# A4：图神经网络实验报告 (20 points)

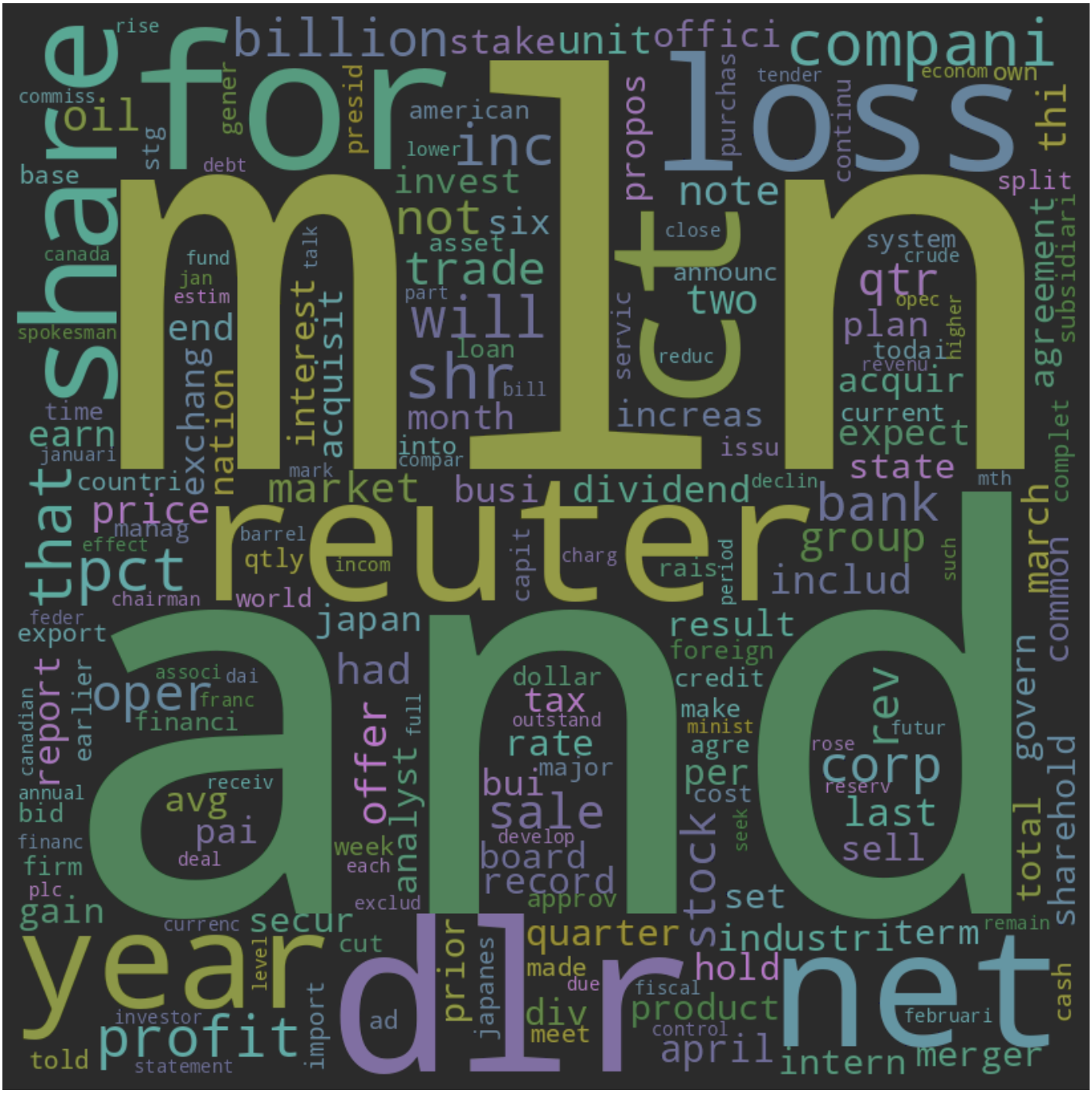
## 一、使用GAT和GCN在r8和r52数据集上进行文本分类

---

### 数据集

本实验采用的`R8`和`R52`数据集是两个常用于文本分类和自然语言处理（NLP）研究的数据集，他们是路透社（Reuters）的新闻数据集的子集。
他们具有不同的类别和文档数量，R8有8类，R52有52类，被广泛用于评估不同文本分类方法的效果。

R8数据集 整体词云图




## 模型结构：

使用图注意力网络(GAT)和图卷积网络(GCN), 详见 `R8+GAT&GCN.ipynb`

## 图数据的构建方式：

- 将每个文档视作一个图结点，所有文档视作一个图。
- 使用TF-IDF(词频-逆文档频率)计算文档的特征，这样比将每一个词视作结点，文档视作图要更加简单
- TF-IDF是一种用于信息检索和文本挖掘的常用加权技术。它是一个统计方法，用以评估一个词语对于一个文件集或一个语料库中的其中一份文件的重要程度。

**一、计算公式**:

1. **词频（TF，Term Frequency）**：
   $\mathrm{TF}(t,d) = \frac{\text{在文档 } d \text{ 中词条 } t \text{ 出现的次数}}{\text{文档 } d \text{ 中的词条总数}}$
2. **逆文档频率（IDF，Inverse Document Frequency）**：
   $\mathrm{IDF}(t,D) = \log \frac{\text{语料库中的文档总数 } |D|}{\text{包含词条 } t \text{ 的文档数目 (即 } df(t,D)) + 1}$

在这里加1是为了避免分母为零（即当词条不出现在语料库中时）。

3. **TF-IDF**:
   $\text{TF-IDF}(t,d,D) = \mathrm{TF}(t,d) \times \mathrm{IDF}(t,D)$
   这里，`t`代表词条，`d`代表文档，`D`代表语料库。

**二、示例计算**:
假设我们有一个文档集合D，包含以下两个文档：

- 文档1: "The car is driven on the road."
- 文档2: "The truck is driven on the highway."

计算词条"car"在文档1中的TF-IDF值的步骤如下：

1. **计算TF**:
   - 在文档1中，"car"这个词出现1次，文档总词数为7（假设我们不考虑停用词如"the", "is"等）。
   - 因此，TF("car", 文档1) = 1/7。
2. **计算IDF**:
   - 语料库D有2个文档，而只有文档1包含词条"car"。
   - 因此，IDF("car", D) = log(2 / 1 + 1) = log(2)。
3. **计算TF-IDF**:
   - TF-IDF("car", 文档1, D) = (1/7) * log(2)。

这个值表明词条"car"在文档1中的重要性。TF-IDF值越高，词条在文档中的重要性越大。

**三、实际应用**:
- 使用在训练集上构建TF-IDF特征，在测试集上使用训练集统计的词频和逆文档频率计算特征。
- 在实际应用中，TF-IDF计算通常利用专门的工具或库来实现，例如Python中的`scikit-learn`库，它提供了一个方便的`TfidfVectorizer`类来计算文档集合的TF-IDF。

## GAT与GCN的机制和公式:

图注意力网络（Graph Attention Networks, GANs）和图卷积网络（Graph Convolutional Networks, GCNs）是两种在图数据上工作的神经网络架构。它们都旨在学习图中节点的特征表示，但它们的实现方式有所不同。

1. 图卷积网络（GCNs）

- GCNs 的核心思想是将卷积操作从传统的欧几里得空间（如图像，时间序列等）扩展到图结构数据。
- 在 GCNs 中，节点的特征更新依赖于它们邻居的特征。每个节点的新特征是其邻居节点特征的加权平均，这个过程类似于在图上的信息传播。

**关键公式**:
一种常见的 GCN 层可以表示为：
$H^{(l+1)} = \sigma(\hat{D}^{-\frac{1}{2}}\hat{A}\hat{D}^{-\frac{1}{2}}H^{(l)}W^{(l)})$
其中：

- $H^{(l)}$ 是第 $l$ 层的节点表示。
- $\hat{A}$ 是添加自循环的邻接矩阵。
- $\hat{D}$ 是 $\hat{A}$的度矩阵。
- $W^{(l)}$是可学习的权重矩阵。
- $\sigma$ 是非线性激活函数，如 ReLU。

2. 图注意力网络（GATs）

- GATs 通过引入注意力机制来加权邻居节点的特征，允许模型自动学习邻居节点对目标节点的重要性。
- 在 GATs 中，每个节点的更新不仅取决于其邻居的特征，而且取决于一个学习到的注意力得分，这个得分反映了邻居节点对当前节点的重要性。

**关键公式**:
GAT 层可以表示为：
$\text{Attention}(h_i, h_j) = \text{softmax}_j(\text{LeakyReLU}(a^T[Wh_i \| Wh_j]))$
$h_i' = \sigma\left(\sum_{j \in \mathcal{N}(i)} \text{Attention}(h_i, h_j) W h_j\right)$
其中：

- $h_i$和 $h_j$ 是节点的原始特征向量。
- $W$ 是可学习的权重矩阵。
- $a$ 是可学习的注意力机制参数。
- $\|$ 表示向量拼接。
- $\mathcal{N}(i)$ 表示节点 $i$ 的邻居。
- $\sigma$ 是非线性激活函数。

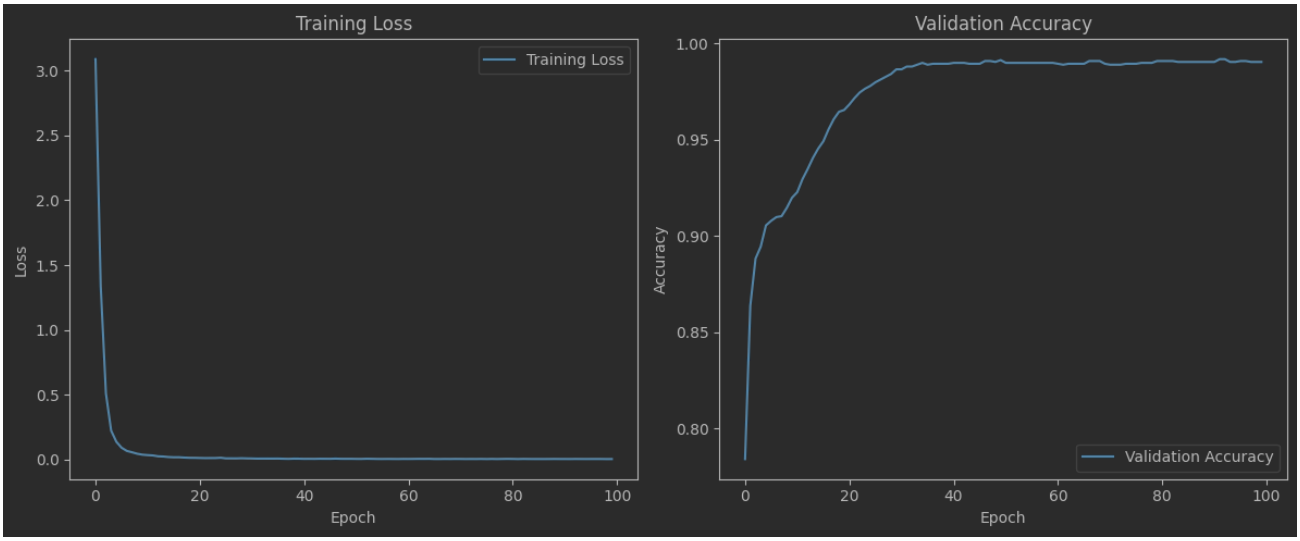
3. 总结
   GAT与GCN的比较：

- GAT: 使用注意力机制来赋予不同邻居不同的重要性。这使得GAT能够更加灵活地捕捉节点间的复杂关系。计算更为复杂，处理不规则图上更具适应性。
- GCN: 没有注意力机制,GCNs 通过传播和聚合邻居的特征来更新节点特征。它使用简化的卷积操作，其中所有邻居对中心节点的贡献均等。计算相较简单，适合处理规则图。

## 训练结果：

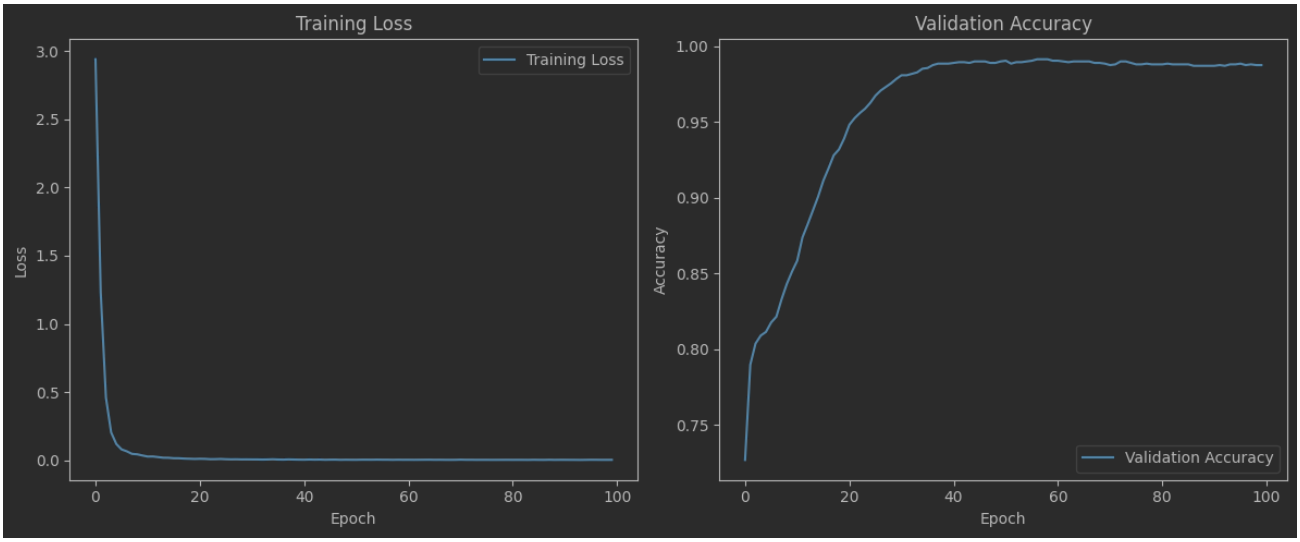
GAT:
使用0.005学习率，训练100个epoch

- 图损失与验证集准确率曲线如下：




- 最佳准确率99.14%
- 模型结果保存为 `model_GAT_state_dict.pt`。
  GCN:
  使用0.005学习率，训练100个epoch

- 图损失与验证集准确率曲线如下：




- 最佳准确率98.23%
- 模型结果保存为`model_GCN_state_dict.pt`。

# 二、附加题：使用lstm在r8和r52数据集上完成文本分类

---

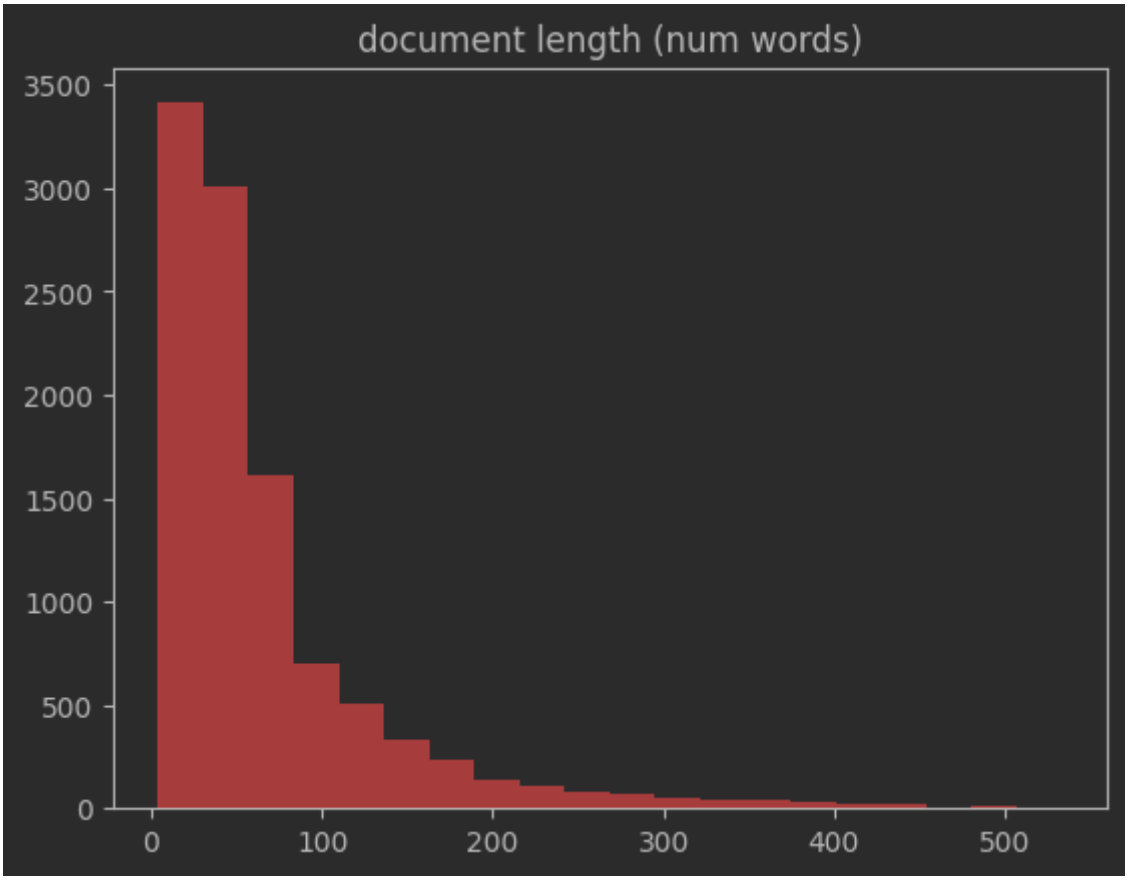
## 实现方法：

用于选取的是序列模型，因此需要调整特征的提取和表示方法。在GNN 的实现中采用了文档级别的特征向量，但是在这个序列模型的实现中使用了词级别的特征，即word2vec方法。
考虑到r8的词汇表中存在大量简写词的情况，不容易找到合适的预训练词向量，在本例中使用随机词嵌入方法，对每个单词生成随机的词向量作为嵌入。

详见附带的jupyter notebook文件`R8+LSTM.ipynb`

## 序列构建：

整个r8数据集中文档的长度直方图

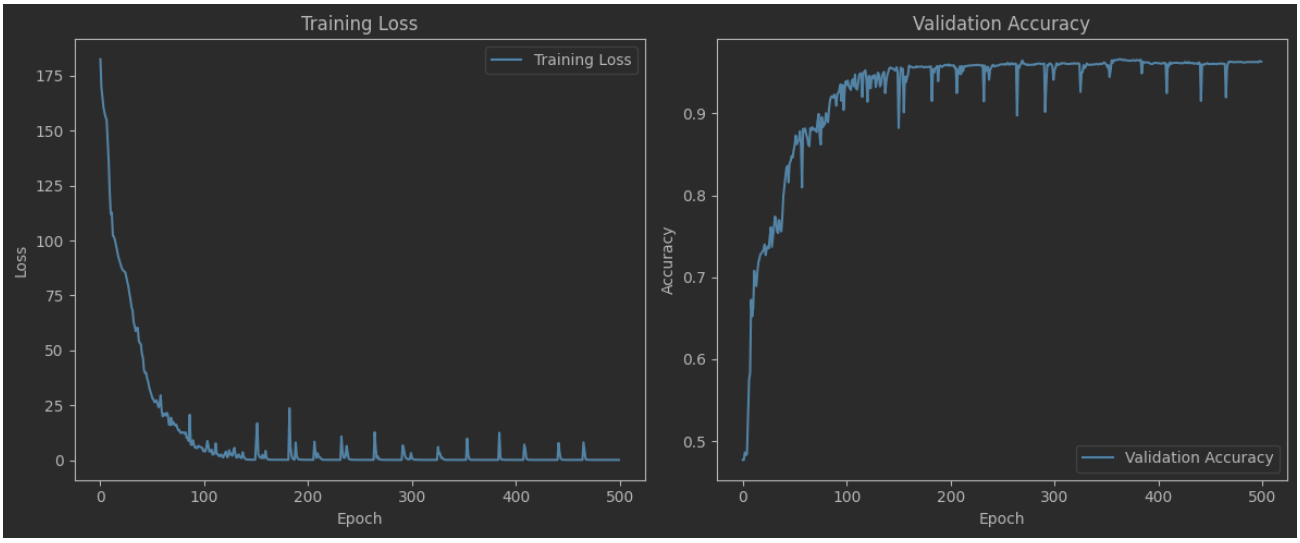



故本实验中将文档序列padding到固定的100。不足的用 0 补齐，过长的截断。

## 训练结果：

使用0.001学习率，训练500个epoch

- 损失与验证集准确率曲线如下：




- 在性能上出现了较大范围的波动
- 模型结果保存为 `model_lstm_state_dict.pt`

### 与GNN实验结果比较分析：

使用词级别的嵌入方法+lstm序列模型大大提升了计算的复杂性，尽管这种方法理论上能够取得更好的分类效果，但是由于模型结构复杂度不够、训练时间不够、嵌入维度过低、嵌入不够合理等诸多因素，本人复现的lstm模型和GNN相比在准确率上还有一些差距。

# 三、异构图数据解决方案设计

要设计一个能够处理包含文本、图像、以及元数据（如作者、发表期刊或会议等）信息的多模态异构文档分类系统，我们可以使用异构图神经网络（Heterogeneous Graph Neural Network, HGNN）整合这些不同类型的数据。它能够处理不同类型的节点和边，从而捕获多模态数据间的复杂关系。

1. 数据预处理

- **文本数据**: 使用预训练的词嵌入模型（如BERT、GloVe）来转换文本数据为向量表示。
- **图像数据**: 采用卷积神经网络（CNN）提取图像特征。
- **元数据**: 将元数据（如作者、期刊）转换为分类编码或独热编码。

2. 构建异构图

- **节点定义**: 将文档、图像、元数据作为不同类型的节点。
- **边定义**: 基于文档内容和元数据之间的关系定义边。

3. 异构图神经网络架构

- **多模态特征融合**: 使用异构图注意力层来学习不同类型节点间的关系。
- **层级结构**: 使用多层图神经网络捕获局部和全局结构信息。
- **聚合策略**: 为不同类型的节点设计不同的聚合函数，以便更有效地整合信息。

4. 分类器设计

- 将异构图神经网络的输出传递给一个或多个全连接层，进行最终的文档分类。

## 可行性分析

- **技术成熟度**: 异构图神经网络在处理多模态数据方面已显示出优越性，表明此方案的技术基础是可行的。
- **数据整合**: 通过将文本、图像和元数据作为不同类型的节点，可以有效地捕获和利用异构数据之间的复杂关系。
- **模型复杂度**: 虽然模型可能相对复杂，但现代计算资源足以处理这种类型的模型。

## 创新性分析

- **多模态异构图**: 将多模态数据（文本、图像、元数据）整合到一个**异构图**中是一种新颖的方法，可提供更全面的数据表示。
- **定制化的注意力机制**: 为不同类型的节点和边设计**定制化的注意力机制**，可以更有效地捕获数据间的复杂关系。
- **深层次的特征学习**: 通过使用**多层图网络**，可以学习数据的深层次特征，这对于提高分类性能可能非常有帮助。