程序报告

姓名：叶睿宸　　**学号：**22551338　　**学院（系）专业：**软件学院人工智能

1 算法描述

1.1 算法历史演变

图神经网络（GNNs）是一类专门用于处理图结构数据的深度学习模型。它们通过捕捉节点、边和图整体的结构信息，广泛应用于社交网络、推荐系统、生物信息学等领域。以下是 GNN 算法的历史演进过程：

1. 早期阶段：基于图的传统方法

在深度学习兴起之前，图数据的分析主要依赖于传统的图算法，这种方式理论成熟，具有良好的数学基础。在小规模图上表现良好。但同时也存在一定的局限性，需要手工设计特征，难以捕捉复杂的图结构信息。并且无法利用深度学习的强大表示能力，早期传统的基于图的方法主要有如下几种：

PageRank（1998）：用于网页排序的经典算法，基于图的随机游走。

Spectral Clustering（2000s）：基于图拉普拉斯矩阵的谱分解，用于图聚类。

手工特征工程：通过设计图的统计特征（如度、聚类系数、中心性等）作为输入特征。

2. 图神经网络阶段

图神经网络 GNN 的概念最早由 Scarselli 等人在 2005 年提出，目标是通过递归神经网络（RNN）在图结构上进行信息传播。核心思想是通过迭代更新节点的隐藏状态，直到收敛为稳定值。显然这种方式的局限性也很明显，计算复杂度高，难以扩展到大规模图，并且收敛速度慢，训练过程不稳定。

递归式图神经网络（RecGNN）：由 Scarselli 等人提出，目标是通过递归神经网络（RNN）在图结构上进行信息传播。每个节点的隐藏状态通过邻居节点的特征和自身特征迭代更新，直到收敛为稳定值。更新公式： $h_v = f(x_v, \{h_u : u \in \mathbb{N}_v\})$其中，$h_v$是节点 v 的隐藏状态，$\mathbb{N}_v$是节点 v 的邻居。

3. 图卷积网络（GCN）阶段

图卷积神经网络架构 GCN 在 2017 年 Kipf 和 Welling 的《Semi-Supervised Classification with Graph Convolutional Networks》中被提出。

核心思想是将卷积操作推广到图结构上，通过邻居节点的特征聚合更新节点表示。

其公式为：$H^{l+1} = \sigma(\tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2}H^l W^l)$。其中，$\tilde{A}$ 是加了自环的邻接矩阵，$\tilde{D}$ 是度矩阵，H^l是第l层的节点表示。这种方式简单高效，适合半监督学习任务，并且能够捕捉图的局部结构信息。但局限性是只能捕捉固定范围的邻域信息（如 2 层 GCN 只能捕捉 2 阶邻居信息），其次存在过平滑问题（层数过多时，节点表示趋于一致）。

4. 图注意力网络（GAT）阶段

图注意力网络 GAT 的提出（2018）：Velickovic 等人提出的《Graph Attention Networks》。引入注意力机制，为不同邻居分配不同的权重，从而实现更灵活的特征聚合。该种架构能够动态调整邻居节点的权重，提升模型的表达能力。适合异构图（不同类型的节点和边）。缺

陷是计算复杂度较高，尤其在大规模图上。

5. 图采样方法与大规模图学习

随着图规模的增大，传统 GNN 方法在全图上训练的计算成本过高，图采样方法应运而生。

GraphSAGE（2017）：Hamilton 等人提出的《Inductive Representation Learning on Large Graphs》。核心思想是通过采样固定数量的邻居节点进行特征聚合，支持大规模图的归纳学习。优点，支持大规模图的分批训练。能够处理动态图和新节点。

PinSAGE（2018）：用于推荐系统的图采样方法，结合了随机游走和特征聚合。

6. 图神经网络的扩展与优化

异构图神经网络（Heterogeneous GNNs）：处理包含多种节点和边类型的异构图。代表方法，HAN（Heterogeneous Graph Attention Network）。

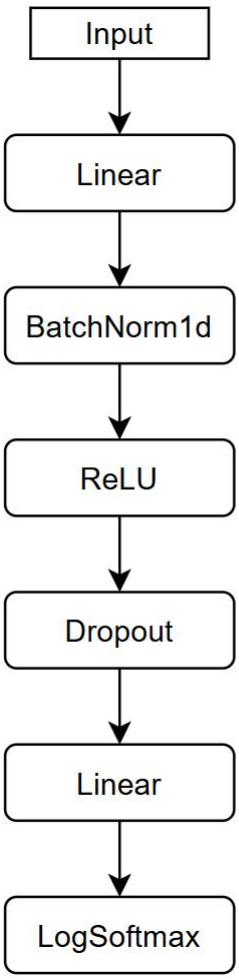
动态图神经网络（Dynamic GNNs）：处理随时间变化的动态图。代表方法：DySAT、TGAT。

图生成模型：用于生成新图或补全缺失的图结构。代表方法：GraphRNN、GraphGAN。

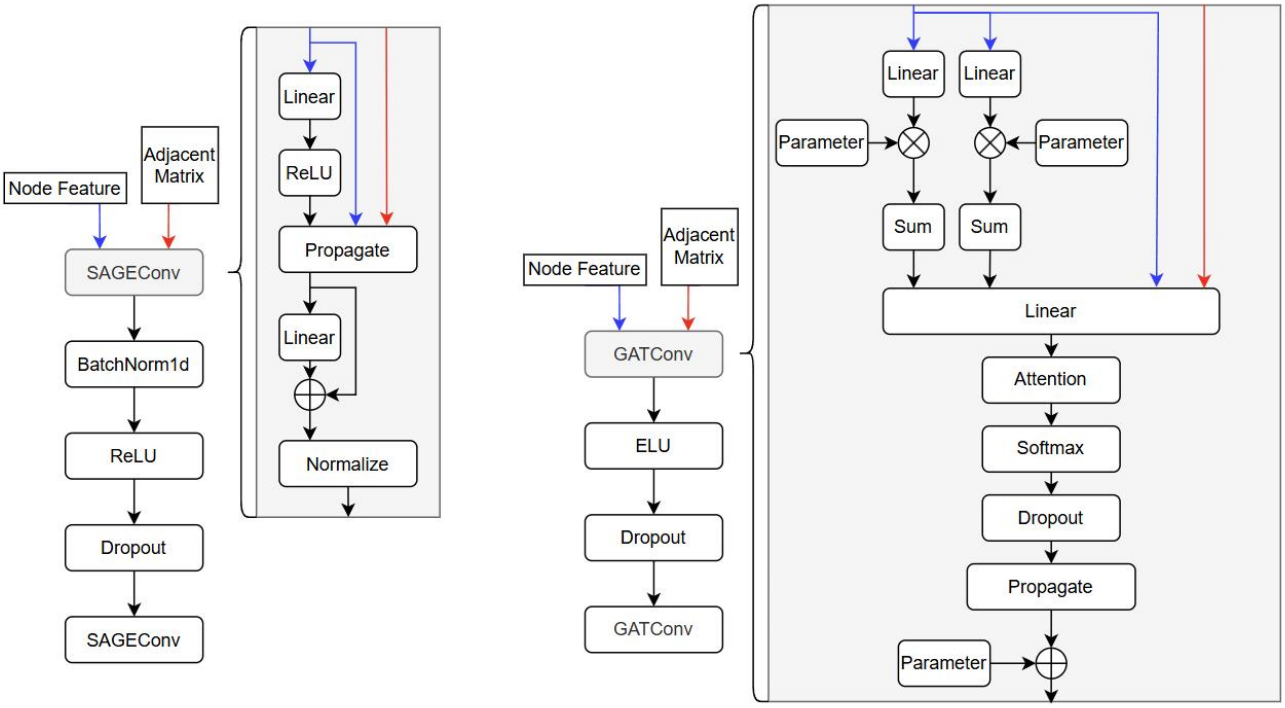
1.2 算法流程图

本次实验主要采用了传统的 MLP 算法，还有实验推荐的 GAT 和 GraphSAGE 算法，这里展示下三种方法的整体架构图：

两层 MLP 模型架构图：



两层 GraphSAGE（左图）和 GAT（右图）模型架构图：



2 算法性能分析

2.1 计算效率

MLP 算法：仅对节点特征进行逐层线性变换和非线性激活，不考虑图的结构信息。计算复杂度为 $O(N \cdot F_{in} \cdot F_{out})$，其中 N 是节点数，$F_{in}$和$F_{out}$分别是输入和输出特征维度。再计算效率方面，高效（由于不涉及图的邻接关系，计算量仅与节点数和特征维度相关），该算法在相同量级规模的任务中，MLP 的效率最高。由于本实验采用 DGraphFin 数据集，有一定数据规模，并且最后提交需要在 CPU 上进行推理，因此 MLP 模型在扩充模型深度也依旧能达到任务对计算速度的要求。在三种算法中计算效率最高。

GAT 算法：使用注意力机制动态计算每条边的权重，捕捉节点与邻居之间的关系。由于涉及注意力矩阵的计算（需要计算每条边的权重），因此复杂度是随机数据集结点个数呈二次方增长的，因此训练和推理过程中的计算效率是相当低的。这种算法适合中小规模图（边数较少的图上），对于本实验的数据集，在训练过程中二层的 GAT 模型无法使用完整的邻接矩阵进行训练，只能通过取点邻近子图的方式进行注意力计算，这样相当于是牺牲了模型的性能去换取训练速度。这种方式的计算效率在三种算法中最低。

GraphSAGE 算法：通过采样固定数量的邻居节点进行特征聚合，避免了全图计算。采样机制可以有效的避免全图计算，显著降低了计算量，适合大规模图（在节点数和边数较多的图上）。GraphSAGE 的效率优于 GAT，在三种算法中处于中间水平。

2.2 准确性

MLP 算法：仅利用节点的特征信息，忽略了图的结构信息。准确性较低，在当前对图

结构信息敏感的金融异常检测任务中，MLP 的表现在三种算法中最差。MLP 比较适用于当图的邻接关系稀疏或无关紧要时，这个时候 MLP 的准确性可能接近 GNN。

GAT 算法：通过注意力机制动态调整邻居节点的权重，能够捕捉更细粒度的图结构信息。准确性较高，特别是在异构图或邻居节点对目标节点影响不均匀的场景中，GAT 的表现较好，比如社交网络、推荐系统等任务中。由于本实验不涉及任何异构图结构，因此在三种算法中最后测得的准确率得分处于中等水平。

GraphSAGE 算法：通过采样邻居节点并聚合特征，能够捕捉局部图结构信息。准确性较高，在大规模图上，GraphSAGE 的准确性接近 GAT，且计算效率更高。该种算法相当于是兼顾了计算效率和准确性。同时这种算法也非常适合大规模图，在节点分类和图嵌入任务中，GraphSAGE 的表现优异。该算法在本任务中最后测得的准确率最高，因为金融异常检测任务对图结构信息敏感，并且不涉及任何异构图，因此 GraphSAGE 在此场景下能达到最佳的性能。

2.3 泛化能力

MLP 算法：MLP 仅依赖节点的特征进行学习，完全忽略图的结构信息。训练时，节点的特征通过全连接层逐层映射到高维空间。该算法对未见节点的泛化较弱，因为 MLP 不利用图结构信息，未见节点的特征必须与训练节点的特征分布相似，否则模型难以泛化。如果节点特征分布发生变化（如新增节点的特征与训练节点差异较大），MLP 的表现会显著下降。对未见图结构的泛化，MLP 算法无泛化能力，因为这种算法完全忽略图结构信息，新增的边或子图不会对模型产生任何影响。对噪声和异常数据的鲁棒性较差，MLP 对节点特征的噪声较为敏感，缺乏图结构信息的约束，容易受到异常值的影响。最终实验表明 MLP 在训练集上的效果比较好，但在测试集上的准确率大幅下降，在训练过程中很容易出现过拟合的现象。

GAT 算法：动态调整邻居节点的权重，能够捕捉更细粒度的图结构信息。泛化能力较强，通过注意力机制，GAT 能够更好地捕捉节点之间的关系，泛化能力较强。对未见图结构的泛化较强，GAT 能够动态调整注意力权重，适应新增的边或子图。在异构图中（不同类型的节点和边），GAT 的注意力机制可以捕捉不同类型的关系，在此场景 GAT 拥有三种算法中最强的泛化能力。对噪声和异常数据的鲁棒性较强，因为注意力机制能够降低噪声节点的影响，但如果图中存在大量异常边，GAT 的性能可能下降。由于本实验的显存或内存有限，在训练过程中无法使用完整的邻接矩阵，因此没有发挥 GAT 完整的泛化能力，该算法在本实验的结果表明容易在训练过程中出现过拟合，最终在测试集上的推理准确度位于三者的中等水平。

GraphSAGE 算法：通过采样邻居节点进行特征聚合，能够捕捉局部和全局的图结构信息。其对未见节点的泛化最强，GraphSAGE 支持归纳学习，能够在训练时未见的节点上生成合理的表示。即使新增节点的特征分布与训练节点不同，GraphSAGE 仍能通过邻居特征生成有效表示。对未见图结构的泛化较强，GraphSAGE 能够适应新增的边或子图，采样机制使其在大规模图上表现良好。并且聚合函数的选择（如均值、最大值）对泛化能力也有一定影响，最终实验采用均值的聚合函数，因为对比下来该聚合函数在测试集上的表现最佳。对噪声和异常数据的鲁棒性较强，采样机制降低了噪声节点的影响，但采样数量和策略对鲁棒性有一定影响。最终选用该算法，在平台取得了 AUC 超过 0.75 的成绩，远超前两种算法最后在平台上的得分。

3 研究展望

GraphSAGE（Graph Sample and Aggregate）是一种高效的图神经网络模型，能够通过采样邻居节点并聚合特征来学习节点表示。它在处理大规模图和动态图方面表现出色，但仍有许多优化和扩展的空间。以下是对 GraphSAGE 模型的研究展望：

3.1 动态采样策略

当前 GraphSAGE 使用固定数量的邻居采样策略，可能导致信息丢失或冗余。可以考虑采用自适应采样，根据节点的重要性或邻居的相关性动态调整采样数量。或是采用层级采样，在不同的网络层中使用不同的采样策略（如浅层采样更多邻居，深层采样更少邻居）。基于注意力的采样，这种方向相当于是将 GraphSAGE 和 GAT 两者的优势结合起来，结合注意力机制，优先采样对目标节点影响较大的邻居。

3.2 聚合函数改进

目前 GraphSAGE 的聚合函数（如均值、最大值、LSTM）在某些任务中可能不足以捕捉复杂的邻居关系。采用图卷积聚合，结合 GCN 的卷积操作，增强聚合函数的表达能力。多模态聚合，在聚合过程中结合节点的多模态特征（如提供文本 Promt 等）。基于 Transformer 结构的聚合，使用 Transformer 的自注意力机制对邻居特征进行加权聚合，进一步提升模型捕捉节点间联系，提升泛化能力和预测准确性的能力。

3.3 模型轻量化

尽管 GraphSAGE 的采样机制降低了计算复杂度，但在超大规模图上仍可能存在性能瓶颈。考虑将模型剪枝加入算法中，对冗余的聚合层或采样节点进行剪枝，减少计算量。或是通过知识蒸馏的方式，通过训练轻量化的学生模型，保留性能的同时降低模型复杂度。最普遍的就是利用量化技术，对模型参数进行量化存储，减少内存占用。

3.4 动态图学习

GraphSAGE 的采样和聚合机制主要针对静态图，无法直接处理随时间变化的动态图。因此如果要在动态图的场景下发挥 GraphSAGE 架构的性能，需要提供时间感知采样，在采样时考虑时间维度，优先选择最近的邻居节点。增量式更新，设计增量式训练方法，仅对新增节点和边进行更新，避免全图重新训练。时序聚合，结合时间信息对邻居特征进行时序建模，捕捉动态变化。

3.5 超大规模图训练策略

在超大规模图（如社交网络、知识图谱）上，GraphSAGE 的采样机制可能面临内存和计算瓶颈导致训练无法完成。进行分布式训练，将图数据分片，利用多 GPU 或多节点进行分布式训练。图分块方法，将大图划分为多个子图，在子图上独立训练并合并结果。在线学习，结合流式数据处理技术，实时更新节点表示。