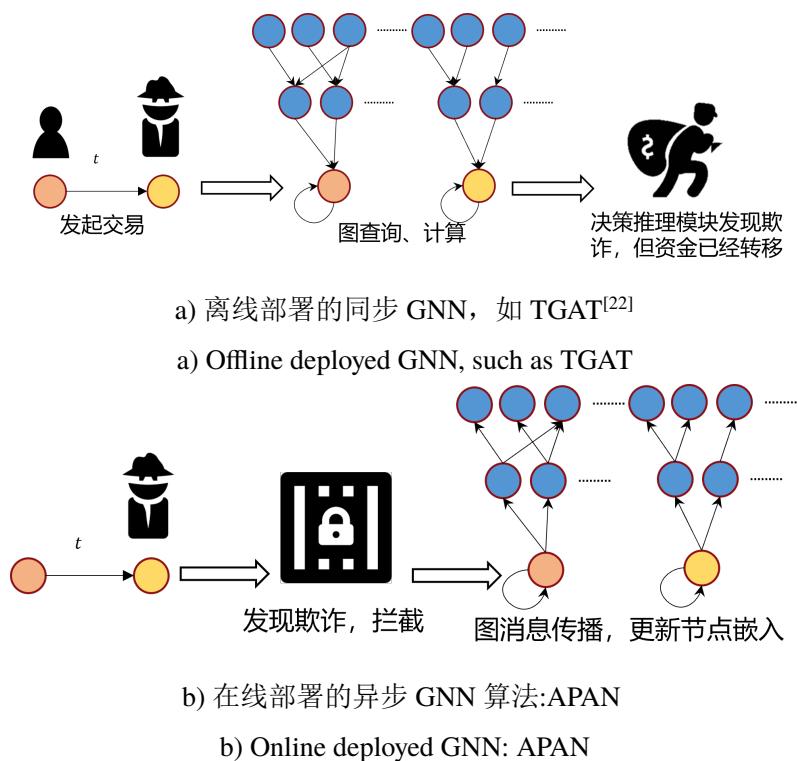


图 6-1 同步及异步动态 GNN

Figure 6-1 Synchronous and asynchronous temporal GNN

从在线业务决策系统中分离出来,从而获得更高的系统稳定性和可扩展性。TGAT 使用图聚合技术来建模时态图结构,而 APAN 使用图传播。APAN 满足我们在线部署的实时性要求。直观地说,异步时间图可以满足我们的要求,但是设计一个异步时间图并不像调整图计算阶段的顺序那么简单。

异步传播注意网络(APAN)是我们首次提出的满足上述异步时态图算法框架的模型。APAN 有两个链路:同步推理链路和异步传播链路。在异步链路中,一旦交互完成,交互的详细信息将作为“邮件”发送到其 k 阶邻居的“邮箱”;在同步链路中,当交互发生时,APAN 不需要查询动态图中的邻居,而只读取相关节点的“邮箱”,并生成实时推理。

APAN 已经在三个现实世界的时态图数据集(包括两个公共数据集和一个从支付宝收集的在线支付数据集)上进行了广泛测试。与其他最先进的图深度学习模型相比,APAN 在具有竞争力的性能的同时显著提高了推理速度。此外,由于 APAN 是一种学习动态图嵌入的方法,因此该模型可以与几乎所有的图学习任务相结合,比如节点分类、边分类、链路预测、节点聚类等等。

除了高推理速度、高性能以及高的兼容性之外,异步传播机制也为 APAN 带



17010226

来了更多好处。

1. 由于交互的详细信息存储在邮箱中, APAN 有可能成为一个可解释的模型, 详见第6.3.4小节中的描述。
2. APAN 克服了传统动态图算法对批量大小敏感的缺点, 详见第6.4.7小节中的描述。
3. 在某些任务中, 提高推理速度也可以有效地提高业务价值, 详见第6.4.6小节中的描述。

## 6.2 部署图深度学习模型的瓶颈分析



图 6-2 用常规思路部署图深度学习模型的困难

Figure 6-2 The difficulty of deep graph learning models deploying in a regular way

我们将用图6-2具体化地介绍图模型的在线部署。图模型与普通的机器学习模型最大的不同在于其读取的数据是来自于图数据库, 而不是普通的关系型数据库。图数据库读取的最大问题在于无法百分之百保证其实时性, 其读取速度跟图数据库的规模以及图的结构息息相关。例如, 在最极端情况下, 假设一个节点与图上的所有节点都有连接, 那么读取该节点的邻居就意味着要广度搜索遍历整张图, 这毫无疑问很难在一个规定的最短时间内完成。同理, 图模型对于图数据的处理时间也受到图的规模影响。在这从图数据库读取数据到生成推理结果写回在线数据库的整条链路上, 唯一不受到图的规模影响的就是推理决策模块, 因为该模块通常仅仅只是一个普通的多层感知机。这启迪我们应该将容易受到图的规模影响的图数据读取和图神经网络计算模块从在线链路上抽离, 移动在离线的链路中, 从而保证在线链路的整体时效性。

### 6.2.1 时间复杂度分析

在本小节中, 我们分别针对图查询、图计算和模型推理三大步骤的时间复杂度进行分析。

首先是图查询步骤, 由于图算法需要访问每个节点的  $k$  阶邻居子图, 则图数据库需要建立广度优先搜索树结构以应对访问, 并且当每个新事件到来的时候, 图数据库需要不断地更新这个树以保证搜索结果的正确性。针对全联通图上的一个



17010226

节点来说，进行广度优先搜索具有对图的规模近似于线性的复杂度  $O(|\mathcal{V}| + |\mathcal{E}|)$ ，其中  $|\mathcal{V}|$  和  $|\mathcal{E}|$  分别是该节点个数与边的个数。而针对动态图而言，边的个数通常远远大于节点的个数，因此，一个节点的搜索时间复杂度可以写为  $O(|\mathcal{E}|)$ ，而针对全图节点的搜索时间复杂度可以写成  $O(|\mathcal{V}| \cdot |\mathcal{E}|)$ 。

方法	种类	计算机制	时间复杂度
Bruna 等 <sup>[71]</sup>	基于谱	全阶数图傅里叶变换	$O( \mathcal{V} ^3)$
Henaff 等 <sup>[193]</sup>	基于谱	全阶数图傅里叶变换	$O( \mathcal{V} ^3)$
ChebNet <sup>[72]</sup>	基于谱	基于多项式核的图傅里叶变换	$O( \mathcal{E} )$
Kipf&Welling 等 <sup>[73]</sup>	基于谱	一阶图傅里叶变换	$O( \mathcal{E} )$
DCNN <sup>[194]</sup>	基于空间	基于多项式核的图傅里叶变换 + 扩散核	$O( \mathcal{V} ^2)$
DGCN <sup>[195]</sup>	基于空间	一阶空间消息聚合 + 扩散核	$O( \mathcal{V} ^2)$
MPNNs <sup>[196]</sup>	基于空间	一阶空间消息聚合	$O( \mathcal{E} )$
SAGE <sup>[77]</sup>	基于空间	一阶空间消息聚合 + 采样	$O( \mathcal{V} )^*$
DiffPool <sup>[197]</sup>	基于空间	自适应阶数聚合	$O( \mathcal{V} ^2)$
GAT <sup>[75]</sup>	基于空间	一阶空间消息聚合	$O( \mathcal{E} )$
JK-Nets <sup>[81]</sup>	基于空间	自适应阶数聚合	$O( \mathcal{E} )$
ECC <sup>[198]</sup>	基于空间	一阶空间消息聚合	$O( \mathcal{E} )$
R-GCNs <sup>[199]</sup>	基于空间	一阶空间消息聚合	$O( \mathcal{E} )$
PinSage <sup>[200]</sup>	基于空间	随机游走	$O( \mathcal{V} )^*$
FastGCN <sup>[78]</sup>	基于空间	一阶空间消息聚合 + 采样	$O( \mathcal{V} )^*$
SGC <sup>[201]</sup>	基于空间	基于多项式核的空间消息聚合	$O( \mathcal{E} )$
GIN <sup>[74]</sup>	基于空间	一阶空间消息聚合	$O( \mathcal{E} )$

表 6-1 一些图神经网络算法的时间复杂度分析

Table 6-1 Time complexity of current popular graph neural network algorithms

其次是图计算步骤，在表6-1中我们列出了一些在产业界中常用的经典图深度学习算法的时间复杂度，其中所有的数据均来自于文献<sup>[202]</sup>。这些复杂度的估计均只考虑了亿级别的图规模参数（节点数  $|\mathcal{V}|$ 、边数  $|\mathcal{E}|$ ）而不考虑网络的规模等等数百级别的复杂度参数。从表格中我们可以看出，绝大多数图深度学习模型达到了  $O(|\mathcal{V}|^2)$  或  $O(|\mathcal{E}|)$  以上级别的时间复杂度，甚至有些算法的复杂度与节点数成二次方和三次方关系。注意，标有 \* 号的方法的复杂度受到采样规模的影响，实际上为  $O(|\mathcal{V}|s^L)$ ， $s$  和  $L$  分别代表模型的邻居采样个数以及图神经网络的层数。而由于  $s$  和  $L$  通常较小，我们认为其复杂度与图的节点数目是近似于线性的。而由于动态图中的边数通常都是节点数的几十倍，因此就算是与边数成线性关系的模型也几乎无法在大规模动态图上实时地部署和应用。不过，一些轻量级的图深



17010226

度学习算法通过引入采样机制，以牺牲准确率为代价将复杂度优化到了  $O(|\mathcal{V}|s^L)$  级别，一般认为，这种与节点数近似成线性时间复杂度关系的轻量级的图深度学习模型通常不会成为模型实时部署的瓶颈，就算问题的规模上升到数亿级别，人们仍然可以想办法通过并行计算、分布式计算等方式优化这类线性复杂度的算法到近似实时的速度。

注意在我们分析图计算步骤的复杂度时，只考虑了跟节点数量  $|\mathcal{V}|$  和边数量  $|\mathcal{E}|$  的关系，没有考虑图神经网络本身的规模。这是因为图神经网络的宽度本身可能是上百最多上千，而在大规模图中的节点数目可以轻松破亿。所以图算法模型的规模在复杂度分析中可以忽略不计，算法所处理的图的规模才是我们主要考虑的部分。也就是说图模型中的图计算机制很复杂，而模型本身的神经网络并不复杂。因此，对图神经网络进行轻量化压缩并不能降低其本身的算法复杂度级别。故一些针对欧氏神经网络模型的轻量化方法不在本章的对比实验范围之内。

而在最后的模型推理阶段则几乎不存在延迟和时间复杂度的问题。因为模型推理通常是使用一个普通的前馈神经网络，根据矢量化的特征嵌入，输出预测的标签值或者回归的实数值，这意味着模型推理的速度与图的规模无关，而只与神经网络的规模有关。一条数据经过一个层数为  $l$  的前馈神经网络的时间复杂度近似跟神经网络每层的平均的神经元个数  $n$  成线性关系： $O(n_1+n_2+\dots+n_l) \Rightarrow O(n)$ ，而神经元个数总是有限的，一般不会超过数百，与动辄上亿规模的图相比起来显得如此微不足道。更何况，当今时代基于 GPU、TPU<sup>[203]</sup>、FPGA<sup>[145]</sup>的神经网络并行计算技术已经非常成熟，甚至可以认为前馈型神经网络的复杂度近似于常数  $O(1)$ 。这意味着图深度学习方法的模型推理阶段完全不会成为算法的实时性瓶颈。

### 6.3 基于异步信息传播的实时动态图模型

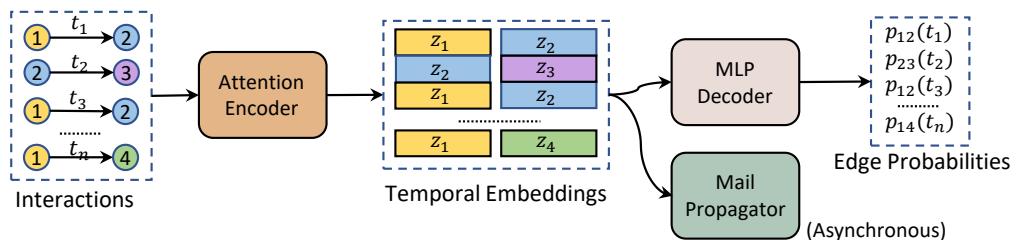


图 6-3 异步信息传播模型的整体框架

Figure 6-3 The overall framework of Asynchronous Propagation Attention Network (APAN)

图6-3概述了我们提出的异步注意力传播模型（Asynchronous Propagation Attention Network, APAN）的整体框架，它是第一个用于实时动态图嵌入的异步动



17010226

态图算法。APAN 主要可分为三个部分：基于注意力机制的编码器、基于多层感知器（MLP）的解码器和异步邮件传播模块。请注意，编码器和解码器位于同步链路中，它们不需要从图数据库中查询邻居的信息。因此，从交互发生到模型推断的时间延迟将非常短，用户将获得非常流畅的体验。在模型推断之后，邮件传播器将根据交互生成一个“邮件”，然后沿着时间边缘将其传播到  $k$  阶邻居的“信箱”。

**同步链路：**当两个节点发生交互后，编码器根据该事件的具体信息、节点的上一时刻的历史嵌入和邮箱中的邮件数据更新节点嵌入。请注意，如果一个节点在一个批处理中涉及多个交互，则嵌入将只更新一次。尽管如此，总的来说，新事件中所有节点的时间嵌入都需要实时更新，这促进了模型捕捉图的动态特性。之后，MLP 解码器将利用这些更新的节点嵌入来实现下游任务，例如关系推理、节点分类、边分类和节点聚类等等。由于编码器和解码器都是纯神经网络不包含图计算，我们不需要在图数据库中查询图上的邻居，因此完成这两个阶段将非常快。

**异步链路：**生成某个节点的动态嵌入后，邮件传播器首先根据交互事件的信息创建一封邮件，然后沿着动态图上的边将其传播到其  $k$  阶邻居的邮箱。邮件包含与该节点相关的交互的信息。由于邮件传播器处于异步链路中，不会损害用户体验，因此我们可以在该模块中处理一些更复杂的计算，例如聚合多层邻居或计算子图统计值。

### 6.3.1 基于注意力机制的编码器

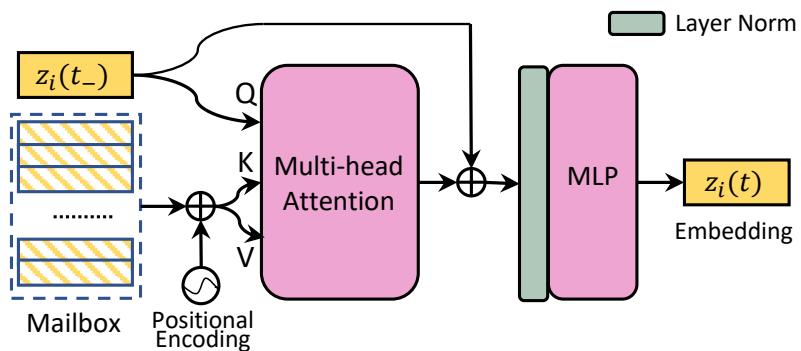


图 6-4 APAN 的编码器架构图

Figure 6-4 Encoder architecture diagram of APAN

图6-4提供了 APAN 中基于注意的编码器的详细信息。该编码器引入了一个经典的注意力体系结构，根据上一时刻节点嵌入  $z(t-) \in \mathcal{R}^d$  和邮箱  $\mathcal{M}(t) \in \mathcal{R}^{m \times d}$  中的信息的相关性，来创建当前时刻的节点嵌入  $z(t) \in \mathcal{R}^d$ 。其中  $m$  是邮箱中的



17010226

最大邮件数， $\oplus$  表示按位加法， $z(t-)$  表示节点最后一次参与交互时的嵌入。邮箱记录了其邻居（包括  $k$  阶邻居）过去参与的互动的详细信息。通过这种方式，编码器间接地实现了动态邻居信息的聚合，以更新其节点嵌入。为了实现这一目标，编码器由三个主要部分设计，即位置编码、多头注意力聚合和层标准化。

**位置编码：**如果我们要考虑到收到邮件的到达顺序，我们需要尝试对每一封邮件进行位置编码。因为我们已经设置了邮箱中邮件的最大数量，所以我们可以将位置信息转换成固定维度的稀疏 one-hot 格式向量，然后将它们馈送到嵌入查找层（Embedding look-up layer）<sup>①</sup>。one-hot 格式向量是指，向量的所有维度中，有且仅有一个维度是非 0 值，其他维度都是 0，而嵌入查找层的输出是一个稠密向量，这种稠密向量具备平稳特性，更适合神经网络学习。

对于每个节点的邮箱  $\mathcal{M}(t) = (mail_1, mail_2, \dots, mail_m)$ ，位置编码层通过以下方式将位置信息组合到原始邮箱矩阵中：

$$\hat{\mathcal{M}}(t) = \mathcal{M}(t) + \mathcal{P}(t) = [mail_1 + p_1, \dots, mail_m + p_m]^\top \quad (6-1)$$

其中  $\hat{\mathcal{M}}(t), \mathcal{M}(t), \mathcal{P}(t) \in \mathbb{R}^{m \times d}$ ， $m$  邮箱能容纳的最大信息数量， $d$  是邮件向量的维度。邮件维度  $d$  默认为数据集边特征的维度，我们将在 6.3.3 节中解释。

**多头注意力机制：**标准化的点积注意力<sup>[175]</sup>被用作编码器的注意模块。注意层的隐藏机制可以定义为：

$$\begin{aligned} \text{Attn}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) &= \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}}\right)\mathbf{V}, \\ \mathbf{Q} &= z(t-) \mathbf{W}_Q, \\ \mathbf{K} &= \hat{\mathcal{M}}(t) \mathbf{W}_K, \\ \mathbf{V} &= \hat{\mathcal{M}}(t) \mathbf{W}_V \end{aligned} \quad (6-2)$$

其中  $\mathbf{Q}$  被称为“查询（Queries）”， $\mathbf{K}$  和  $\mathbf{V}$  分别被称为“键（Keys）”和“值（Values）”。点积注意力的输出就是  $\mathbf{V}$  的加权和，其权重通过  $\mathbf{Q}\mathbf{K}^\top$  这对矩阵的点积计算给出。 $\mathbf{Q}\mathbf{K}^\top$  之间第  $i$  行的点积越大，则  $\mathbf{V}$  矩阵对应的第  $i$  行对于最终的输出越重要。 $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V \in \mathbb{R}^{d \times d_h}$  ( $d_h$  是注意力层的输出维度) 是用于学习合适的  $\mathbf{Q}\mathbf{K}^\top\mathbf{V}$  关系的投影权重矩阵。通过这些矩阵，注意力模型可以根据不同的数据，给出不同的注意力输出。通过使用点积注意力层，APAN 模型可以捕获某节点上一时刻的节点嵌入  $z(t-)$  以及其节点邮箱  $\mathcal{M}(t)$  的相关关系，这也意味着注意力模块可以根据接收到的来自节点历史交互的邻居的邮件来确定如何更新节点嵌入。

<sup>①</sup> 具体定义见 <https://pytorch.org/docs/stable/generated/torch.nn.Embedding.html>。嵌入查找层是一个简单的查找表，用于存储固定字典和大小的嵌入项。该模块通常用于存储单词嵌入，并使用索引检索它们。模块的输入是索引列表，输出是相应的单词嵌入。



17010226

在实际应用中，注意模型通常采用多个注意头来形成多个子空间和力模型来学习不同方面的信息。为了构建多头注意模块，我们构造了多个注意并将它们连接起来。假设我们在注意力模型中设置了  $k$  个注意头：

$$\begin{aligned} \text{head}_i &= \text{Attn}(Q_i, K_i, V_i), i = 1, \dots, k \\ \text{MultiHead}(Q, K, V) &= \text{Concat}(\text{head}_1, \dots, \text{head}_k) W^O, \end{aligned} \quad (6-3)$$

其中  $W^O \in R^{d \times d}$ ,  $\text{head}_i \in R^{\frac{d}{k}}$ 。

**层标准化 (Layer normalization)**：由于不同节点的注意力输出是不同的，我们需要一个标准化方案来限制输出的均值和方差。层标准化<sup>[176]</sup>是注意力模型中最常见的标准化选择，因为复杂的注意机制可能会破坏统一训练批次内的数据统计分布。如果我们使用批标准化<sup>[204]</sup>，可能会导致次优结果。层标准化通过计算用于标准化的平均值和方差来实现这一目标，这些平均值和方差来自一个层中神经元的所有求和输入：

$$\begin{aligned} a &= \text{MultiHead}(Q, K, V) + z(t-) \\ \mu &= \frac{1}{d} \sum_{i=1}^d a_i, \quad \sigma = \sqrt{\frac{1}{d} \sum_{i=1}^d (a_i - \mu)^2} \\ \bar{a} &= f\left(\frac{g}{\sqrt{\sigma^2}} \odot (a - \mu) + b\right) \end{aligned} \quad (6-4)$$

$d$  代表了层标准化层的输入， $\mu$  和  $\sigma$  代表层标准化学习到的均值和方差，这两者在不同的隐藏层中是共享的。 $\odot$  是两个向量之间的按元素相乘。可学习的参数  $b$  和  $g$  被定义为偏差和增益，以确保规范化操作对原始信息没有影响。之后，层标准化的输出将被传送到多层感知机网络中，以生成新的节点动态嵌入。

### 6.3.2 MLP 解码器

APAN 可广泛用于各种下游任务，其中的注意力编码模块和邮件传播模块可以直接使用，无需任何架构更改，而 MLP 解码器需要进行微调以适应不同的任务。MLP 解码器的任务是利用动态节点嵌入为下游任务生成预测。例如，如果我们需要预测两个节点之间是否会有交互，那么这两个动态节点嵌入应该连接为  $(z_i(t)||z_j(t))$ ，并传递给解码器，让解码器判断；如果我们需要判断一个边是否是欺诈交易，那么两个节点嵌入和边特征应该连接为  $(z_i(t)||\mathbf{f}_{ij}^e(t)||z_j(t))$  并传送给解码器。灵活的解码器选择是 APAN 可以适应于各种下游任务的必要保证。



17010226

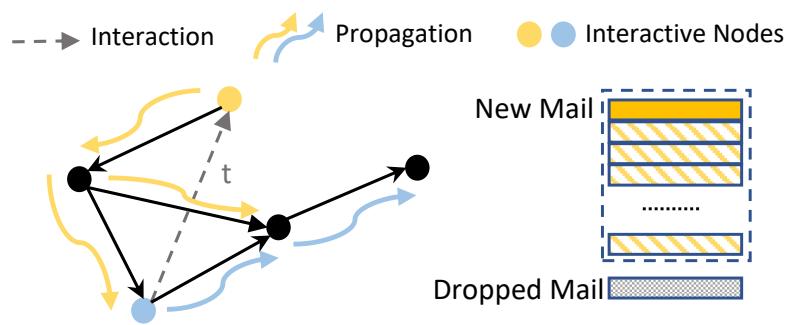


图 6-5 异步邮件传播模块的工作流

Figure 6-5 The workflow of asynchronous mail propagation module

### 6.3.3 异步邮件传播模块

在图6-5中，我们用最简单的方式演示我们 APAN 模型的邮件传播器的整个工作流。在注意力编码器生成节点嵌入之后，邮件传播器首先创建一个交互邮件，然后沿着动态边将其传播到  $k$  阶（在本文中  $k=2$ ）邻居的邮箱。通过信息邮件传播机制，一个节点可以通过访问其邮箱获得其邻居的历史交互信息。

黄色节点和蓝色节点在时间  $t$  交互，其交互附带边特征  $\mathbf{f}_{ij}^e(t)$ ，所以这次交互可以用  $(z_i(t), \mathbf{f}_{ij}^e(t), z_j(t))$  来表示， $z_i(t)$  和  $z_j(t)$  产生于注意力的编码器，代表节点  $v_i$  和  $v_j$  在  $t$  时刻的节点嵌入。APAN 中的邮件传播过程可以用以下两个数学公式来描述：

$$\begin{aligned} Mail : mail(t) &= \phi(z_i(t), \mathbf{f}_{ij}^e(t), z_j(t)), \\ Mailbox : M(t) &= \psi\left(M(t-), \rho\left(\{f(mail(t))\}_{N^{(k)}(v_i, v_j; t)}\right)\right), \end{aligned} \quad (6-5)$$

其中  $\phi(\cdot)$  是一个邮件生成函数，用于总结交互的具体信息。 $N^{(k)}(v_i, v_j; t) = N^{(k)}(v_i; t) \cup N^{(k)}(v_j; t)$  代表节点  $v_i$  和  $v_j$  的时态邻居集合。 $f(\cdot)$  是一个邮件传递函数，用于定义消息在传播中衰减的方式。 $\rho(\cdot)$  是一种邮件合并函数，当一个节点同时收到多封邮件时，可将多封传入邮件聚合为一封邮件。 $\psi$  是根据新收到的邮件更新邮箱的更新函数。下面我们按顺序一个一个介绍各个函数的功能。

**邮件生成函数 ( $\phi$ ):** 一旦一个节点参与了一次交互，该函数目标是生成一封邮件，记录该节点在该交互中发生的情况。邮件的一种最简单形式就是两个交互节点的当前嵌入和当前交互的边缘特征的总和，即  $(mail(t) = z_i(t) + \mathbf{f}_{ij}^e(t) + z_j(t))$ 。请注意，邮件还额外带有时间戳信息。我们使用求和而不是串联的原因是求和可以节省邮箱占用的内存容量。缺点是求和限制了节点嵌入的维数，在某些情况下可能会获得次优解。



17010226

**时态邻居采样 ( $\mathcal{N}_{ij}^{(k)}(t)$ )：**邮件生成后，我们应该将其发送到其他节点，以便让它们知道邻居发生了什么。然而，向所有邻居发送邮件无疑是效率低下的。在大多数 GNN 算法中，节点消息在采样子图上传播。不同之处在于，一些算法使用均匀采样<sup>[77]</sup>，一些算法使用加权采样<sup>[200]</sup>，其他算法使用自适应邻居采样<sup>[205]</sup>。在本文中，我们将最近邻居（即交互时间最靠近当前的 k 个邻居）抽样策略应用到我们的 APAN 中，因为动态图方法旨在建模快速变化的趋势并更新节点嵌入。因此，最近邻居的采样更容易恢复时变信息，类似的实验和结论可以在该研究中找到。

**邮件传播函数 ( $f$ )：**在确定传播边界后，也就是说，在确定采样的邻居之后，我们需要一个函数，用于找到邮件的合理衰减或映射模式。这个想法是非常直观的，现在我们想象往一个平静的水塘中投入一颗石子，产生的水波会随着扩散的距离以某种形式进行衰减。在 APAN 中，邮件传播函数被简单地设置为恒等传播。由于邮件传播路径严格遵守节点及其邻居之间的交互形成的图结构。因此，历史图数据的结构特征可以通过邮件传播函数来获取。

**邮件合并函数 ( $\rho$ )：**实际上，一个节点通常在邮件传播期间接收多封邮件。活动（高热度）节点通常比非活动（低热度）节点接收到更多邮件。为了避免这种不平衡，我们使用“平均”操作的合并缩减函数将多封邮件转换为一封。这样，每个节点在每个批次中只接收一封邮件。这种机制不但确保了邮件的传递不会受到节点交互频次的影响，同时该设计也为下一个邮箱更新模块带来了更多的稳定性。

**邮箱更新函数 ( $\psi$ )：**节点收到邮件后，其邮箱应更新，以总结节点邻居的历史状态。为了尽可能简洁，我们直接采用先进先出的队列数据结构来更新邮箱。通过这种队列结构，邮箱将保留最新邮件信息并丢弃旧邮件。

邮箱和信息邮件传播机制除了对于在线部署的动态 GNN 模型是必要的之外，该机制也会给动态 GNN 模型带来其他好处。比如在流媒体系统（尤其是分布式流媒体系统）中，我们无法保证事件将以时间戳顺序到达。因此，它会给一些依赖 RNN 动态更新的机器学习模型带来不稳定性，例如 TGN<sup>[24]</sup> 和 JODIE<sup>[21]</sup>。邮箱机制修复了这一严重问题，因为在读取邮箱中的信件时，邮箱中存储的事件和消息不需要严格地按照时间戳排序，允许数据延迟导致的顺序错位。

#### 6.3.4 异步动态 GNN 框架

从概念上讲，我们首次提出的异步动态 GNN 框架旨在解决基于 GNN 的方法很难在毫秒级在线平台上部署的问题。而 APAN 只是符合该框架的最简单模型之



17010226

一。注意力编码器和邮件传播器中的几乎每个模块仍有很大的改进空间。与我们提出的这些简单模块相比，其他更复杂的模块可能有更大的潜力来改进异步动态GNN框架：

**邮箱更新函数：**键值存储网络<sup>[128]</sup>框架为增强邮箱更新模块提供了一个可能的方向。

**位置编码：**Xu等人<sup>[22]</sup>提出的时间核函数可以用来编码时间戳，以取代APAN中的位置编码模块。

**可解释性：**假设在时间  $t$  发生交互，那么改次交互就会生成一封存储着交互信息的邮件，其包含一对节点嵌入向量  $z_i(t), z_j(t)$  以及交互边的特征  $\mathbf{f}_{ij}^e(t)$ 。然后我们可以使用注意力权重来计算节点邮箱中的哪封邮件对最终节点嵌入的影响最大。这种可解释性是其他模型所不具备的，因为其他模型无法存储  $z_i(t)$  和  $z_j(t)$ ，他们只能获取边特征  $\mathbf{f}_{ij}^e(t)$ 。

**时态邻居采样：**本章所使用的最近邻居采样存在着自适应能力不强的问题，因为我们无法确保最近的 K 个邻居一定需要接收模型广播的全部信息。虽然在大部分时刻，即使更新节点的邮箱都是有益的。但如果存在某些邻居并不需要更新信息，而我们强制进行了更新，会造成信息的丢失和冗余。而基于强化学习的图神经网络邻居采样算法<sup>[206]</sup>为我们提供了有效的工具，可以使用一些智能策略来引导邻居的采样。

**邮件传播函数：**本章所使用的邮件传播函数是一个简单的恒等式，而其在传播过程中并没有利用图的结构。如果我们在邮件逐层传播的过程中乘上加入文献<sup>[201]</sup>所述的图结构核  $\tilde{T}^k, \tilde{T} = \tilde{D}^{-1/2} \tilde{A} \tilde{D}^{-1/2} = (D + I_n)^{-1/2} (A + I_n) (D + I_n)^{-1/2}$ ，就可以将图结构信息融入其中。

然而，引入太多技巧会影响读者对我们提出的异步动态图整体架构的信任。我们确实已经完成了一些相关扩展的开发，但我们只在实验数据集上测试了它们，没有在实际环境中部署它们。因此，我们将在未来的工作中提出这些改进。我们也期待图机学习界的研究人员提出更好的异步时态图模型。

## 6.4 对比实验与分析

我们在从支付宝平台收集的一个大型工业数据集和两个公开的动态图基准数据集上，针对关系推理、节点分类和边缘分类任务，测试了各种先进的基线（在可能的情况下修改设置以适用于动态图）和我们所提方法的性能。我们模型的源代码是使用 PyTorch<sup>[148]</sup>和 Deep Graph Library<sup>[43]</sup>实现的，并发布在 Github 存储库<sup>①</sup>中。

① <https://github.com/WangXuhongCN/APAN>



17010226

#### 6.4.1 数据集

	Wikipedia	Reddit	Alipay
边数	157474	672447	2776009
节点数	9227	10984	761750
边特征维度	172	172	101
训练集中的节点数	7475	10844	760289
验证集和测试集中已见节点数	3131	10181	379368
验证集和测试集中未见节点数	1752	140	1461
时间跨度	30 天	30 天	14 天
数据集分割方法	70%-15%-15%	70%-15%-15%	10d-2d-2d
有标签的交互数量	217	366	11632
标签类型	禁止编辑	禁止发帖	禁止交易

表 6-2 实验中使用的数据集的统计数据

Table 6-2 Statistics of the datasets used in our experiments

在本文中，我们使用了三个真实的动态图数据集，包括两个公共数据集和一个工业数据集，来广泛评估 APAN 的性能。表格6-2显示了我们实验中使用的数据集的统计信息。

Wikipedia 和 Reddit 数据集是两个动态图数据集<sup>①</sup>[21]。Wikipedia，中文名可译为“维基百科”，Wikipedia 的节点代表用户和维基百科页面。节点之间的链接表示百科页面被用户编辑，该数据集是一个二部动态图（二部图的定义详见第二章中的定义2.7），在一个月的时间跨度内，由大约 9300 个节点和 16 万个动态边组成。在该数据集中，大概 1% 的用户在使用过程中被禁止发布，即视为异常用户。Reddit 是一个国外使用最广泛的综合类论坛，该数据集也是一个二部动态图，收集了一个月的用户交互数据，其中包含近 11000 个节点和 70 万条动态边。Reddit 数据集中的交互是指用户通过帖子与 Reddit 论坛上的某个话题进行交互。在该数据集中，0.5% 的用户在使用过程中被禁止在某个话题下发布帖子，对于这部分用户我们视之为异常。在这两个数据集中，用户的编辑由一段文本代表，并转换为 172 维的 LIWC 类别<sup>[164]</sup>（共 172 维）的边特征向量。根据交互时间戳，数据集被拆分为 70%/15%/15%，分别作为训练集/验证集/测试集。

支付宝数据集是从支付宝平台收集的金融交易数据集，由 ~76 万个节点和 ~277 万条时间边组成。每条边都有一个标签，指示某个交易是否为欺诈，并且每

① 下载地址：<http://snap.stanford.edu/jodie>



17010226

条边关联着 101 维的特征，它是支付宝平台上记录的特殊用户和交易属性。支付宝数据集中使用 10 天-2 天-2 天的间隔拆分为训练、验证和测试集。虽然 APAN 模型已经在实际业务平台上进行了速度、内存和准确性方面的全面测试。但由于隐私政策，我们无法披露更多信息和实际业务指标。相反，我们在支付宝数据集中报告常见指标，以描述 APAN 算法的业务效果。

与论文 TGAT, TGN 类似，由于动态图算法侧重于对交互事件进行建模，这些数据集中不存在节点特征，因此我们将相同的零特征向量分配给所有节点。

#### 6.4.2 下游任务

在不同的下游任务中，我们需要选择不同的损失函数和指标来评估各种模型的性能。例如，在关系推理预测任务中，我们使用准确度（Accuracy）和平均精度（Average Precision, AP）作为度量。为了使用交叉熵损失函数进行训练，我们设计了一种时变的负采样策略，以构建正负样本对：

$$\ell = \sum_{(v_i, v_j, \mathbf{f}_{ij}^e, t) \in \mathcal{G}} -\log (\sigma (-\mathbf{z}_i(t)^\top \mathbf{z}_j(t))) \\ - \mathbb{E}_{v_n \sim P_n(v)} \log (\sigma (\mathbf{z}_i(t)^\top \mathbf{z}_n(t))), \quad (6-6)$$

其中，求和项是节点  $v_i$  和  $v_j$  在时间  $t$  的训练交互， $\sigma$  是一个 sigmoid 函数<sup>[207]</sup>， $P_n(v)$  是负采样分布。请注意，动态图的负样本池也在不断地动态变化。首先，从未交互过的节点不能作为负数据进行采样。其次，随着互动的继续和时间的拓展，一对历史上的正负样本可能不再有效。

在节点和边缘分类任务中，由于标签分布的倾斜，我们使用 ROC 曲线<sup>[208]</sup>下的面积（The area under the ROC curve）AUC<sup>[209]</sup>作为度量。

#### 6.4.3 基线模型

作为一种动态图方法，APAN 的主要竞争对手有五种动态图嵌入方法：CTDNE<sup>[126]</sup>、DynRep<sup>[26]</sup>、JODIE<sup>[21]</sup>、TGAT<sup>[22]</sup> 和 TGN<sup>[24]</sup>。此外，我们还包括六种静态图嵌入方法，以显示动态图算法的优势：DeepWalk<sup>[64]</sup>、Node2Vec<sup>[65]</sup>、SAGE<sup>[77]</sup>、GAT<sup>[75]</sup>、GAE 以及 VGAE<sup>[123]</sup>。我们已经在第2.4小节中介绍了这些基线。请注意，我们的实验设置严格遵循 TGAT<sup>[22]</sup> 和 TGN<sup>[24]</sup>。在 Wikipedia 和 Reddit 数据集中，所有基线的结果都严格继承自其原始论文。公平地说，我们使用与原始论文相同的数据处理和拆分方法。对于支付宝数据集，我们根据这两篇论文中描述的基线设置实现了我们自己的版本，以探索这些图算法在大规模工业数据集上的性能。



17010226

#### 6.4.4 模型配置

对于所有数据集，我们使用 AdamW<sup>[178]</sup>随机梯度优化器，学习率为 0.0001，训练、验证和测试时我们使用的批量大小均为 200，Dropout 比率为 0.1，并且使用了早停策略<sup>[165]</sup>，早停耐心值为 5。注意力头的数量设置为 2，图神经网络的消息传递层数为 2。对于编码器和解码器中的 MLP 网络，我们采用了隐藏大小为 80 的两层前馈神经网络。请注意，上述参数均取自 TGAT 和 TGN 的原始论文，我们没有采用复杂的超参数调整来改善 APAN 结果。

APAN 的节点嵌入维数固定为原始边特征维数，因此它不是一个超参数。对于所有三个数据集，邮箱插槽（即最大能存储的邮件总数）和采样邻居的数量都设置为 10。在随后的实验中，我们将证明 APAN 方法对超参数不敏感。只要参数设置在合理范围内，APAN 几乎不会导致灾难性性能。

#### 6.4.5 结果分析

	Wikipedia		Reddit	
	Accuracy	AP	Accuracy	AP
GAE	72.85 (0.7)	91.44 (0.1)	74.31 (0.5)	93.23 (0.3)
VAGE	78.01 (0.3)	91.34 (0.3)	74.19 (0.4)	92.92 (0.2)
DeepWalk	76.67 (0.5)	90.71 (0.6)	71.43 (0.6)	83.10 (0.5)
Node2vec	78.09 (0.4)	91.48 (0.3)	72.53 (0.4)	84.58 (0.5)
GAT	87.34 (0.3)	94.73 (0.2)	92.14 (0.2)	97.33 (0.2)
SAGE	85.93 (0.3)	93.56 (0.3)	92.31 (0.2)	97.65 (0.2)
CTDNE	79.42 (0.4)	92.17 (0.5)	73.76 (0.5)	91.41 (0.3)
DyRep	87.77 (0.2)	94.59 (0.2)	92.11 (0.2)	97.98 (0.1)
JODIE	87.04 (0.4)	94.62 (0.5)	90.91 (0.3)	97.11 (0.3)
TGAT	88.14 (0.2)	95.34 (0.1)	<u>92.92 (0.3)</u>	98.12 (0.2)
TGN	<u>89.51 (0.4)</u>	<b>98.46 (0.1)</b>	92.56 (0.2)	<u>98.70 (0.1)</u>
APAN	<b>90.74 (0.1)</b>	<u>98.12 (0.2)</u>	<b>94.34 (0.1)</b>	<b>99.22 (0.2)</b>

表 6-3 关系推理任务的实验结果分析

Table 6-3 Experimental result analysis of link prediction task

表格6-3显示了我们的 APAN 与 11 条最先进的基线模型在关系推理任务上的对比实验结果。在关系推理任务中，我们进行了 10 次随机试验，汇报了平均的准确率和 AP (Average Precision) 以及他们的标准差。请注意，最好的结果用加粗字



17010226

	Node classification		Edge classification
	Wikipedia	Reddit	Alipay
GAE	74.85 (0.6)	58.39 (0.5)	\
VGAE	73.67 (0.8)	57.98 (0.6)	\
GAT	82.34 (0.8)	64.52 (0.5)	69.47 (0.4)
SAGE	82.42 (0.7)	61.24 (0.6)	67.91 (0.5)
CTDNE	75.89 (0.5)	59.43 (0.6)	\
DyRep	84.59 (2.2)	62.91 (2.4)	65.09 (1.0)
JODIE	83.17 (0.5)	59.90 (2.1)	81.89 (0.7)
TGAT	83.69 (0.7)	<u>65.56 (0.7)</u>	77.84 (0.9)
TGN	<u>88.56 (0.3)</u>	<b>68.63 (0.7)</b>	<b>84.01 (0.9)</b>
APAN	<b>89.86 (0.3)</b>	65.34 (0.4)	<u>83.37 (0.7)</u>

表 6-4 节点/边分类任务的实验结果分析

Table 6-4 Experimental results analysis of node / edge classification task

体排版，第二个好的结果用下划线突出显示。

显然，几乎所有基于动态图的方法都优于静态图方法。无监督图嵌入方法，如 GAE、DeepWalk、Node2Vec 和 CTDNE，性能较差，因为这些方法学习的嵌入对于下游任务不可知的，对下游任务的贡献有限且间接。与其他最先进的方法相比，我们的 APAN 实现了具有竞争力的性能。尤其是在 Reddit 数据集中，APAN 比其他方法表现出惊人的性能。

APAN 等异步动态图算法具有传统同步算法所不具备的优势。同步动态图方法通常在节点交互后使用更新函数创建节点，而在异步时态图中，只要节点的邻居参与交互，该节点的邮箱就会更新。换句话说，异步动态图算法中的节点更新频率高于同步动态图算法中的节点更新频率。正是这种差异使得 APAN 在动态图嵌入方面具有更强大的能力。我们也可以从表格 6-4 中节点或边分类任务的结果中得出类似的结论。

请注意，本研究中使用的结构和超参数对于我们的应用来说是足够的，尽管它们仍然可以改进。APAN 的主要改进是大大提高了推理速度，APAN 的算法架构特别适合在互联网平台上在线部署。



17010226

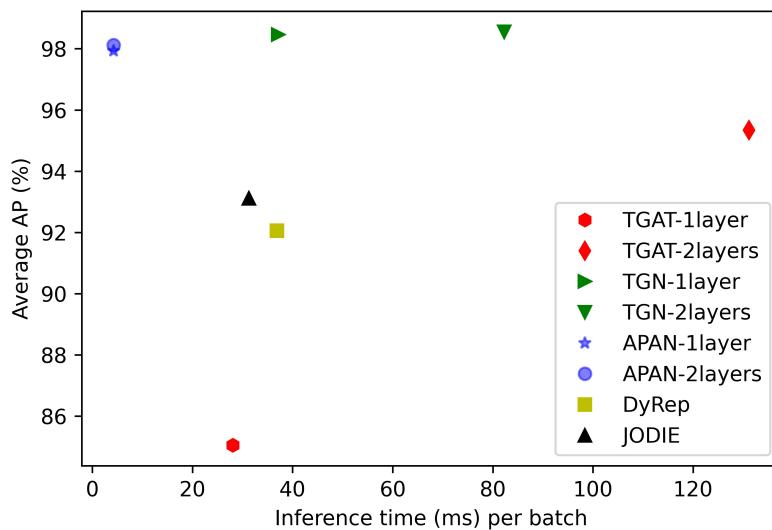


图 6–6 关于推理时运行效率的对比实验结果

Figure 6–6 Comparative experimental results on the efficiency of reasoning

#### 6.4.6 运行效率

在本节中，我们将 APAN 的效率与其他基线进行比较。这些实验在一台 Linux PC 上运行，该 PC 配有 Intel Core i7-7820X CPU (8 核, 3.60GHz) 和 12 GB NVIDIA TITAN X (Pascal) GPU。这些模型的实现版本来自两个公共存储库<sup>①</sup>。

在现实的在线互联网平台上，模型的在线推理时间比训练时间更重要。以支付宝反欺诈系统为例，交易只有通过反欺诈系统才能执行。较长的推理时间会占用太多的计算资源，并导致在线推理引擎的不稳定性，但 APAN 通过将大部分计算放在异步链路上克服了这一问题。此外，如果推理时间过长，会极大地损害用户体验。因此，低推理延迟的模型将大大提高投资回报率，并增加模型的商业价值。

我们进行了运行时间的实验，以模拟一批交互（交互个数我们设置为 200）通过这些动态图算法所需的平均等待时间。图6–6和图6–7分别显示了 APAN 在推理和训练阶段的速度，该实验统一在 Wikipedia 数据集和关系推理任务的设置中进行。图中的标记越靠近左上角，模型的运行效率以及运行性能越好。注意，我们只计算从发生交互到模型推断的时间，不包括 APAN 异步链路上的时间。

在推理阶段，APAN 比 TGN 快 8.7 倍，而准确率几乎相同。JODIE 和 DyRep 受

① <https://github.com/twitter-research/tgn>

<https://github.com/StatsDLMathsRecomSys/Inductive-representation-learning-on-temporal-graphs>



17010226

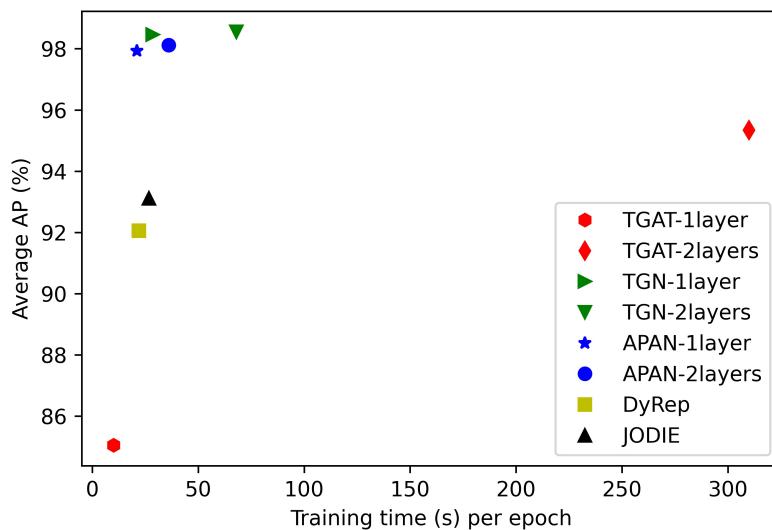


图 6-7 关于训练时运行效率的对比实验结果

Figure 6-7 Comparative experimental results on running efficiency during training

到表达能力的限制，因此他们的准确率表现落后于 APAN。随着层数的增加，TGN 和 TGAT 的性能有所提高，但推理速度也大大降低；APAN 的推理速度不会随着层的变化而变化，因为它的异步邮件传播机制。这意味着我们可以在 APAN 中应用更复杂的网络计算，在实时推理的限制下进一步提高性能。

请注意，在实际系统中，APAN 的速度增益会远大于 8.7 倍。因为在我们的实现中，整个图存储在单个 PC 内存中，而它存储在真实平台的分布式图数据库中。大规模分布式数据库的读取效率将成为整个系统的瓶颈，而在使用单机内存时则不是。APAN 框架因为将推理过程和图查询过程解耦，所以不会受到大规模分布式数据库的读取效率影响。

在训练阶段，APAN 的测试结果和速度几乎与当前最快的算法 TGN 相同。原因是，在训练阶段，APAN 与其他动态图算法非常相似。APAN 只是交换了计算顺序，没有引入额外的计算。此外，在我们的实现中，采用了底层高效的信息传播机制，因此增加层数不会导致训练时间的显著增加。

#### 6.4.6.1 结合第三四五章的方法的能效分析

由于 APAN 是兼容性非常强的图嵌入模型，其可以在保持几乎性能不大幅降低的同时加速几乎任意动态图模型中的图嵌入模块，并由此应用在各个广泛的动力图学习任务中。然而，如果某个模型的主要计算瓶颈不在于图嵌入部分，而在于



17010226

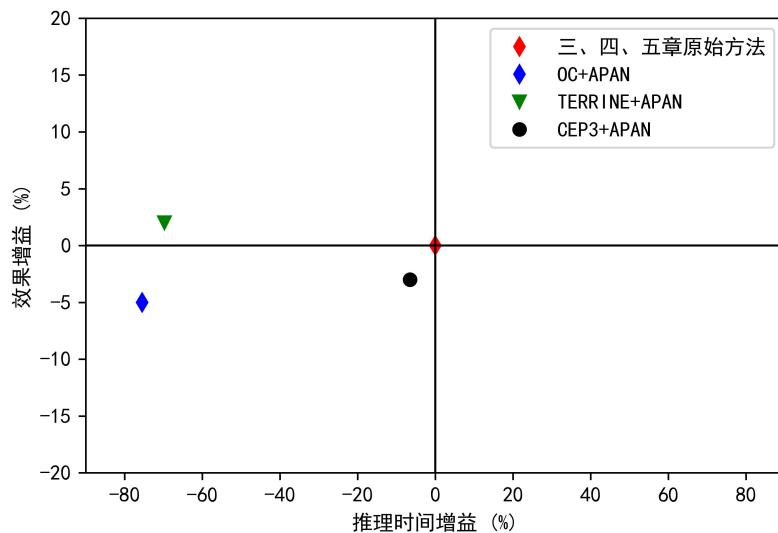


图 6-8 将 APAN 方法与前面三章结合进行的实验，越靠近左上角，APAN 的能效提升越明显

Figure 6-8 Experiments combining the APAN method with the previous three chapters' methods, the closer to the upper left corner, the more obvious the performance and efficiency improvement of APAN

其他的模块，那么 APAN 的加速效果通常会变得不显著。图6-8说明了如果 APAN 方法与前面三章的内容结合会带来怎么样的改变。由于前面三章的各个任务的评价指标和运行时间都很难统一，因此我们统一使用加速的百分比以及性能损失的百分比来进行比较。红色为三四五章的原始方法，蓝、绿、黑分别代表三、四、五章的模型引入 APAN 之后的加速和性能提升效果。由此图我们可以看出，APAN 对于第三、四章的模型加速效果比较明显，而对第五章的加速效果不太明显，这主要是因为第五章的速度瓶颈不在于图深度学习的编码器，而可能在于基于随机点过程的事件预测器中的蒙特卡洛采样函数。这启迪我们需要进一步第优化 APAN 的适用范围。

#### 6.4.7 参数敏感性

**k 阶邻居采样：** k 表示图算法在邻居采样期间所经过的层数。k 阶邻居的概念在某种程度上类似于社交网络中的 k 度连接。k 越大，算法需要处理的信息规模就越大，对内存和计算资源的需求也就越大。以前的 GNN 著作<sup>[73,77]</sup>从经验和理论上证明，过大的 k 阶采样不会给算法带来额外的增益，反而会降低系统的运行速度。我们的实验也支持这一结论，并且从直觉上讲是合理的。在社交网络中，人



17010226

们只与较低度数的关系互动，一般超过 3 度的好友关系就不会产生关联了。

表格6-5显示邻居采样的阶数选择如何影响模型的 AP 和训练时间，这个实验是在 Wikipedia 数据集和关系推理任务中进行测试的。对于这三种模型，2 阶采样在性能和复杂性之间实现了最佳平衡。不幸的是，TGN 和 TGAT 的源代码在使用更大的 K 时面临内存不足（OOM）问题。我们的 APAN 在速度和内存方面更友好，尤其是在面对大规模图数据时。

Models	邻居采样阶数	Average AP (%)	训练时间 (秒)
APAN	1	97.93	21.86
	2	98.12	35.75
	3	98.38	56.78
	4	98.36	86.68
	5	97.45	125.05
	6	97.67	164.70
	7	97.07	205.26
TGN	1	98.46	28.5
	2	98.55	68.41
	3	98.63	464.96
	4+	OOM	OOM
TGAT	1	85.05	10
	2	95.34	310
	3+	OOM	OOM

表 6-5 邻居采样的阶数选择对模型的性能和运行效率的影响

Table 6-5 The influence of the k-hop neighbor sampling on the performance and efficiency of the model

**批次大小：**除了高推理速度外，异步信息传播机制还为 APAN 带来了更多好处：对训练批次大小的鲁棒性。动态图算法的常见缺点是对训练批次大小的敏感性。动态图算法最理想的条件是更新由单个事件触发的节点，即批大小等于 1。其原因如下：假设一个批次从  $t_0$  开始，而批次中的某个交互发生在时间  $t$ ，则  $t_0$  和  $t$  之间的最新交互将丢失，因为动态图模型假设一个批中的交互事件同时到达。因此，批量越大，丢失的信息越多，动态图模型的性能越差。

为了确保良好的性能，TGAT 和 TGN 在 70 万级 Reddit 数据集中采用了小到 200 的批量。然而，在现实世界中，互联网平台可能需要判断每批数千乃至数万个事件，以适应业务需求。基于此，我们需要一种对批量大小不敏感的动态图算法。



17010226

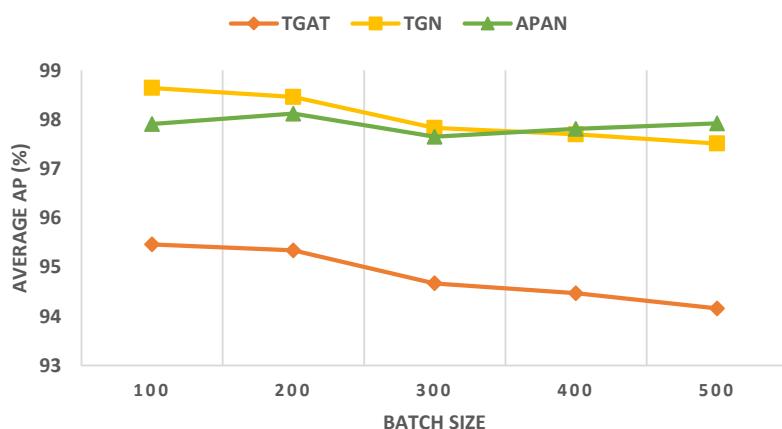


图 6-9 训练批次大小对于模型性能的影响

Figure 6-9 Effect of training batch size on model performance

幸运的是，异步动态图框架不需要查看最新的交互。同步动态图模型在交互发生时查询最新的动态子图，而我们的 APAN 首先输出节点嵌入，然后查询再子图。因此，APAN 会丢失节点的最新的一次交互信息。让我们用时间序列预测给出一个生动的例子。给定一个时间序列  $x(1), \dots, x(t-2), x(t-1), x(t)$ ，同步时态图模型捕获关系  $x(t-1) \rightarrow x(t)$ ，而 APAN 映射  $x(t-2) \rightarrow x(t)$ 。只要节点的变化趋势是连续的，我们可以很自然的确认使用  $x(t-2)$  来预测  $x(t)$  是合理的。

因此，APAN 被迫学习了如何在不使用最新子图的情况下创建推理。异步信息传播机制有效地避免了批量大小对性能的影响。此外，APAN 更能容忍系统延迟，因为在这种情况下，最新信息不会及时到达，同步的动态图模型的性能会受到严重影响，而 APAN 不会。

图6-9中反映的是在链接预测任务和 Wikipedia 数据集设置中，训练批次大小与模型性能之间的关系。我们可以得出结论，在同步的动态图算法中，训练批次大小越大，算法性能越差。在现实世界中，平台可能需要每批处理数千个事件。基于上文所述的异步传播带来的好处，APAN 对训练批量大小不敏感。然而，TGAT 和 TGN 的性能随着批量的增加而降低。

**采样邻居的数量和邮箱容量：**在图6-10中，我们在 Wikipedia 数据集和关系推理任务的设置下，展示了 APAN 对两个重要的超参数——采样邻居数和邮箱容量（Mailbox Slot）——的低敏感性。在这两个参数的网格搜索实验中组成的 16 个结果中，APAN 表现出强大的低参数敏感性，最好和最差的结果波动仅为 0.6%。虽然 APAN 对这两个参数不敏感，但我们仍然分析了造成性能差异的原因。

**a) 邮箱容量：**我们可以得出结论，少量的邮箱容量就足够了，因为动态图模



17010226

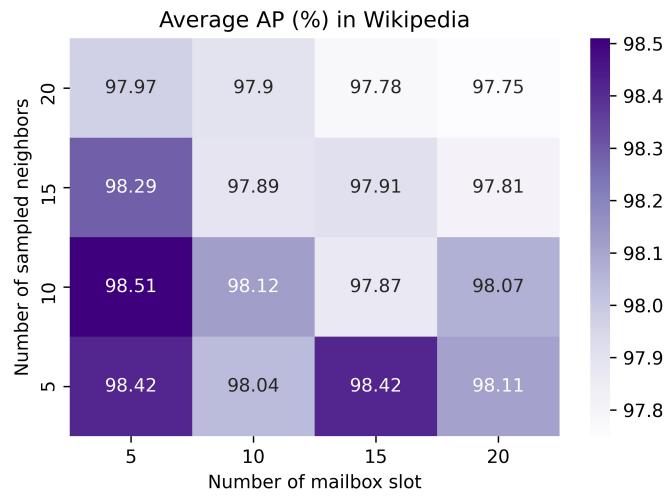


图 6–10 APAN 模型中，邻居采样个数和邮箱容量大小对模型性能的影响

Figure 6–10 In apan model, the number of neighbor samples and mailbox capacity affect the performance of the model

型只需要参考短期交互就可以在下游任务中进行推理（类似的结论见 TGAT<sup>[22]</sup>）。一个有大量容量的邮箱将存储历史上的冗余信息，并使模型难以学习。**b)** 采样邻居数：采样邻域数是 GNN 模型中最常见的参数。如果聚集了太多的邻居，模型可能无法区分聚集后的表示；如果采样的邻居太少，可能会错过重要的邻居。

此外，我们还解释了读者可能对邮箱机制带来的额外内存使用的担忧。首先，图6–10中的实验已经证明，非常少的邮箱容量就已经可以获得竞争性的结果。其次，内存使用仅与节点数量有关，在大多数情况下，节点数量是有限的，而交互边的数量是无限的（例如，Reddit 数据集中有 1 万节点却有近 70 万条边）。由于 GNN 模型需要存储大量历史交互边信息，因此邮箱机制不是整个系统的内存瓶颈。

总之，APAN 是一个非常鲁棒的模型，因为它对主要的超参数不敏感，人们可以花费更少的精力在调节参数和模型细节上，而可以花更多的时间用于理解自己的业务上。

## 6.5 本章小结

在本文中，我们提出了异步信息传播注意力网络（Asynchronous Propagation Attention Network，APAN），一种用于超快速推理的异步动态图算法框架。APAN 旨在改造传统的动态图算法，以适应互联网平台上的在线部署和实时推理。我们的大量实验结果表明，所提出的 APAN 可以在推理速度提高 8 倍的情况下获得具



17010226

有竞争力的性能，并且其具有对主要参数的低敏感性。据我们所知，APAN 是第一个可以达到毫秒级的 GNN 算法，可以帮助实现在线分布式图数据库中的超大规模推理。它可能会增强行业未来的设计解决方案，即如何在推荐系统、金融系统、社交网络等方面适应 GNN 模型。在未来，我们将为提出的异步都动态图框架探索更多的改进机会。



17010226

## 第七章 全文总结

### 7.1 主要研究结论

在这个信息爆炸的时代，人们对于复杂系统的建模的需求越来越高，越来越多的复杂场景以图的形式表示数据，比如社交网络、电子商务、交通网络和工业物联网。因此，图的建模与分析具有较高的科研价值和实际意义。图神经网络，作为一种基于深度学习的图表示学习模型，已经逐渐成为图挖掘领域的热点之一。本论文致力于将图神经网络技术引入到这类大规模复杂图系统的异常检测与分析任务中所存在的几点重要问题进行改进。

我们针对图深度学习技术在动态图上的异常检测和动态分析等任务上存在的有偏估计、时空协同性不足、多模态适应能力不足以及无法实时性部署问题，分别在以下四个方面进行了新颖的改进。

1) 由于过去的机器学习和异常检测算法无法考虑实体之间（例如好友关系、转账关系等）的相关性，而同时一些基于传统方法的图异常检测文献存在非线性表达能力弱、运行效率低以及方法参数繁多复杂等问题，大大制约了图异常检测技术在业界中的实际应用。而深度图神经网络无论是在大规模数据的处理还是在非线性建模等方面都有十分出色的表现，并且它的超参数较少，学习成本低。然而，数据类别不平衡非常容易给图深度学习算法带来**有偏估计问题**并最终导致局部次优解。为此，我们提出了基于超球面学习的图神经网络框架 OCGNN，试图将正常数据与异常的数据通过图神经网络的映射，用特征映射空间的超球面加以区分，既弥补了传统的异常检测算法在处理和提取图关系特征的不足，也以半监督学习的方式克服了有监督图深度学习算法的有偏估计问题。该框架可以兼容几乎所有图神经网络模型骨架，其使用范围甚至可以从静态图泛化到动态图，并且该框架在带来了性能提升的同时，还比最先进的图异常检测方法复杂度更低更易于使用。为了支撑这些结论，我们进行了大量包括性能实验、参数敏感性实验、运行效率实验在内的各项实验，其结果表明，提出的学习框架相比传统的异常检测算法确实实现了显著的改进。

2) 动态图深度学习存在**时空协同性不足问题**：过去的工作通常仅仅考虑空间和时间的分别建模，而没有强调两者之间的协同性。我们首先改进了动态图模型的训练目标，提出了相较传统静态关系推理更困难的时空间的联合建模任务——**动态关系推理**。我们认为要处理动态关系预测问题，模型必须要学习预测实体之间动态的关系变化并且必须要考虑节点特征和网络结构关系在空间和时间维度上



17010226

的联合建模，这增强了模型在动态建模方面的能力。由此，我们则提出时间-空间联合建模的图神经网络算法 TERRINE，在综合考虑节点特征和相关性的基础上，加入时序性建模。我们认为一个交互事件的“源节点-时间-目标节点”三个变量在特征向量空间中应该满足三角闭合关系，即源节点向量 + 时间向量  $\approx$  目标节点向量，并为了在大规模数据上训练这个模型设计了同时考虑时间和空间的大规模负采样算法。在推理阶段，给定源节点，利用与其他节点之间的相对位置关系即可确认目标节点，再利用三角关系求得时间向量，最后再对时间向量进行岭回归预测出具体的交互时间。这让我们的模型不光可以预测节点对于其他节点的在空间上的影响，而且可以根据时间的不同给出灵活的预测，比单纯的基于空间的关系预测更加适应实际应用的需要。广泛的实验结果表明了，根据此目标设计的动态图模型在动态关系预测任务中取得了更优异的性能表现。

3) 图上的演化预测模型存在应对不同区域上的不同演化模式的**多模态适应能力不足问题**：一些过去的工作仅仅只使用单独的一个随机过程网络来描述整个图的演化，忽略了图上不同的区域可能具有不同模态的演化模式，带来了模型对于不同演化模式的适应能力不足问题。而另一些纷纷以超高的复杂度代价，在每两个节点之间建立一个点过程模型，这通常需要  $|\mathcal{V}|^2$  复杂度的计算资源，当社区过大时算法存在运行效率瓶颈。而我们将整个图上的演化预测任务拆分为对各个社群分别进行演化预测，提出了**社群关系演化预测任务**。针对这个任务，我们提出了基于图神经网络的层次化随机时序点过程社群演化建模方法 CEP3，受益于图神经网络带来的关联性建模能力，我们可以对一个图上的交互事件“源节点-时间-目标节点”进行层次化级联建模，先估计事件发生的时间，再分别估计事件的源节点和目标节点。同时我们也设计了基于动态信息传播的自回归框架，它可以根据前一事件的发生对社区造成的影响来决定当前事件的预测结果。我们解决了在图上利用时序点过程存在的可拓展性问题，并通过层次化建模降低了预测模型的复杂度。经过大量且广泛的实验，我们证明了我们提出的层次化随机时序点过程模型不但在性能上远超其他基线模型，而且在处理速度和对并行化训练的支持上都更加友好。

4) 现今的图神经网络算法存在**无法实时性部署问题**，而无论是网络剪裁、硬件加速还是数据库优化都有一个理论优化上限。如果不修改关键的瓶颈部分，图模型则始终没办法做到在线推理。我们经过对时间复杂度的系统性分析，认为由于图模型需要的数据过多，其最本质的性能瓶颈在于，读取图数据库会严重拖慢模型的推理时间。因此，我们提出了异步信息传播注意力网络 APAN，改造了传统的动态图算法，重新设计了图神经网络模型的计算和图数据查询机制以解决查询



17010226

历史图数据过慢的问题，并适应互联网平台上的在线部署和实时推理要求。具体来说，我们使图神经网络模型的图查询和图数据更新阶段转移到模型推理的后面，这样模型推理步骤是在线实时完成，而图查询步骤则是异步离线完成。这种方法将推理步骤和图查询步骤解耦，如此，繁重的图查询操作将不会影响模型推理的速度，这帮助我们成功地将复杂的图算法从在线业务决策系统中分离出来，从而获得更高的系统稳定性和可扩展性。我们还通过一种网络中的消息信箱机制来保证就算在异步查询计算步骤阻塞的情况下，在线推理步骤也可以保证较高的准确度。大量实验结果表明，文中所提出的框架可以在推理速度提高 8 倍的情况下获得具有竞争力的性能，并且该模型对主要参数的敏感性较低。该框架将可以帮助实现在线分布式图数据库中的超大规模推理。它提出了目前 GNN 模型在实际系统中的应用改善方案，即如何在推荐系统、金融系统、社交网络等方面实时性地部署应用 GNN 模型。

## 7.2 研究展望

本文主要针对图神经网络在复杂动态图系统应用中的关键技术和相关建模理论进行了深入的研究。除了本文的研究内容之外，图神经网络还有许多值得深入探索研究的方向，如下所示：

1. 可解释性仍需改进。虽然图神经网络在各类应用场景的应用和理论研究正如雨后春笋般爆发，但是受制于其部分主要组件仍然是传统的神经网络，其模型也不可避免地受到神经网络的影响，解释性非常低，以至于可以被视为是黑箱模型。继续研究图神经网络的解释性是优先级非常高的任务之一，其原因主要有两点。第一，相比于普通的神经网络，图神经网络具有更好解释性的潜力。因为图神经网络建模的就是图上节点之间的相关性关系，如果能把节点的相关性以某种方式可视化，那么就可以很好地展示图神经网络模型究竟学习到了什么。第二，正是因为图系统中存在大量的不可预见的影响，我们更加迫切需要高度的解释性作为使用模型的安全保障。

2. 仍需针对具体细分场景给出模型建议。本论文没有针对某个特殊的场景进行模型的单独设计，而是深入研究通用模型理论，追求设计一些即插即用的模型。然而，在现实情况下，即插即用的模型很少能取得最优秀的效果，为了更好的适应实际环境，将模型做定制化修改是有必要的。比如在生物医药中，由于分子或蛋白质在 3D 空间中折叠、反应、聚合，而目前的图神经网络其实更多考虑的是 2D 平面空间中的关联关系，如何提出一种 3D 曲线连边以更好地反应分子的折叠位置关系是一个有价值的话题。



17010226

3. 图神经网络结合图频域分析相关理论。早期的图神经网络几乎都是基于图频域分析理论来构建模型的，然而，目前流行的图神经网络为了其高效和快速计算的能力，已经逐渐放弃了图频域傅立叶分析这个优美的数学武器，转而在时域上使用各种形式的网络结构对节点聚合求平均。这类简单的时域网络虽然也具有良好的效果，但是根据图傅立叶的理论，这种做法等于是1阶超低通滤波器，过滤并丢弃了所有的高频信号换来的计算速度有的时候不一定值得。如果我们可以简单的时域网络中，仅仅加入可以快速计算的频域信号作为特征，将会有很大的机会带来显著的增益。



17010226

## 参考文献

- [1] 郭丽丽, 丁世飞. 深度学习研究进展[J]. 计算机科学, 2015, 42(5): 6.
- [2] Hinton G, Deng L, Yu D, et al. DEEP NEURAL NETWORKS FOR ACOUSTIC MODELING IN SPEECH RECOGNITION[J]. IEEE SIGNAL PROCESSING MAGAZINE, 2012, 29(6): 82-97.
- [3] Sutskever I, Vinyals O, Le Q V. Sequence to Sequence Learning with Neural Networks[C]. in: NIPS. 2014: 3104-3112.
- [4] Krizhevsky A, Sutskever I, Hinton G E. ImageNet Classification with Deep Convolutional Neural Networks[C]. in: NIPS. 2012: 1106-1114.
- [5] Bronstein M M, Bruna J, LeCun Y, et al. Geometric Deep Learning: Going beyond Euclidean data[J]. IEEE Signal Processing Magazine, 2017, 34(4): 18-42. DOI: 10.1109/MSP.2017.2693418.
- [6] Izenman A J. Introduction to manifold learning[J]. Wiley Interdisciplinary Reviews Computational Statistics, 2012, 4(5): 439-446.
- [7] 卜月华. 图论及其应用[M]. 图论及其应用, 2002.
- [8] 刘忠雨. 深入浅出图神经网络 GNN 原理解析[M]. 机械工业出版社, 2020.
- [9] Bronstein M M, Bruna J, Cohen T, et al. Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges[J]. CoRR, 2021, abs/2104.13478.
- [10] Wu Z, Pan S, Chen F, et al. A comprehensive survey on graph neural networks[J]. IEEE transactions on neural networks and learning systems, 2020, 32(1): 4-24.
- [11] Zhou J, Cui G, Hu S, et al. Graph neural networks: A review of methods and applications[J]. AI Open, 2020, 1: 57-81.
- [12] Cai H, Zheng V W, Chang K C C. A comprehensive survey of graph embedding: Problems, techniques, and applications[J]. IEEE Transactions on Knowledge and Data Engineering, 2018, 30(9): 1616-1637.
- [13] Akoglu L, Tong H, Koutra D. Graph based anomaly detection and description: a survey[J]. Data Min. Knowl. Discov., 2015, 29(3): 626-688.



17010226

- [14] Van Engelen J E, Hoos H H. A survey on semi-supervised learning[J]. Machine Learning, 2020, 109(2): 373-440.
- [15] Pimentel M A F, Clifton D A, Clifton L A, et al. A review of novelty detection [J]. Signal Process., 2014, 99: 215-249.
- [16] Durgut R, Turaci T, Kutucu H. A heuristic algorithm to find rupture degree in graphs[J]. Turkish Journal of Electrical Engineering & Computer Sciences, 2019, 27(5): 3433-3441.
- [17] Sandryhaila A, Moura J. Discrete signal processing on graphs: Graph fourier transform[C]. in: IEEE International Conference on Acoustics. 2013: 6167-6170.
- [18] Kriege N M, Johansson F D, Morris C. A survey on graph kernels[J]. Applied Network Science, 2020, 5(1): 1-42.
- [19] Zhao T, Zhang X, Wang S. GraphSMOTE: Imbalanced Node Classification on Graphs with Graph Neural Networks[C]. in: WSDM. ACM, 2021: 833-841.
- [20] Li X, Lu P, Hu L, et al. A novel self-learning semi-supervised deep learning network to detect fake news on social media[J]. Multimedia Tools and Applications, 2021: 1-9.
- [21] Kumar S, Zhang X, Leskovec J. Predicting Dynamic Embedding Trajectory in Temporal Interaction Networks[C]. in: KDD. ACM, 2019: 1269-1278.
- [22] Xu D, Ruan C, Körpeoglu E, et al. Inductive representation learning on temporal graphs[C]. in: ICLR. OpenReview.net, 2020.
- [23] Wang X, Lyu D, Li M, et al. APAN: Asynchronous Propagation Attention Network for Real-time Temporal Graph Embedding[C]. in: SIGMOD Conference. ACM, 2021: 2628-2638.
- [24] Rossi E, Chamberlain B, Frasca F, et al. Temporal Graph Networks for Deep Learning on Dynamic Graphs[J]. CoRR, 2020, abs/2006.10637.
- [25] Zhang Z, Bu J, Li Z, et al. TigeCMN: On exploration of temporal interaction graph embedding via Coupled Memory Neural Networks[J]. Neural Networks, 2021, 140: 13-26.
- [26] Trivedi R, Farajtabar M, Biswal P, et al. DyRep: Learning Representations over Dynamic Graphs[C]. in: ICLR (Poster). OpenReview.net, 2019.



17010226

- [27] Trivedi R, Dai H, Wang Y, et al. Know-Evolve: Deep Temporal Reasoning for Dynamic Knowledge Graphs[C]. in: Proceedings of Machine Learning Research: ICML: vol. 70. PMLR, 2017: 3462-3471.
- [28] Wu W, Liu H, Zhang X, et al. Modeling Event Propagation via Graph Biased Temporal Point Process[J]. IEEE Transactions on Neural Networks and Learning Systems, 2020: 1-11. DOI: 10.1109/TNNLS.2020.3004626.
- [29] Hall E C, Willett R M. Tracking Dynamic Point Processes on Networks[J]. IEEE Trans. Inf. Theory, 2016, 62(7): 4327-4346.
- [30] Lu Y, Wang X, Shi C, et al. Temporal Network Embedding with Micro- and Macro-dynamics[C]. in: CIKM. ACM, 2019: 469-478.
- [31] Zuo Y, Liu G, Lin H, et al. Embedding Temporal Network via Neighborhood Formation[C]. in: KDD. ACM, 2018: 2857-2866.
- [32] Cao J, Lin X, Cong X, et al. Deep Structural Point Process for Learning Temporal Interaction Networks[C]. in: ECML/PKDD. Springer, 2021.
- [33] Du N, Dai H, Trivedi R, et al. Recurrent marked temporal point processes: Embedding event history to vector[C]. in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2016: 1555-1564.
- [34] Mei H, Eisner J. The Neural Hawkes Process: A Neurally Self-Modulating Multivariate Point Process[C]. in: NIPS. 2017: 6754-6764.
- [35] Wang Y, Du N, Trivedi R, et al. Coevolutionary Latent Feature Processes for Continuous-Time User-Item Interactions[C]. in: NIPS. 2016: 4547-4555.
- [36] Farajtabar M, Wang Y, Gomez-Rodriguez M, et al. Coevolve: A joint point process model for information diffusion and network evolution[J]. The Journal of Machine Learning Research, 2017, 18(1): 1305-1353.
- [37] Wang X, Chen S, He Y, et al. CEP3: Community Event Prediction with Neural Point Process on Graph[J]. CoRR, 2022, abs/2205.10624.
- [38] Meng C, Mouli S C, Ribeiro B, et al. Subgraph Pattern Neural Networks for High-Order Graph Evolution Prediction[C]. in: AAAI. AAAI Press, 2018: 3778-3787.
- [39] Li Z, Ding X, Liu T. Constructing Narrative Event Evolutionary Graph for Script Event Prediction[C]. in: IJCAI. ijcai.org, 2018: 4201-4207.



17010226

- [40] Abadal S, Jain A, Guirado R, et al. Computing Graph Neural Networks: A Survey from Algorithms to Accelerators[J]. ACM Comput. Surv., 2022, 54(9): 191:1-191:38.
- [41] Zeng H, Prasanna V K. GraphACT: Accelerating GCN Training on CPU-FPGA Heterogeneous Platforms[C]. in: FPGA. ACM, 2020: 255-265.
- [42] Liang S, Wang Y, Liu C, et al. EnGN: A High-Throughput and Energy-Efficient Accelerator for Large Graph Neural Networks[J]. IEEE Trans. Computers, 2021, 70(9): 1511-1525.
- [43] Wang M, Yu L, Zheng D, et al. Deep Graph Library: Towards Efficient and Scalable Deep Learning on Graphs[J]. ICLR Workshop on Representation Learning on Graphs and Manifolds, 2019.
- [44] Zhu R, Zhao K, Yang H, et al. AliGraph: A Comprehensive Graph Neural Network Platform[J]. Proc. VLDB Endow., 2019, 12(12): 2094-2105.
- [45] Zhang D, Huang X, Liu Z, et al. AGL: A Scalable System for Industrial-purpose Graph Machine Learning[J]. Proc. VLDB Endow., 2020, 13(12): 3125-3137.
- [46] Falcao T A, Furtado P M, Queiroz J S, et al. Comparative Analysis of Graph Databases for Git Data[C]. in: Journal of Physics: Conference Series: vol. 1944: 1. 2021: 012004.
- [47] Li Y, Yu R, Shahabi C, et al. Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting[C/OL]. in: 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings. OpenReview.net, 2018. <https://openreview.net/forum?id=SJiHXGWAZ>.
- [48] Newman M E. Detecting community structure in networks[J]. The European physical journal B, 2004, 38(2): 321-330.
- [49] Zhang J, Nawata K. Multi-step prediction for influenza outbreak by an adjusted long short-term memory[J]. Epidemiology & Infection, 2018, 146(7): 809-816.
- [50] Capistrán C, Constandse C, Ramos-Francia M. Multi-horizon inflation forecasts using disaggregated data[J]. Economic Modelling, 2010, 27(3): 666-677.
- [51] Sulakshin S S. A Quantitative Political Spectrum and Forecasting of Social Evolution.[J]. International Journal of Interdisciplinary Social Sciences, 2010, 5(4).



17010226

- [52] Blondel V D, Guillaume J L, Lambiotte R, et al. Fast unfolding of communities in large networks[J]. Journal of statistical mechanics: theory and experiment, 2008, 2008(10): P10008.
- [53] Kaniuth E, Lau A T M. Fourier and Fourier-Stieltjes algebras on locally compact groups: vol. 231[M]. American Mathematical Soc., 2018.
- [54] 汪荣贵, 王晓华, 杨娟, 李书杰编著. 离散数学及其应用[M]. 北京机械工业出版社, 2017.
- [55] Skarding J, Gabrys B, Musial K. Foundations and modelling of dynamic networks using Dynamic Graph Neural Networks: A survey[J]. CoRR, 2020, abs/2005.07496.
- [56] Cao Y, Wang X, He X, et al. Unifying Knowledge Graph Learning and Recommendation: Towards a Better Understanding of User Preferences[C]. in: WWW. ACM, 2019: 151-161.
- [57] Yu W, Cheng W, Aggarwal C C, et al. NetWalk: A Flexible Deep Embedding Approach for Anomaly Detection in Dynamic Networks[C]. in: KDD. ACM, 2018: 2672-2681.
- [58] Do K, Tran T, Venkatesh S. Graph Transformation Policy Network for Chemical Reaction Prediction[C]. in: KDD. ACM, 2019: 750-760.
- [59] Tenenbaum J B, Silva V D, Langford J C. A Global Geometric Framework for Nonlinear Dimensionality Reduction[J]. Science, 2000, 290(5500): 2319-2323.
- [60] Roweis S, Saul L. Nonlinear Dimensionality Reduction by Locally Linear Embedding[J]. Science, 2000, 290(5500): 2323-2326.
- [61] Belkin M, Niyogi P. Laplacian Eigenmaps for Dimensionality Reduction and Data Representation[J]. MIT Press, 2003.
- [62] Mikolov T, Chen K, Corrado G, et al. Efficient Estimation of Word Representations in Vector Space[C]. in: ICLR (Workshop Poster). 2013.
- [63] Mikolov T, Sutskever I, Chen K, et al. Distributed Representations of Words and Phrases and their Compositionality[C]. in: NIPS. 2013: 3111-3119.
- [64] Perozzi B, Al-Rfou R, Skiena S. DeepWalk: online learning of social representations[C]. in: KDD. ACM, 2014: 701-710.



17010226

- [65] Grover A, Leskovec J. node2vec: Scalable feature learning for networks[C]. in: Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining. 2016: 855-864.
- [66] Tang J, Qu M, Wang M, et al. LINE: Large-scale Information Network Embedding[C]. in: WWW. ACM, 2015: 1067-1077.
- [67] Wang D, Cui P, Zhu W. Structural Deep Network Embedding[C]. in: KDD. ACM, 2016: 1225-1234.
- [68] Wang X, Cui P, Wang J, et al. Community preserving network embedding.[C]. in: AAAI: vol. 17. 2017: 203-209.
- [69] Qiu J, Dong Y, Ma H, et al. Network embedding as matrix factorization: Unifying deepwalk, line, pte, and node2vec[C]. in: Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining. 2018: 459-467.
- [70] Qiu J, Dong Y, Ma H, et al. Netsmf: Large-scale network embedding as sparse matrix factorization[C]. in: The World Wide Web Conference. 2019: 1509-1520.
- [71] Bruna J, Zaremba W, Szlam A, et al. Spectral Networks and Locally Connected Networks on Graphs[C]. in: ICLR. 2014.
- [72] Defferrard M, Bresson X, Vandergheynst P. Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering[C]. in: NIPS. 2016: 3837-3845.
- [73] Kipf T N, Welling M. Semi-Supervised Classification with Graph Convolutional Networks[C]. in: ICLR (Poster). OpenReview.net, 2017.
- [74] Xu K, Hu W, Leskovec J, et al. How Powerful are Graph Neural Networks? [J/OL]. CoRR, 2018, abs/1810.00826. arXiv: 1810.00826. <http://arxiv.org/abs/1810.00826>.
- [75] Velickovic P, Cucurull G, Casanova A, et al. Graph Attention Networks[J/OL]. CoRR, 2017, abs/1710.10903. arXiv: 1710.10903. <http://arxiv.org/abs/1710.10903>.
- [76] Bresson X, Laurent T. Residual Gated Graph ConvNets[J]. CoRR, 2017, abs/1711.07553.
- [77] Hamilton W L, Ying Z, Leskovec J. Inductive Representation Learning on Large Graphs[C]. in: NIPS. 2017: 1024-1034.



17010226

- [78] Chen J, Ma T, Xiao C. FastGCN: Fast Learning with Graph Convolutional Networks via Importance Sampling[C]. in: ICLR (Poster). OpenReview.net, 2018.
- [79] Chiang W, Liu X, Si S, et al. Cluster-GCN: An Efficient Algorithm for Training Deep and Large Graph Convolutional Networks[C]. in: KDD. ACM, 2019: 257-266.
- [80] Zeng H, Zhou H, Srivastava A, et al. GraphSAINT: Graph Sampling Based Inductive Learning Method[C]. in: ICLR. OpenReview.net, 2020.
- [81] Xu K, Li C, Tian Y, et al. Representation Learning on Graphs with Jumping Knowledge Networks[C]. in: Proceedings of Machine Learning Research: ICML: vol. 80. PMLR, 2018: 5449-5458.
- [82] Li P, Wang Y, Wang H, et al. Distance Encoding: Design Provably More Powerful Neural Networks for Graph Representation Learning[C]. in: NeurIPS. 2020.
- [83] You J, Ying R, Leskovec J. Position-aware Graph Neural Networks[C]. in: Proceedings of Machine Learning Research: ICML: vol. 97. PMLR, 2019: 7134-7143.
- [84] Nikolentzos G, Vazirgiannis M. Random Walk Graph Neural Networks[C]. in: NeurIPS. 2020.
- [85] Zhang C, Bi J, Xu S, et al. Multi-Imbalance: An open-source software for multi-class imbalance learning[J]. Knowl. Based Syst., 2019, 174: 137-143.
- [86] Lemaitre G, Nogueira F, Aridas C K. Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning[J]. J. Mach. Learn. Res., 2017, 18: 17:1-17:5.
- [87] Ilonen J, Paalanen P, Kamarainen J, et al. Gaussian mixture pdf in one-class classification: computing and utilizing confidence values[C]. in: ICPR (2). IEEE Computer Society, 2006: 577-580.
- [88] Yeung D, Chow C. Parzen-Window Network Intrusion Detectors[C]. in: ICPR (4). IEEE Computer Society, 2002: 385-388.
- [89] Breunig M M, Kriegel H, Ng R T, et al. LOF: Identifying Density-Based Local Outliers[C]. in: ACM SIGMOD International Conference on Management of Data (SIGMOD). ACM, 2000: 93-104.



17010226

- [90] Liu F T, Ting K M, Zhou Z. Isolation Forest[C]. in: ICDM. IEEE Computer Society, 2008: 413-422.
- [91] Tax D M J, Duin R P W. Support Vector Data Description[J]. Mach. Learn., 2004, 54(1): 45-66.
- [92] Scholkopf B, Platt J C, Shawe-Taylor J, et al. Estimating the Support of a High-Dimensional Distribution[J]. Neural Computation, 2001, 13(7): 1443-1471.
- [93] Olive D J. Principal Component Analysis[M]. in: Robust Multivariate Analysis. Springer, 2017: 189-217.
- [94] Candès E J, Li X, Ma Y, et al. Robust principal component analysis?[J]. J. ACM, 2011, 58(3): 11:1-11:37.
- [95] Wang C, Wang J, Wang C, et al. Actor Model Anomaly Detection Using Kernel Principal Component Analysis[C]. in: Lecture Notes in Computer Science: ICONIP (4): vol. 11304. Springer, 2018: 545-554.
- [96] Zhou C, Paffenroth R C. Anomaly Detection with Robust Deep Autoencoders[C]. in: KDD. ACM, 2017: 665-674.
- [97] Chen J, Sathe S, Aggarwal C C, et al. Outlier Detection with Autoencoder Ensembles[C]. in: SDM. SIAM, 2017: 90-98.
- [98] Kingma D P, Welling M. Auto-Encoding Variational Bayes[C]. in: International Conference on Learning Representations (ICLR). 2014.
- [99] Wang X, Du Y, Lin S, et al. adVAE: A self-adversarial variational autoencoder with Gaussian anomaly prior knowledge for anomaly detection[J]. Knowl. Based Syst., 2020, 190: 105187.
- [100] Goodfellow I J, Pouget-Abadie J, Mirza M, et al. Generative Adversarial Nets[C]. in: Annual Conference on Neural Information Processing Systems (NeurIPS). MIT Press, 2014: 2672-2680.
- [101] Liu Y, Li Z, Zhou C, et al. Generative Adversarial Active Learning for Unsupervised Outlier Detection[J]. IEEE Trans. Knowl. Data Eng., 2020, 32(8): 1517-1528.
- [102] Ruff L, Görnitz N, Deecke L, et al. Deep One-Class Classification[C]. in: Proceedings of Machine Learning Research: ICML: vol. 80. PMLR, 2018: 4390-4399.



17010226

- [103] Osada G, Omote K, Nishide T. Network Intrusion Detection Based on Semi-supervised Variational Auto-Encoder[C]. in: Lecture Notes in Computer Science: ESORICS (2): vol. 10493. Springer, 2017: 344-361.
- [104] Abdallah A, Maarof M A, Zainal A. Fraud detection system: A survey[J]. J. Netw. Comput. Appl., 2016, 68: 90-113.
- [105] 侯世旺, 同淑荣. 基于神经网络——模糊集的多元过程质量异常检测及分类[J]. 制造业自动化, 2008, 30(2): 5.
- [106] Schlegl T, Seeböck P, Waldstein S M, et al. Unsupervised Anomaly Detection with Generative Adversarial Networks to Guide Marker Discovery[C]. in: Lecture Notes in Computer Science: IPMI: vol. 10265. Springer, 2017: 146-157.
- [107] Akcay S, Atapour-Abarghouei A, Breckon T P. Gandomaly: Semi-supervised anomaly detection via adversarial training[C]. in: Asian conference on computer vision. 2018: 622-637.
- [108] Gao J, Liang F, Fan W, et al. On community outliers and their efficient detection in information networks[C]. in: KDD. ACM, 2010: 813-822.
- [109] Perozzi B, Akoglu L. Scalable Anomaly Ranking of Attributed Neighborhoods [C]. in: SDM. SIAM, 2016: 207-215.
- [110] Sánchez P I, Müller E, Laforet F, et al. Statistical Selection of Congruent Subspaces for Mining Attributed Graphs[C]. in: ICDM. IEEE Computer Society, 2013: 647-656.
- [111] Li J, Dani H, Hu X, et al. Radar: Residual Analysis for Anomaly Detection in Attributed Networks[C]. in: IJCAI. ijcai.org, 2017: 2152-2158.
- [112] Peng Z, Luo M, Li J, et al. ANOMALOUS: A Joint Modeling Approach for Anomaly Detection on Attributed Networks[C]. in: IJCAI. ijcai.org, 2018: 3513-3519.
- [113] Ding K, Li J, Bhanushali R, et al. Deep Anomaly Detection on Attributed Networks[C]. in: SDM. SIAM, 2019: 594-602.
- [114] Li Y, Huang X, Li J, et al. SpecAE: Spectral AutoEncoder for Anomaly Detection in Attributed Networks[C]. in: CIKM. ACM, 2019: 2233-2236.
- [115] Adamic L A, Adar E. Friends and neighbors on the Web[J]. Soc. Networks, 2003, 25(3): 211-230.



17010226

- [116] Bell R M, Koren Y. Lessons from the Netflix prize challenge[J]. SIGKDD Explor., 2007, 9(2): 75-79.
- [117] Qi Y, Bar-Joseph Z, Klein-Seetharaman J. Evaluation of different biological data and computational classification methods for use in protein interaction prediction [J]. Proteins: Structure, Function, and Bioinformatics, 2006, 63(3): 490-500.
- [118] Stanfield Z, Coşkun M, Koyutürk M. Drug response prediction as a link prediction problem[J]. Scientific reports, 2017, 7(1): 1-13.
- [119] Liben-Nowell D, Kleinberg J. The link-prediction problem for social networks[J]. Journal of the American society for information science and technology, 2007, 58(7): 1019-1031.
- [120] Barabási A L, Albert R. Emergence of scaling in random networks[J]. science, 1999, 286(5439): 509-512.
- [121] Katz L. A new status index derived from sociometric analysis[J]. Psychometrika, 1953, 18(1): 39-43.
- [122] Ma X, Sun P, Wang Y. Graph regularized nonnegative matrix factorization for temporal link prediction in dynamic networks[J]. Physica A: Statistical mechanics and its applications, 2018, 496: 121-136.
- [123] Kipf T N, Welling M. Variational Graph Auto-Encoders[J]. CoRR, 2016, abs/1611.07308.
- [124] Zhang M, Chen Y. Link Prediction Based on Graph Neural Networks[C]. in: NeurIPS. 2018: 5171-5181.
- [125] Chami I, Ying Z, Ré C, et al. Hyperbolic Graph Convolutional Neural Networks [C]. in: NeurIPS. 2019: 4869-4880.
- [126] Nguyen G H, Lee J B, Rossi R A, et al. Continuous-time dynamic network embeddings[C]. in: Companion Proceedings of the The Web Conference 2018. 2018: 969-976.
- [127] Wang Y, Chang Y Y, Liu Y, et al. Inductive Representation Learning in Temporal Networks via Causal Anonymous Walks[C/OL]. in: International Conference on Learning Representations. 2021. <https://openreview.net/forum?id=KYPz4YsCPj>.



17010226

- [128] Chen X, Xu H, Zhang Y, et al. Sequential Recommendation with User Memory Networks[C]. in: WSDM. ACM, 2018: 108-116.
- [129] Scargle J D. An Introduction to the Theory of Point Processes, Vol. I: Elementary Theory and Methods[J]. Technometrics, 2004, 46(2): 257.
- [130] 张国明, 王俊淑, 江南, 等. 关注点推荐算法的霍克斯过程法[J]. 测绘学报, 2018, 47(9): 9.
- [131] Aalen O, Borgan O, Gjessing H. Survival and event history analysis: a process point of view[M]. Springer Science & Business Media, 2008.
- [132] Wang Y, Shen H, Liu S, et al. Cascade Dynamics Modeling with Attention-based Recurrent Neural Network[C / OL]. in: Sierra C. Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017. ijcai.org, 2017: 2985-2991. <https://doi.org/10.24963/ijcai.2017/416>. DOI: 10.24963/ijcai.2017/416.
- [133] Xiao S, Yan J, Yang X, et al. Modeling the Intensity Function of Point Process Via Recurrent Neural Networks[C / OL]. in: Singh S P, Markovitch S. Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA. AAAI Press, 2017: 1597-1603. <http://aaai.org/o cs/index.php/AAAI/AAAI17/paper/view/14391>.
- [134] Shchur O, Türkmen A C, Januschowski T, et al. Neural Temporal Point Processes: A Review[C]. in: IJCAI. ijcai.org, 2021: 4585-4593.
- [135] Xiao S, Farajtabar M, Ye X, et al. Wasserstein learning of deep generative point process models[J]. arXiv preprint arXiv:1705.08051, 2017.
- [136] Olkin I, Pukelsheim F. The distance between two random vectors with given dispersion matrices[J]. Linear Algebra and its Applications, 1982, 48(1): 257-263.
- [137] Li S, Xiao S, Zhu S, et al. Learning Temporal Point Processes via Reinforcement Learning[C]. in: NeurIPS. 2018: 10804-10814.
- [138] Wu Q, Zhang Z, Gao X, et al. Learning Latent Process from High-Dimensional Event Sequences via Efficient Sampling[C]. in: NeurIPS. 2019: 3842-3851.
- [139] Jaeger H. Tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the "echo state network" approach: vol. 5[M]. GMD-Forschungszentrum Informationstechnik Bonn, 2002.



17010226

- [140] Choi Y, El-Khamy M, Lee J. Universal Deep Neural Network Compression[J]. IEEE J. Sel. Top. Signal Process., 2020, 14(4): 715-726.
- [141] Blalock D W, Ortiz J J G, Frankle J, et al. What is the State of Neural Network Pruning?[C]. in: MLSys. mlsys.org, 2020.
- [142] Zhou Y, Moosavi-Dezfooli S, Cheung N, et al. Adaptive Quantization for Deep Neural Network[C]. in: AAAI. AAAI Press, 2018: 4596-4604.
- [143] Gou J, Yu B, Maybank S J, et al. Knowledge Distillation: A Survey[J]. Int. J. Comput. Vis., 2021, 129(6): 1789-1819.
- [144] Zhang B, Zeng H, Prasanna V K. Hardware Acceleration of Large Scale GCN Inference[C]. in: ASAP. IEEE, 2020: 61-68.
- [145] Mittal S. A survey of FPGA-based accelerators for convolutional neural networks [J]. Neural Comput. Appl., 2020, 32(4): 1109-1139.
- [146] Yan M, Deng L, Hu X, et al. HyGCN: A GCN Accelerator with Hybrid Architecture[C]. in: HPCA. IEEE, 2020: 15-29.
- [147] Li J, Louri A, Karanth A, et al. GCNAX: A Flexible and Energy-efficient Accelerator for Graph Convolutional Neural Networks[C]. in: HPCA. IEEE, 2021: 775-788.
- [148] Paszke A, Gross S, Massa F, et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library[C]. in: NeurIPS. 2019: 8024-8035.
- [149] Pang B, Nijkamp E, Wu Y N. Deep learning with tensorflow: A review[J]. Journal of Educational and Behavioral Statistics, 2020, 45(2): 227-248.
- [150] Ham T J, Wu L, Sundaram N, et al. Graphicionado: A high-performance and energy-efficient accelerator for graph analytics[C]. in: MICRO. IEEE Computer Society, 2016: 56:1-56:13.
- [151] Wang Y, Davidson A A, Pan Y, et al. Gunrock: a high-performance graph processing library on the GPU[C]. in: PPoPP. ACM, 2016: 11:1-11:12.
- [152] Peng H, Li J, Gong Q, et al. Fine-grained Event Categorization with Heterogeneous Graph Convolutional Networks[C]. in: IJCAI. ijcai.org, 2019: 3238-3245.
- [153] Yu B, Yin H, Zhu Z. Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting[C]. in: IJCAI. ijcai.org, 2018: 3634-3640.



17010226

- [154] Wang D, Qi Y, Lin J, et al. A Semi-Supervised Graph Attentive Network for Financial Fraud Detection[C]. in: ICDM. IEEE, 2019: 598-607.
- [155] Ma X, Shi W. AESMOTE: Adversarial Reinforcement Learning With SMOTE for Anomaly Detection[J]. IEEE Trans. Netw. Sci. Eng., 2021, 8(2): 943-956.
- [156] Chen Z, Zhou L, Yu W. ADASYN-Random Forest Based Intrusion Detection Model[C]. in: SPML. ACM, 2021: 152-159.
- [157] Ali-Gombe A, Elyan E. MFC-GAN: Class-imbalanced dataset classification using Multiple Fake Class Generative Adversarial Network[J]. Neurocomputing, 2019, 361: 212-221.
- [158] Ribeiro L F R, Saverese P H P, Figueiredo D R. struc2vec: Learning Node Representations from Structural Identity[C]. in: KDD. ACM, 2017: 385-394.
- [159] 李亚亚, 王昌. 希尔伯特空间诞生探源[J]. 自然辩证法研究, 2013, 29(12): 5.
- [160] Glorot X, Bordes A, Bengio Y. Deep Sparse Rectifier Neural Networks[C]. in: JMLR Proceedings: AISTATS: vol. 15. JMLR.org, 2011: 315-323.
- [161] Glorot X, Bengio Y. Understanding the difficulty of training deep feedforward neural networks[C]. in: JMLR Proceedings: AISTATS: vol. 9. JMLR.org, 2010: 249-256.
- [162] Sen P, Namata G, Bilgic M, et al. Collective Classification in Network Data[J]. AI Magazine, 2008, 29(3): 93-106.
- [163] 孙孟柯, 张红梅. 基于 Bag of words 模型的图像检索系统的设计与实现[J]. 电脑知识与技术, 2012(2X): 4.
- [164] Andrei A, Dingwall A, Dillon T, et al. Developing a Tagalog Linguistic Inquiry and Word Count (LIWC) 'Disaster' Dictionary for Understanding Mixed Language Social Media: A Work-in-Progress Paper[C]. in: LaTeCH@EACL. The Association for Computer Linguistics, 2014: 91-94.
- [165] Yao Y, Rosasco L, Caponnetto A. On early stopping in gradient descent learning [J]. Constructive Approximation, 2007, 26(2): 289-315.
- [166] Loshchilov I, Hutter F. Decoupled Weight Decay Regularization[C]. in: ICLR. 2019.



17010226

- [167] Srivastava N, Hinton G E, Krizhevsky A, et al. Dropout: a simple way to prevent neural networks from overfitting[J]. *J. Mach. Learn. Res.*, 2014, 15(1): 1929-1958.
- [168] Maaten L v d, Hinton G. Visualizing data using t-SNE[J]. *Journal of machine learning research*, 2008, 9(Nov): 2579-2605.
- [169] 薄树奎, 李盛阳, 朱重光, 等. 基于统计学的最近邻查询中维数灾难的研究[J]. *计算机工程*, 2006, 32(21): 6-8.
- [170] Kazemi S M, Goel R, Jain K, et al. Representation Learning for Dynamic Graphs: A Survey[J]. *J. Mach. Learn. Res.*, 2020, 21: 70:1-70:73.
- [171] Wang P, Xu B, Wu Y, et al. Link prediction in social networks: the state-of-the-art [J]. *Sci. China Inf. Sci.*, 2015, 58(1): 1-38.
- [172] Martínez V, Berzal F, Talavera J C C. A Survey of Link Prediction in Complex Networks[J]. *ACM Comput. Surv.*, 2017, 49(4): 69:1-69:33.
- [173] Sak H, Senior A W, Beaufays F. Long short-term memory recurrent neural network architectures for large scale acoustic modeling[C]. in: INTERSPEECH. ISCA, 2014: 338-342.
- [174] Vaswani A, Shazeer N, Parmar N, et al. Attention is All you Need[C]. in: NIPS. 2017: 5998-6008.
- [175] Vaswani A, Shazeer N, Parmar N, et al. Attention is All you Need[C]. in: NIPS. 2017: 5998-6008.
- [176] Ba L J, Kiros J R, Hinton G E. Layer Normalization[J]. *CoRR*, 2016, abs/1607.06450.
- [177] Bordes A, Usunier N, Garcia-Duran A, et al. Translating Embeddings for Modeling Multi-relational Data[C]. in: NIPS. 2013: 2787-2795.
- [178] Loshchilov I, Hutter F. Decoupled Weight Decay Regularization[C]. in: ICLR (Poster). OpenReview.net, 2019.
- [179] Hoerl A E, Kennard R W. Ridge Regression: Biased Estimation for Nonorthogonal Problems[J]. *Technometrics*, 2000, 42(1): 80-86.
- [180] Madan A, Cebrian M, Moturu S, et al. Sensing the" health state" of a community [J]. *IEEE Pervasive Computing*, 2011, 11(4): 36-45.



17010226

- [181] Hawkes A G. Spectra of some self-exciting and mutually exciting point processes [J]. *Biometrika*, 1971, 58(1): 83-90.
- [182] Savage D, Wang Q, Zhang X, et al. Detection of Money Laundering Groups: Supervised Learning on Small Networks[C]. in: AAAI Technical Report: AAAI Workshops: vol. WS-17. AAAI Press, 2017.
- [183] Das A K, Mishra S, Gopalan S S. Predicting CoVID-19 community mortality risk using machine learning and development of an online prognostic tool[J]. *PeerJ*, 2020, 8: e10083.
- [184] Gu Y, Sun Y, Gao J. The Co-Evolution Model for Social Network Evolving and Opinion Migration[C]. in: KDD. ACM, 2017: 175-184.
- [185] Rossetti G, Cazabet R. Community Discovery in Dynamic Networks: A Survey [J]. *ACM Comput. Surv.*, 2018, 51(2): 35:1-35:37.
- [186] Xu D, Ruan C, Korpeoglu E, et al. Self-attention with Functional Time Representation Learning[C]. in: NeurIPS. 2019: 15889-15899.
- [187] Rahimi A, Recht B. Random Features for Large-Scale Kernel Machines[C]. in: NIPS. Curran Associates, Inc., 2007: 1177-1184.
- [188] Çinlar E, Agnew R A. On the Superposition of Point Processes[J/OL]. *Journal of the Royal Statistical Society: Series B (Methodological)*, 1968, 30(3): 576-581. eprint: <https://rss.onlinelibrary.wiley.com/doi/pdf/10.1111/j.2517-6161.1968.tb00758.x>. <https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.2517-6161.1968.tb00758.x>. DOI: <https://doi.org/10.1111/j.2517-6161.1968.tb00758.x>.
- [189] Cho K, van Merriënboer B, Gulcehre C, et al. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation[C/OL]. in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Doha, Qatar: Association for Computational Linguistics, 2014: 1724-1734. <https://www.aclweb.org/anthology/D14-1179>. DOI: 10.3115/v1/D14-1179.
- [190] Sutskever I, Vinyals O, Le Q V. Sequence to Sequence Learning with Neural Networks[C/OL]. in: Ghahramani Z, Welling M, Cortes C, et al. Advances in Neural Information Processing Systems: vol. 27. Curran Associates, Inc., 2014. <https://proceedings.neurips.cc/paper/2014/file/a14ac55a4f27472c5d894ec1c3c743d2-Paper.pdf>.



17010226

- [191] Traag V A, Waltman L, van Eck N J. From Louvain to Leiden: guaranteeing well-connected communities[J]. CoRR, 2018, abs/1810.08473.
- [192] 何天文, 王红. 基于语义语法分析的中文语句困惑度评价[J]. 计算机应用研究, 2017, 34(12): 6.
- [193] Henaff M, Bruna J, LeCun Y. Deep Convolutional Networks on Graph-Structured Data[J]. CoRR, 2015, abs/1506.05163.
- [194] Atwood J, Towsley D. Diffusion-Convolutional Neural Networks[C]. in: NIPS. 2016: 1993-2001.
- [195] Zhuang C, Ma Q. Dual Graph Convolutional Networks for Graph-Based Semi-Supervised Classification[C]. in: WWW. ACM, 2018: 499-508.
- [196] Gilmer J, Schoenholz S S, Riley P F, et al. Neural Message Passing for Quantum Chemistry[C]. in: Proceedings of Machine Learning Research: ICML: vol. 70. PMLR, 2017: 1263-1272.
- [197] Ying Z, You J, Morris C, et al. Hierarchical Graph Representation Learning with Differentiable Pooling[C]. in: NeurIPS. 2018: 4805-4815.
- [198] Simonovsky M, Komodakis N. Dynamic Edge-Conditioned Filters in Convolutional Neural Networks on Graphs[C]. in: CVPR. IEEE Computer Society, 2017: 29-38.
- [199] Schlichtkrull M S, Kipf T N, Bloem P, et al. Modeling Relational Data with Graph Convolutional Networks[C]. in: Lecture Notes in Computer Science: ESWC: vol. 10843. Springer, 2018: 593-607.
- [200] Ying R, He R, Chen K, et al. Graph Convolutional Neural Networks for Web-Scale Recommender Systems[C]. in: KDD. ACM, 2018: 974-983.
- [201] Wu F, Jr. A H S, Zhang T, et al. Simplifying Graph Convolutional Networks[C]. in: Proceedings of Machine Learning Research: ICML: vol. 97. PMLR, 2019: 6861-6871.
- [202] Zhang Z, Cui P, Zhu W. Deep Learning on Graphs: A Survey[J]. CoRR, 2018, abs/1812.04202.
- [203] Sengupta J, Kubendran R, Neftci E, et al. High-speed, real-time, spike-based object tracking and path prediction on google edge TPU[C]. in: 2020 2nd IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS).



17010226

2020: 134-135.

- [204] Ioffe S, Szegedy C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift[C]. in: JMLR Workshop and Conference Proceedings: ICML: vol. 37. JMLR.org, 2015: 448-456.
- [205] Huang W, Zhang T, Rong Y, et al. Adaptive Sampling Towards Fast Graph Representation Learning[C]. in: NeurIPS. 2018: 4563-4572.
- [206] Liu Z, Wu Z, Zhang Z, et al. Bandit Samplers for Training Graph Neural Networks [C]. in: NeurIPS. 2020.
- [207] Han J, Moraga C. The Influence of the Sigmoid Function Parameters on the Speed of Backpropagation Learning[C]. in: Lecture Notes in Computer Science: IWANN: vol. 930. Springer, 1995: 195-201.
- [208] Fawcett T. An introduction to ROC analysis[J]. Pattern Recognit. Lett., 2006, 27(8): 861-874.
- [209] Lobo J M, Jiménez-Valverde A, Real R. AUC: a misleading measure of the performance of predictive distribution models[J]. Global ecology and Biogeography, 2008, 17(2): 145-151.



17010226

## 致 谢

时光如梭，在上海交通大学五年多的博士学习生活即将结束，回首这些年的点点滴滴，既有困境，也有收获，心中不禁充满无限感慨，感觉一切皆如昨日。在博士论文完成之际，我要向辛勤教育和指导我的导师，不断帮助和关心我的各位同学朋友，以及永远支持和鼓励我的女朋友和父母表达最衷心的感谢。

首先，我要对我的导师杨煜普教授表达最真挚的谢意，感谢杨老师多年来对我学习上的谆谆教导和生活上的细致关怀。作为杨老师指导的最后一个博士生，我感到非常的荣幸以及骄傲。这些年我取得的进步和成就与杨老师的悉心教导是分不开的。在五年的博士研究生生活中，杨老师不仅为我提供了一个宽松、自由的研究环境，而且还在学术研究中给予了我悉心的指导和帮助，不断地引导我在科研工作中取得进步。杨教授深厚的学术造诣、开阔的视野和严谨的治学态度给我留下了深刻的印象，这不仅对我读博期间的学习、生活和工作产生了深远的影响，而且将继续指引我在以后的人生路上不断前进。

接下来，我要衷心地感谢实验室的各位师兄师姐和同窗，感谢宫亮、姜腾、张泽涵、李双宏、詹承俊、崔平六位博士师兄对我科研工作上的帮助，感谢秦超、苏梦珂、王一琳、许朝雄、曾萧、黄彬、江文键在我的学习和工作中所给予的关心。与实验室同学的交流令我受益良多，拓宽了自己的知识面。在每周例会上与大家的讨论，既营造了浓厚的学习氛围，也让我意识到自己研究中的不足，给了我更多的启发。作为一个团结友爱、奋发向上的科研集体，我们彼此扶持，相互鼓励，共同提高，建立了深厚的友谊，这份友谊我将永远珍惜。

此外，我还要感谢我的女朋友陆冰晶，她不光在生活中陪伴我，给我带来无数的温暖与欢笑，还在科研上给予了我帮助。她在我最焦虑的时刻鼓励我，在我写论文思绪混乱的时候帮我梳理论点，在我研究小成的时候发自内心地为我开心。点点滴滴我都铭记在心，山水一程，幸之与你相遇，希望与你共度此生。

最后，也是最重要的，我要特别感谢我的父母对我的支持和鼓励，你们自始至终的支持是我最强大的精神支柱。感谢父母多年来对我不求回报的辛勤培养，是你们给了我健康的体魄，教会了我做人的道理。27年的时光一晃而过，我很遗憾因为求学和工作的原因没能一直陪在父母身边，希望将来可以有机会多陪伴你们，我爱你们。



17010226

## 学术论文和科研成果目录

### 学术论文

- [1] 第一作者. Xuhong Wang, et al. "adVAE: A self-adversarial variational autoencoder with Gaussian anomaly prior knowledge for anomaly detection." *Knowledge-Based Systems* 190 (2020): 105187. (SCI 期刊, 中科院分区一区, IF 8.139, h-index 94)
- [2] 第一作者. Xuhong Wang, et al. "One-class graph neural networks for anomaly detection in attributed networks." *Neural computing and applications* 33.18 (2021): 12073-12085. (SCI 期刊, 中科院分区二区, IF 5.102, h-index 57)
- [3] 第一作者. Xuhong Wang, et al. "APAN: Asynchronous propagation attention network for real-time temporal graph embedding." *Proceedings of the 2021 International Conference on Management of Data (SIGMOD)*. 2021. (EI 检索, 中国计算机学会 CCF-A 类会议, h-index 104)
- [4] 第一作者. Xuhong Wang, et al. "Variational autoencoder based fault detection and location method for power distribution network." *2020 8th International Conference on Condition Monitoring and Diagnosis (CMD)*. IEEE, 2020. (EI 检索, 交大评级 A 类会议)
- [5] 第一作者. Xuhong Wang, et al. "Partial discharge pattern recognition with data augmentation based on generative adversarial networks." *2018 Condition Monitoring and Diagnosis (CMD)*. IEEE, 2018. (EI 检索, 交大评级 A 类会议)
- [6] 第一作者(审稿中). Xuhong Wang, et al. "Event Forecasting on Continuous Time Dynamic Graph with Graph Temporal Point Process" *2022 International Conference of Data Mining (CIKM)*. 2022. (EI 检索, CCF-B 类会议)
- [7] 第一作者(审稿中). Xuhong Wang, et al. "Translating Edges to Triangle-wise Embeddings via Temporal Relational Reasoning Network." *IEEE Transactions on Neural Networks and Learning Systems*. (SCI 期刊, 中科院分区一区, IF 13.994)
- [8] 第二作者. Yanyi Chu, Xuhong Wang, et al. "MDA-GCNFTG: identifying miRNA-disease associations based on graph convolutional networks via graph sampling



17010226

through the feature and topology graph.” Briefings in Bioinformatics 22.6 (2021): bbab165. (SCI 期刊, 中科院分区一区, IF 11.622)

- [9] 第二作者. Ping Cui, Xuhong Wang, and Yupu Yang. ”Nonparametric manifold learning approach for improved process monitoring.” The Canadian Journal of Chemical Engineering 100.1 (2022): 67-89. (SCI 期刊, 中科院分区四区, IF 2.5)
- [10] 第二作者. Ping Cui, Xuhong Wang, and Yupu Yang. ”Statistics manifold learning approach and its application to non-Gaussian process monitoring.” 2020 39th Chinese Control Conference (CCC). IEEE, 2020. (EI 检索, 交大评级 B 类会议)
- [11] 第二作者. Wenjian Jiang, Xuhong Wang, et al. ”Adaptive Sampling Temporal Neural Network.” 2022 4th International Conference on Advances in Computer Technology, Information Science and Communications (CTISC) , 2022 (EI 检索)
- [12] 第二作者. Bin Huang, Xuhong Wang, et al., ”One-class Temporal Graph Attention Neural Network for Dynamic Graph Anomaly Detection.” 2021 2nd International Conference on Electronics, Communications and Information Technology (CECIT), 2021, pp. 783-790. (EI 检索)
- [13] 第三作者. Shijie Lin, Fang Xu, Xuhong Wang, et al. ”Efficient Spatial-Temporal Normalization of SAE Representation for Event Camera.” IEEE Robotics and Automation Letters 5.3 (2020): 4265-4272. (SCI 期刊, 中科院分区二区, IF 3.741)

### 科研项目

- [14] 非完整数据过程的鲁棒故障检测与故障认知方法 (国家自然科学基金项目: 61273161)
- [15] 局部放电瞬态电磁脉冲形成的微观机理及宏观表征研究 (国家自然科学基金项目:51777122)
- [16] 变电站机器人巡检数据融合及故障诊断系统 (中国南方电网科技项目)



17010226

## 上海交通大学

### 学位论文原创性声明

本人郑重声明：所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的作品成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

学位论文作者签名：汪旭鸿

日期：2022年8月23日

## 上海交通大学

### 学位论文使用授权书

本学位论文作者完全了解学校有关保留、使用学位论文的规定，同意学校保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。

本学位论文属于  公开论文

内部论文， 1年/ 2年/ 3年 解密后适用本授权书。

秘密论文，\_\_\_\_年（不超过10年）解密后适用本授权书。

机密论文，\_\_\_\_年（不超过20年）解密后适用本授权书。

（请在以上方框内打“√”）

学位论文作者签名：汪旭鸿

指导教师签名：杨煜章

日期：2022年8月23日

日期：2022年8月23日



17010226



# 上海交通大学博士学位论文答辩决议书

017032910027

姓名	汪旭鸿	学号	017032910027	所在学科	控制科学与工程
指导教师	杨煜普	答辩日期	2022-08-22	答辩地点	腾讯会议: 595 351 227
论文题目	基于图深度学习的异常检测及动态关系建模				

投票表决结果: 5/5/5 (同意票数/实到委员数/应到委员数) 答辩结论: 通过 未通过

评语和决议:

论文选题于图深度学习方法，开展异常检测及动态关系建模研究，属于当前研究的热点问题，选题具有重要的理论意义和工程应用参考价值。经过作者认真研究，取得如下结果： 1) 提出了一种基于超球面学习的半监督图神经网络框架，解决了图异常检测场景中数据不平衡导致的有偏估计问题； 2) 针对时空协同性不足问题，通过引入时空约束来压缩模型的可行解空间，提出了时空三角闭合约束、时间向量范数单调性约束的时空建模方法； 3) 针对关系演化预测模型在图上的应对不同演化模式时存在的多模态适应能力不足问题，提出了一种基于图神经网络的层次化随机时序点过程模型； 4) 针对图神经网络算法难以实时性部署的问题，提出了异步信息传播注意力网络，提高了网络整体的稳定性和可扩展性。论文工作表明作者掌握了本学科坚实宽广的基础理论和系统深入的专门知识，具有独立从事科研工作的能力。学位论文满足写作和学术规范，答辩过程表述清楚，回答问题正确。

答辩委员会经讨论和无记名投票表决，认为论文达到了博士学位论文的水平，一致同意通过博士论文答辩，建议授予汪旭鸿同学工学博士学位。

2022 年 8 月 22 日

答辩委员会成员签名	职务	姓名	职称	单位	签名
	主席	王磊	教授	同济大学	
	委员	葛万成	教授	同济大学	
	委员	李建勋	教授	上海交通大学电子信息与电气工程学院(自动化系)	
	委员	蔡云泽	研究员	上海交通大学	
	委员	卢俊国	教授	上海交通大学电子信息与电气工程学院(自动化系)	
	秘书	宫亮	高级工程师	上海交通大学	