研究与开发

# 基于GNN与注意力机制的文本分类模型

曾谁飞[1,2]，孟瑶[1,2,3,4]，刘静[3,4]

（1. 青岛海尔电冰箱有限公司，山东 青岛 266700；
2. 海尔优家智能科技（北京）有限公司，北京 100006；
3. 华东师范大学软件工程学院，上海 200062；
4. 上海市高可信计算重点实验室，上海 200062）

**摘　要：**针对图数据动态聚合未知邻节点学习能力难及融合语义特征不足造成的模型性能欠佳而分类准确率低的问题，提出了一种基于图神经网络（graph neural network，GNN）和注意力机制的分类模型——图注意力文本分类（graph attention text classification，GATC）。首先，构建了一种归纳式学习的图神经模型，利用聚合函数实现动态嵌入未知邻节点，增强模型泛化能力。其次，引入多头潜在注意力机制，通过低秩联合压缩技术减少推理键值缓存，显著地降低了内存占用，提高了模型性能。最后，融合GNN和门循环单元（gated recurrent unit，GRU）网络模型，进一步捕获图数据中结构和时序属性信息的语义特征，实现了特征的高效融合，并提升了模型的分类准确率。实验结果表明，所提方法既有效，又相比算法ADGL（adaptive dynamic graph learning）+MLA（multi-head latent attention）的分类准确率在CSI 100、CSI 300和Rus 1K数据集上分别提高至少4.0%、2.4%和3.1%。

**关键词：**图神经网络；注意力机制；图数据；文本分类

**中图分类号：**TP183

**文献标志码：**A

**doi:** 10.11959/j.issn.1000−0801.2025136

# Text classification model based on GNN and attention mechanism

ZENG Shuifei[1,2], MENG Yao[1,2,3,4], LIU Jing[3,4]

1. Qingdao Haier Refrigerator Co., Ltd., Qingdao 266700, China
2. Haier Uplus Intelligent Technology (Beijing) Co., Ltd., Beijing 100006, China
3. Software Engineering Institute, East China Normal University, Shanghai 200062, China
4. Shanghai Key Laboratory of Trustworthy Computing, Shanghai 200062, China

**Abstract:** Addressing the issue of low classification accuracy raised by the poor performance of the model, which is caused by the difficulty in learning from dynamic aggregation unknown neighboring nodes of graph data and insufficient fusion of semantic features, a model named graph attention text classification(GATC) based on graph neural net-

收稿日期：2025−03−30；修回日期：2025−04−24
通信作者：曾谁飞，zengshuifei@139.com

work (GNN) and attention mechanism was proposed. Firstly, an inductive learning of graph neural network model was constructed, and dynamic embedding the unknown neighboring node was implemented by using an aggregation function to enhance the model's generalization ability. Secondly, the reasoning cache size of key-value was reduced by the introduction of multi-head latent attention mechanism that utilized the low-rank key-value joint compression technology, which significantly diminished memory usage and improved the performance of the model. Finally, the integration of GNN and gated recurrent unit (GRU) network models further captured the semantic feature information of structural and temporal attributes for graph data, resulting in achieving efficient feature fusion and improving the classification accuracy of the model. The experimental results show that the proposed method not only is effective, but also improves the accuracy of classification that is increased at least 4.0%, 2.4% and 3.1% on the CSI 100，CSI 300 and Rus 1K datasets, respectively, compared with the algorithm ADGL+MLA (adaptive dynamic graph learning+ multi-head latent attention).

**Key words:** GNN, attention mechanism, graph data, text classification

## 0 引言

近来，大语言模型技术快速发展，应用逐渐落地，如DeepSeek[1-2]目前在文本生成和自然语言能力方面一枝独秀，这得益于计算效率和模型长序列处理能力，尤其是采用多头潜在注意力机制，大幅地提升了中文领域下游目标任务，如文本分类和知识问答的精准度。还有像股市时空数据、情绪、食材意图等垂类领域文本分类的准确率不仅与基础模型结构设计和性能有关，还与垂类模型优化方法密切关联，如时空数据如何提取序列特征向量，因此提高文本分类的准确率是自然语言处理（natural language processing，NLP）的关键任务。

目前较常见的提高文本分类准确率的方法有深度学习和图神经网络（graph neural network，GNN）方法，这两种方法在提高文本分类准确率方面各有优缺点，如GNN非常擅长图数据领域下游文本分类应用。文献[3]提出了一种适合图结构信息的GCN模型，该模型更擅长捕获局部特征信息。再如，有些学者提出了类似模型图变换器（graph transformer，GT）[4]代替GNN模型，并发现GT模型比GNN更具备任意节点对的交互分类任务能力，但是该方法需与获取全局特征信息的模型相结合才有可能更全面地表征文本。而后涌现出了众多引入注意力机制的研究成果[5-8]，研究者提出图注意力网络GAT[6]，具有更动态的归纳学习能力，还有文献[8]引入图采样与聚合（graph sample and aggregate，GraphSAGE）方法与自注意力机制实现每个节点动态地聚合邻节点特征信息，增强模型稳定性和表征能力。文献[9]提出了一种面向硬件对齐且可原生训练的稀疏注意力机制，提高模型性能的同时加速了推理效率，有效地降低了模型的训练成本。然而，在实际应用场景落地中，无论是深度学习方法还是GNN方法，还是这两者融合引入注意力机制的深度神经网络（deep neural network，DNN）模型，都更多关注构建模型结构本身要素，通常必须从模型性能和效率层面通盘考量，特别是算力不足的情况如何优化模型训练成本和计算效率，如以网络架构Transformer为主的大模型通过Scaling law方法提高模型性能效果，像OpenAI公司已发布的生成式预训练变换器GPT-X（generative pre-trained transformer）[10]和DeepSeek-R1-Zero[1-2]采用纯强化学习新的推理范式。

综合上述问题可知，针对图数据动态聚合未知邻节点学习能力难和融合语义特征不足造成的模型性能欠佳而分类准确率低的问题，本

文提出了一种基于GNN与注意力机制的文本分类模型——图注意力文本分类（graph attention text classification，GATC），该方法的主要贡献和创新点如下。

（1）提出了一种归纳式学习的图神经模型GNN，采用聚合函数采样和聚合（sample and aggregate，SAGE）来实现动态嵌入未知邻节点生成。实验结果表明，该方法较传统直推式学习方法能更有效地学习图邻居未知节点，增强了模型泛化能力。

（2）提出了一种低秩多头潜在注意力（multi-head latent attention，MLA）机制，通过低秩联合压缩技术降低了推理过程中的键值缓存大小，在此基础上设计并实现了一种融合GNN和门循环单元（gated recurrent unit，GRU）网络模型，进一步捕获图数据中结构信息与时序属性的语义特征，既大幅降低了内存占用和提高了模型性能，又实现了特征的高效融合，最终提升了模型分类效果。

（3）为了验证所提的方法，实验数据集源自中国和美国证券市场的3个市场基础指数，分别构建CSI 100（China securities index 100）和CSI 300（China securities index 300）数据集，获取Yahoo Finance上数据构建Rus 1K（Russell 1000）数据集，实验结果表明，本文方法的模型性能和准确率优于对比方法。

## 1　相关工作

本节对所提的文本分类模型进行介绍，主要包括基于GNN的文本分类模型和基于注意力机制的文本分类模型。

（1）基于GNN的文本分类模型

本文重点介绍了图数据动态嵌入生成方法，一方面图数据通常存在节点间关系且涌现出了众多GNN模型，像文献[11]提出DisenGCN（dynamic disentangled graph convolutional network）模型通过动态路由机制和多通道特征纠缠来进行节点分类，未深度分析不同邻节点的特征更新可能有不同的影响。而文献[12]关系图注意力网络（relational graph attention network，RGAT）模型引入图注意力机制实现图节点间的多关系特征理解，提高了分类精度，该方法虽然解决了传统模型无法捕获图节点间的多关系特征的问题，但是无法动态生成新的图节点嵌入，忽略了节点间的时间序列信息。文献[13]提出了一种关系股票排序任务的股票预测模型，通过时间敏感编码方式获得股票关系，但是采纳时序图卷积需时间维度上建模，导致计算复杂度要求高，无法在大规模图数据场景中应用。另一方面，从GNN开始就有许多研究者关注如何将DNN引入图论领域，这是具有一定挑战性的问题。文献[14]即使发现了股票之间存在条件相关性，但无法学习下游任务相应的网络结构信息，导致GNN难以利用DNN的自适应能力。文献[15]提出了一种层次结构的图注意力机制，既能捕捉股票个股之间的微观信息，又能获取市场层面的宏观信息，实际上，该方法并没有考虑如何获取更复杂的关系等全局特征信息。综上，基于GNN的文本分类模型方法聚焦于某一方面语义特征信息，难以充分并全面地表征文本语义信息及其实用性价值。文献[8]提出了一种归纳式学习的GraphSAGE图表示方法，在大规模图数据中能够高效地表征学习新图节点，无须重新训练整个模型网络，该方法能动态生成嵌入节点并体现节点间关系，泛化能力强，但是并未反映时序关系的语义信息。文献[16]提出了一种GNN-GRU融合网络模型提取更全面的文本特征，该模型不仅获取文本长距离的时序语义特征，而且同时处理全局文本信息和局部细节关键信息。因此，本文受文献[8, 16]的启发并运用于文本分类任务。

（2）基于注意力机制的文本分类模型

最近自然语言理解任务和大模型大部分是以

Transformer为主的DNN架构，其中多头注意力（multi-head attention，MHA）机制是最核心的组成模块。MHA是由Vaswani等[17]最早提出的深度学习注意力机制，并且在自然语言理解、图像处理等人工智能（artificial intelligence，AI）技术领域表现突出，尤其与循环神经网络（recurrent neural network，RNN）或卷积神经网络（convolutional neural network，CNN）相结合擅长捕获数据中长距离依赖关系或全局特征信息，提高了下游目标任务如文本分类、问答系统的预测准确率。接着出现各种各样的MHA变体网络模型，多查询注意力（multi-query attention，MQA）[18]、分组查询注意力（grouped-query attention，GQA）[19]、MLA[20]、原生稀疏注意力（native sparse attention，NSA）[9]等方法都是为了在减少键值（key-value，KV）缓存的同时提升模型效果，MLA显著地减少了键值缓存，同时提高了模型的效果，并优于MHA，而MQA和GQA提高模型的性能不如MHA。可见，虽然上述方法都是对KV缓存的优化机制，但是未深度考虑注意力机制的实用性，如内存消耗对目标任务响应时间带来体验价值。

综上所述，基于GNN的文本分类模型和基于注意力机制的文本分类模型尚未充分统筹文本分类任务中融合语义特征的同时提高模型性能。因此，本文提出了一种基于GNN与注意力机制的文本分类模型，从而实现文本分类模型的准确预测。

## 2 方法

基于GNN和注意力机制的文本分类模型架构如图1所示，由3个模块组成：特征聚合模块、特征融合模块和预测分类模块。下面将对各模块具体细节和本文损失方法进行详细介绍。

（1）特征聚合模块

生成文本数据中节点特征序列记为$\boldsymbol{X_t}=[X_t^1, X_t^2,\cdots,X_t^n]^{\mathrm{T}}$，$t$表示时间步，$n$表示图节点，图数据邻接矩阵**adj**（adjacency matrix）为图节点生成嵌入聚合，考虑数据动态时序属性及训练效率的因素，也就是既能动态添加相邻图节点生成嵌入，无须因新图节点的加入对模型重新训练，因此，本文借鉴文献[8]的方法，SAGE平均聚合计算如式（1）和式（2）所示。式（1）实现从第$k$-1层节点特征$\boldsymbol{h}_u^{k-1}$中得到所有邻居节点$u$的特征，接着对这些邻居节点特征逐项变量求平均得到一个特征向量$\boldsymbol{h}_{N(v)}^k$。式（1）中的$\boldsymbol{h}_{N(v)}^k$为节点$v$在第$k$层的邻节点特征聚合结果，Mean表示对邻节点特征逐项求平均操作，$N(v)$表示节点$v$的邻节点集合，$\boldsymbol{h}_u^{k-1}$表示邻节点$u$在第$k$-1层的特征

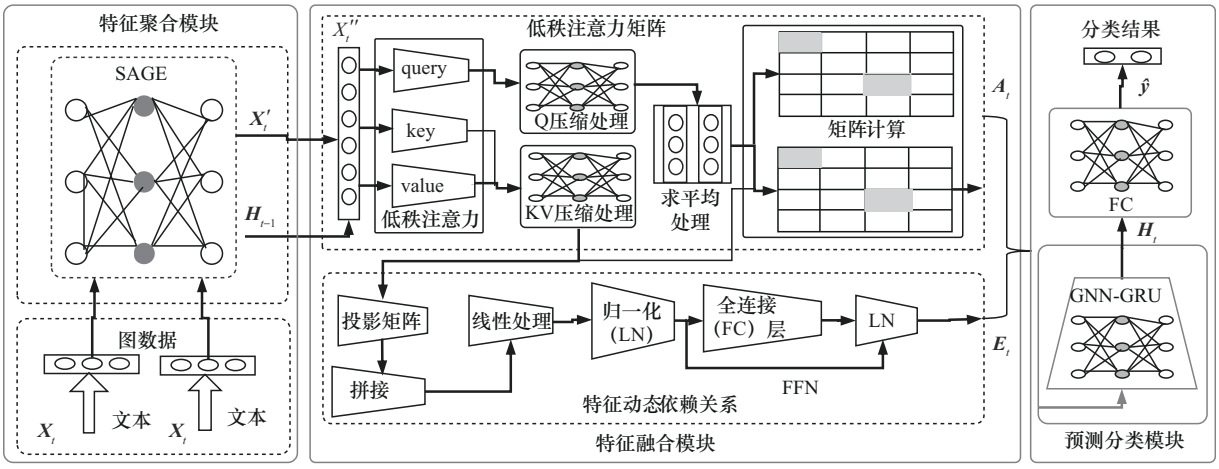


图1 基于GNN和注意力机制的文本分类模型架构

向量。

$$\boldsymbol{h}_{N(v)}^{k}=\mathrm{MEAN}(\{\boldsymbol{h}_{u}^{k-1},\forall u\in N(v)\}) \tag{1}$$

$$\boldsymbol{h}_{v}^{k}=\sigma(\boldsymbol{W}^{k}\cdot\mathrm{Concat}(\boldsymbol{h}_{v}^{k-1},\boldsymbol{h}_{N(v)}^{k})) \tag{2}$$

式（2）更新节点$v$在第$k$层特征向量的表示，式（2）实现两个向量拼接操作，即节点自身特征$\boldsymbol{h}_{v}^{k-1}$和近邻节点$\boldsymbol{h}_{N(v)}^{k}$合并计算，其中，激活函数$\sigma$转化为非线性表示使模型表示能力更强更丰富，$\boldsymbol{W}^{k}$为第$k$层学习权重矩阵，$\boldsymbol{h}_{v}^{k-1}$为节点$v$第$k$−1层的特征向量。当前时间步$t$经过特征聚合模块输出变量为$\boldsymbol{X}_{t}'$，将其与上一时间步$t$−1的隐藏状态$\boldsymbol{H}_{t-1}$进行拼接计算操作，记为$\boldsymbol{X}''_{t}=[\boldsymbol{X}_{t}';\boldsymbol{H}_{t-1}]$。

（2）特征融合模块

计算低秩注意力矩阵：为了获取文本数据更抽象的语义特征和降低内存占用与计算量，并受文献[20]的启示，本文采用低秩注意力机制来降低键值对的缓存大小，具体计算如式（3）~式（13）所示。

$$\boldsymbol{c}_{t}^{\mathrm{KV}}=\boldsymbol{W}^{\mathrm{DKV}}\boldsymbol{h}_{t} \tag{3}$$

$$[\boldsymbol{k}_{t,1}^{\mathrm{C}};\boldsymbol{k}_{t,2}^{\mathrm{C}};\cdots;\boldsymbol{k}_{t,n_h}^{\mathrm{C}}]=\boldsymbol{k}_{t}^{\mathrm{C}}=\boldsymbol{W}^{\mathrm{UK}}\boldsymbol{c}_{t}^{\mathrm{KV}} \tag{4}$$

$$[\boldsymbol{v}_{t,1}^{\mathrm{C}};\boldsymbol{v}_{t,2}^{\mathrm{C}};\cdots;\boldsymbol{v}_{t,n_h}^{\mathrm{C}}]=\boldsymbol{v}_{t}^{\mathrm{C}}=\boldsymbol{W}^{\mathrm{UV}}\boldsymbol{c}_{t}^{\mathrm{KV}} \tag{5}$$

式（3）~式（5）中的$\boldsymbol{h}_{t}$代表上一模块的输出变量$\boldsymbol{X}''_{t}$，$d_h$是每个注意力头的维度，$n_h$是注意力头的数量，$\boldsymbol{c}_{t}^{\mathrm{KV}}\in\mathbf{R}^{d_c}$是用于压缩KV的隐向量，$d_c$（$\leqslant d_h n_h$）表示KV压缩的维度，$\boldsymbol{W}^{\mathrm{DKV}}\in\mathbf{R}^{d_c\times d}$是下投影矩阵，$\boldsymbol{W}^{\mathrm{UK}}$，$\boldsymbol{W}^{\mathrm{UV}}\in\mathbf{R}^{d_h n_h\times d_c}$表示上投影矩阵，$\boldsymbol{k}_{t}^{\mathrm{C}}$和$\boldsymbol{v}_{t}^{\mathrm{C}}$分别为时间步$t$的键和值向量。

$$\boldsymbol{c}_{t}^{\mathrm{Q}}=\boldsymbol{W}^{\mathrm{DQ}}\boldsymbol{h}_{t} \tag{6}$$

$$[\boldsymbol{q}_{t,1}^{\mathrm{C}};\boldsymbol{q}_{t,2}^{\mathrm{C}};\cdots;\boldsymbol{q}_{t,n_h}^{\mathrm{C}}]=\boldsymbol{q}_{t}^{\mathrm{C}}=\boldsymbol{W}^{\mathrm{UQ}}\boldsymbol{c}_{t}^{\mathrm{Q}} \tag{7}$$

式（6）和式（7）中，设输入序列第$t$个令牌的嵌入向量$\boldsymbol{h}_{t}=\mathbf{R}^{d}$，$d$是嵌入维度，$\boldsymbol{c}_{t}^{\mathrm{Q}}\in\mathbf{R}^{d'_c}$为查询压缩后的隐向量，$d'_c$（$\ll d_h n_h$）表示查询压缩维度，$\boldsymbol{q}_{t}^{\mathrm{C}}$表示时间步$t$的查询向量，$\boldsymbol{W}^{\mathrm{DQ}}\in\mathbf{R}^{d'_c\times d}$，$\boldsymbol{W}^{\mathrm{UQ}}\in\mathbf{R}^{d_h n_h\times d'_c}$表示下投影矩阵和上投影矩阵。

$$[\boldsymbol{q}_{t,1}^{\mathrm{C}};\boldsymbol{q}_{t,2}^{\mathrm{C}};\cdots;\boldsymbol{q}_{t,n_h}^{\mathrm{C}}]=\boldsymbol{q}_{t}^{\mathrm{C}}=[\boldsymbol{q}_{t}^{\mathrm{R}},\boldsymbol{q}_{t}^{\mathrm{C}'}] \tag{8}$$

$$[\boldsymbol{k}_{t,1}^{\mathrm{C}};\boldsymbol{k}_{t,2}^{\mathrm{C}};\cdots;\boldsymbol{k}_{t,n_h}^{\mathrm{C}}]=\boldsymbol{k}_{t}^{\mathrm{C}}=[\boldsymbol{k}_{t}^{\mathrm{R}},\boldsymbol{k}_{t}^{\mathrm{C}'}] \tag{9}$$

式（8）和式（9）分别对查询向量和键向量进行解耦，其中，$\boldsymbol{q}_{t}^{\mathrm{R}}$表示解耦旋转位置编码（rotary position embedding，RoPE）查询向量部分，$\boldsymbol{q}_{t}^{\mathrm{C}'}$表示查询向量其他部分，$\boldsymbol{k}_{t}^{\mathrm{R}}$表示解耦RoPE键向量部分，$\boldsymbol{k}_{t}^{\mathrm{C}'}$表示键向量其他部分。

$$\boldsymbol{q}_{t}^{\mathrm{R}'},\boldsymbol{k}_{t}^{\mathrm{R}'}=\mathrm{RoPE}[\boldsymbol{q}_{t}^{\mathrm{R}},\boldsymbol{k}_{t}^{\mathrm{R}}] \tag{10}$$

$$[\boldsymbol{q}_{t,1};\boldsymbol{q}_{t,2};\cdots;\boldsymbol{q}_{t,n_h}]=\boldsymbol{q}_{t}=[\boldsymbol{q}_{t}^{\mathrm{C}'};\boldsymbol{q}_{t}^{\mathrm{R}'}] \tag{11}$$

$$[\boldsymbol{k}_{t,1};\boldsymbol{k}_{t,2};\cdots;\boldsymbol{k}_{t,n_h}]=\boldsymbol{k}_{t}=[\boldsymbol{k}_{t}^{\mathrm{C}'};\boldsymbol{k}_{t}^{\mathrm{R}'}] \tag{12}$$

计算相似度矩阵如式（13）所示。

$$\boldsymbol{R}^{h}=\frac{\boldsymbol{q}_{t,i}\boldsymbol{k}_{j,i}^{\mathrm{T}}}{\sqrt{d_q+d_h^{\mathrm{R}}}} \tag{13}$$

式（13）中$\boldsymbol{R}^{h}$表示在某位置和注意力头$h$下相似度矩阵，$\boldsymbol{q}_{t,i}$和$\boldsymbol{k}_{j,i}$表示时间步$t$的查询向量和键向量，$d_q$表示查询向量的维度，$d_h^{\mathrm{R}}$表示与注意力头$h$旋转位置编码的维度。

获取特征动态依赖关系：受文献[16-17, 21-24]启发，如式（14）~式（17）所示，对相似度矩阵$\boldsymbol{R}^{h}$进行Softmax计算得到$\boldsymbol{A}_{\mathrm{softmax}}$矩阵，接着对$\boldsymbol{A}_{\mathrm{softmax}}$求平均计算得到时间步$t$的平均注意力权重矩阵$\boldsymbol{A}_{t}$。$\boldsymbol{E}_{t}$表示当前时间步$t$的前馈神经网络（feedforward neural network，FNN/FFN）层输出特征向量。式（15）中$\boldsymbol{W}_{\mathrm{o}}$为用于将拼接后的多头注意力输出映射到最终空间的投影矩阵。式(16)中$\boldsymbol{W}_{t}$表示经过MLA注意力评分函数所得的注意力分值。式（17）中$\boldsymbol{O}$表示MLA层的最终输出向量，$\boldsymbol{W}_{n}$和$b_n$（$n$=1和2分别表示第1、2个全连接层）分别表示全连接层的权值矩阵和偏置值。

$$\boldsymbol{A}_{\mathrm{softmax}}=\mathrm{softmax}(\boldsymbol{R}^{h}) \tag{14}$$

$$\boldsymbol{W}_{t}=\boldsymbol{A}_{\mathrm{softmax}}\cdot\boldsymbol{v}_{t}^{\mathrm{C}} \tag{15}$$

$$\boldsymbol{O}=\boldsymbol{W}_{\mathrm{o}}\cdot\mathrm{Concat}(\boldsymbol{W}_{t}) \tag{16}$$

$$\boldsymbol{E}_{t}=\mathrm{FFN}(\boldsymbol{O})=\mathrm{ReLU}(\boldsymbol{O}\boldsymbol{W}_{1}+b_{1})\boldsymbol{W}_{2}+b_{2} \tag{17}$$

（3）预测分类模块

针对已积累的历史数据，通过GNN获取这些数据所变化的动态特征依赖关系后，增强文本数据时序特征的语义信息，提升模型预测分类精度。本文参考文献[16, 21]，具体计算如式下。

$$\boldsymbol{r}_t=\sigma(\mathrm{GNN}_1(\boldsymbol{A}_t, [\boldsymbol{H}_{t-1}; \boldsymbol{E}_t])) \tag{18}$$

$$\boldsymbol{z}_t=\sigma(\mathrm{GNN}_2(\boldsymbol{A}_t, [\boldsymbol{H}_{t-1}; \boldsymbol{E}_t])) \tag{19}$$

$$\tilde{\boldsymbol{H}}_t=\tanh(\mathrm{GNN}_3(\boldsymbol{A}_t, [\boldsymbol{r}_t \otimes \boldsymbol{H}_{t-1}; \boldsymbol{E}_t])) \tag{20}$$

$$\boldsymbol{H}_t=(1-\boldsymbol{z}_t)\otimes \boldsymbol{H}_{t-1}+\boldsymbol{z}_t \otimes \tilde{\boldsymbol{H}}_t \tag{21}$$

式（18）~式（21）中，$\boldsymbol{r}_t \in \mathbf{R}^{n\times d_m}$和$\boldsymbol{z}_t \in \mathbf{R}^{n\times d_m}$分别属于重置门（reset gate）和更新门（update gate），用于跟踪和更新控制信息。$\mathrm{GNN}_i(\cdot)$为（$i$=1,2,3）独立的单层GNN，处理输入数据。$\tilde{\boldsymbol{H}}_t$为经过双曲正切函数处理后的中间节点。

因图数据具有时序特征，预测时间步$\tau$后分类分布趋势情况，在GNN与门控制循环单元GNN-GRU之上叠加一个全连接层，则计算分类分布概率如式（22）所示，$\boldsymbol{H}_t$表示时间步$t$的隐藏状态，其中，$\boldsymbol{W}_y \in \mathbf{R}^{d_m\times 2}$，且$b_y \in \mathbf{R}^2$是可训练参数。

$$\hat{\boldsymbol{y}}(t+\tau)=\sigma(\boldsymbol{H}_t\boldsymbol{W}_y+b_y) \tag{22}$$

（4）损失方法

焦点损失（Focal loss）[24]为

$$\mathrm{FL}(p_t)=-\alpha_t(1-p_t)^{\gamma}\ln(p_t) \tag{23}$$

式（23）中，$p_t$为选择$\hat{\boldsymbol{y}}(t+\tau)$中正确分类预测概率的最大值，$\alpha_t$表示正负分类样本的平衡因子，$\gamma$用于调整难易分类样本损失比重的焦点参数，通常设置为2，使模型关注难分类样本，$(1-p_t)^{\gamma}$表示增加难分类样本权重的焦点因子，若样本分类较容易，则$(1-p_t)^{\gamma}$值很小且损失变小，$\ln(p_t)$表示样本预测与真实标签概率的交叉熵损失。

综上，现描述注意力机制采用MLA为例的GATC算法思路，第（1）~（5）行描述如何聚合文本特征；第（6）行描述如何计算注意力机制相似度矩阵和掩码矩阵；第（7）~（8）行描述如何获取特征动态依赖关系；第（9）~（10）行描述分类概率分布情况；第（11）~（12）行描述如何计算损失和更新模型权重参数。

GATC算法思路如下：

**输入** 过去或之前$T$个时间步文本数据特征向量$\boldsymbol{X}_t, \boldsymbol{X}_{t-1}, \cdots, \boldsymbol{X}_{t-T+1}$，图数据邻接矩阵**adj**。

**输出** 时间步$\tau$后预测分类分布趋势$\hat{\boldsymbol{y}}(t+\tau)$。

（1）针对每个训练轮次：

（2）初始化隐藏状态$\boldsymbol{H}_0$；

（3）构建GATC算法每一层RNN模型；

（4）对时间步$t$，使用式（1）和式（2）对$\boldsymbol{X}_t$特征聚合，得到输出变量；

（5）将输出的$\boldsymbol{X}_t'$与上一时间步$t$−1的隐藏状态$\boldsymbol{H}_{t-1}$拼接，得到$\boldsymbol{X}_t''$；

（6）描述本文采纳MLA机制的算法过程，使用式（3）~式（13）计算相似度矩阵$\boldsymbol{R}^h$，接着计算稀疏化的掩码矩阵；

（7）使用式（14）计算稀疏化注意力权重矩阵$\boldsymbol{A}_{\mathrm{softmax}}$，并对时间步$t$求平均得到的注意力权重矩阵$\boldsymbol{A}_t$；

（8）使用式（16）~式（17）分别计算注意力输出向量$\boldsymbol{O}$和FFN层输出向量$\boldsymbol{E}_t$；

（9）将$\boldsymbol{A}_t$和$\boldsymbol{E}_t$输入GNN-GRU模型，使用式（21）计算得到当前时间步$t$的隐藏状态$\boldsymbol{H}_t$；

（10）使用式（22）计算分类分布概括$\hat{\boldsymbol{y}}(t+\tau)$；

（11）使用式（23）计算焦点损失；

（12）通过反向传播计算梯度并进行梯度裁剪，使模型训练稳定，接着更新权重参数。

## 3 实验验证与结果分析

为评估文中GATC方法的实验结果，在3个不同数据集上进行对比分析，验证所提方法的有效性与优越性。

### 3.1 实验设置与数据集

（1）实验设置

采用的对比基线模型为自适应动态图学习

（adaptive dynamic graph learning，ADGL）[21]，接着分别增加SAGE层、GQA和MLA这两种注意力机制和优化损失函数，最后针对文中GATC模型在3个不同数据集就有效性、优越性、损失函数、模型性能和计算资源等维度进行实验验证和结果分析。其中，准确率（accuracy，Acc）和精准率（precision，Prec）分别代表整体和正样本数中的预测准确程度情况。FLOPS（floating point operations per second）值表示一次前向传播过程中所执行的浮点运算总数，其单位为G，表示十亿次浮点运算。Params（parameters）表示模型训练参数总数，包含所有层中的权重和偏置项，单位为M（millions），表示百万个参数。GATC-MLA模型表示未增加MLA机制，GATC_GQA描述表示增加GQA机制，GATC_F表示GATC为基础的损失函数未优化模型。

（2）数据集

选择图数据特性和关系复杂的文本作为验证文中方法的语料[21]，本文实验用到的数据集源自中国和美国证券市场的3个市场基础指数，调用 Tushare Pro API 分别构建CSI 100数据集和CSI 300数据集，获取 Yahoo Finance 上数据构建Rus 1K数据集，数据集数据组成情况见表1，数据集具体数据分布见表2。CSI 100选取沪深300指数样本股中规模最大的100只股票组成，反映沪深证券市场中最具市场影响力的一批大市值公司的整体状况。CSI 300选取上海证券交易所和深圳证券交易所中市值大、具备流动性强的300支股票作为样本，反映沪深A股市场的整体表现。Rus 1K由富时罗素公司（FTSE Russell）编制，包含美国股市中市值最大的1 000家公司，反映美国大盘股的代表性指数。其中，数据集关键处理的3个步骤环节为数据清洗、数据预处理和数据矩阵处理。数据清洗主要是删除交易日期不足71个月的股票与交易起始日期不是2013年12月1日的股票，并填充缺失值。数据预处理计算股票每个交易日的收益率且保存。数据矩阵处理根据已保存的收益率并借鉴文献[21]，计算标签矩阵如式（24）、相关性矩阵如式（25）和式（26）、邻接矩阵如式（27），设数据集中有$N$组数据，每组数据有12条/个记录，每条数据有$M$个特征，则特征矩阵为（$N$, 12, $M$），最后保存计算结果。

标签矩阵计算如下：

$$y^s(t+\tau)=\begin{cases}0, & p^s_{t+\tau}\leqslant p^s_t\\ 1, & p^s_{t+\tau}>p^s_t\end{cases} \tag{24}$$

相关性矩阵计算如下。

若为Rus 1K数据集，则：

$$f(x)=\begin{cases}1, & \text{corr}\geqslant 0.80\\ 0, & \text{corr}\leqslant -0.65\\ 0, & \text{其他}\end{cases} \tag{25}$$

若为CSI 100和CSI 300数据集，则：

$$f(x)=\begin{cases}1, & \text{corr}\geqslant 0.91\\ 0, & \text{corr}\leqslant -0.70\\ 0, & \text{其他}\end{cases} \tag{26}$$

邻接矩阵计算如下：

**表1　数据集数据组成情况**

| 数据量（单位:支） | CSI 100 | CSI 300 | Rus 1K | 备注 |
|---|---|---|---|---|
| 原始股票数 | 100 | 300 | 1 028 | 删除文件中数据少于71行或起始日期不是2013年12月1日文件 |
| 处理后股票数 | 78 | 221 | 811 | |

**表2　数据集具体数据分布**

| 数据集 | CSI 100和CSI 300 | Rus 1K |
|---|---|---|
| 训练集 | 2015年1月—2018年12月 | 2015年1月—2018年12月 |
| 验证集 | 2019年1月—2019年6月 | 2019年1月—2019年6月 |
| 测试集 | 2019年7月—2020年12月 | 2019年7月—2020年12月 |

$$f(x)=\begin{cases}1, & \mathbf{adj}[a,b]>0\\ 1, & \mathbf{adj}[a,b]<0\\ 0, & \text{其他}\end{cases} \quad (27)$$

（3）超参数设置

主要模型参数说明见表3，模型通用参数见表4，GQA和MLA机制模型参数见表5。

**表3 主要模型参数说明**

| ID | 参数类型 | 主要参数说明 |
|---|---|---|
| 1 | 模型通用参数 | top_k：稀疏化指标，相关性矩阵中取前$k$个最大的值为稀疏化<br>Head（H）：注意力头数<br>Learning rate（L）：学习率<br>Weight（W）：权重衰减 |
| 3 | GQA参数 | q_h：查询query的注意力头数<br>kv_h：键值KV的注意力头数<br>attention_dropout:为了防止过拟合现象需要丢弃部分权重，则在Transformer网络架构计算注意力权重后，变量attention_dropout表示所丢弃的比例 |
| 4 | MLA参数 | num_attention_heads：注意力头数<br>q_lora_rank：压缩query的秩<br>kv_lora_rank：压缩键值KV的秩<br>v_head_dim:在注意力机制中，每个注意力头的值向量维度大小 |
| 5 | 焦点损失参数 | 通过增大$\alpha$，可以增加正样本的损失贡献，反之则减少正样本的损失贡献，用来调整样本分布的均衡性 |

**表4 模型通用参数**

| 数据集 | CSI 100 | | | | | CSI 300 | | | | | Rus 1K | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 模型 | ADGL | ADGL+SAGE | ADGL+SAGE+GQA | GATC_F | GATC_GQA | ADGL | ADGL+SAGE | ADGL+SAGE+GQA | GATC_F | GATC_GQA | ADGL | ADGL+SAGE | ADGL+SAGE+GQA | GATC_F | GATC_GQA |
| top_k | 3 | 3 | 9 | 3 | 3 | 3 | 6 | 3 | 3 | 6 | 3 | 3 | 6 | 3 | 6 |
| H | 5 | 5 | — | 5 | — | 5 | 8 | — | 5 | — | 5 | 5 | — | 5 | — |
| L | 0.01 | 0.005 | 0.000 1 | 0.000 1 | 0.000 1 | 0.001 0 | 0.001 0 | 0.000 5 | 0.000 5 | 0.0050 | 0.0010 | 0.0001 | 0.000 5 | 0.000 1 | 0.001 0 |
| W | 0.005 | 0.000 5 | 0.000 0 | 0.000 5 | 0.000 5 | 0.05 0 | 0.005 0 | 0.000 5 | 0.000 5 | 0.0000 | 0.000 0 | 1.000 0 | 1.000 0 | 0.050 0 | 1.000 0 |
| q_h | — | — | 8 | — | 8 | — | — | 8 | — | 8 | — | — | 8 | — | 8 |
| kv_h | — | — | 4 | — | 4 | — | — | 4 | — | 4 | — | — | 4 | — | 4 |
| α | — | — | — | 0.3 | 0.3 | — | — | — | 0.3 | 0.3 | — | — | — | 0.6 | 0.6 |

**表5 GQA和MLA机制模型参数**

| 数据集 | CSI 100 | | | CSI 300 | | | Rus 1K | | |
|---|---|---|---|---|---|---|---|---|---|
| 模型 | ADGL+MLA | GATC_F | GATC | ADGL+MLA | GATC_F | GATC | ADGL+MLA | GATC_F | GATC |
| top_k | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| L | 0.005 0 | 0.000 1 | 0.001 0 | 0.001 0 | 0.000 1 | 0.000 1 | 0.005 0 | 0.000 1 | 0.001 0 |
| W | 0.000 0 | 1.000 0 | 0.000 5 | 0.050 0 | 0.500 0 | 0.005 0 | 0.000 5 | 0.000 5 | 0.000 5 |
| q_h | — | — | — | — | — | — | — | — | — |
| kv_h | — | — | — | — | — | — | — | — | — |
| attention_dropout | 0.1 | 0.1 | 0.1 | 0.0 | 0.1 | 0.1 | 0.0 | 0.1 | 0.1 |
| num_attention_heads | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 |
| q_lora_rank | 16 | 12 | 16 | 8 | 12 | 16 | 8 | 12 | 12 |
| kv_lora_rank | 16 | 12 | 30 | 12 | 12 | 16 | 12 | 16 | 12 |
| v_head_dim | 12 | 24 | 32 | 24 | 16 | 32 | 24 | 24 | 24 |
| α | — | — | 0.3 | — | — | 0.3 | — | — | 0.6 |

### 3.2　实验结果分析

#### 3.2.1　GATC 方法有效性与对比分析

为了验证所提方法的有效性和优越性，选择不同数据集与多种不同的模型进行方法对比，不同模型在 3 个不同数据集的分类效果对比见表6。首先，本文方法GATC（ID为3）相较于ADGL（ID为1）未增加SAGE聚合算法在CSI 100、CSI 300 和Rus 1K 数据集上准确率和精准率分别提升 4.4 个百分点和 1.5 个百分点、2.9 个百分点和 6.3 个百分点、3.3 个百分点和 1.9 个百分点。结果表明，本文方法 GATC（ID 为 3）提高了预测股票走势和价格的准确率和精准率并优于传统直推式学习方法，这是因为本文方法在 ADGL 模型结构增加了 SAGE 采样方法或 SAGE 层，模型性能与效果表现最佳。其次，该方法相较于 ADGL+SAGE（ID 为 2）增加 SAGE 聚合算法在 CSI 100、CSI 300 和 Rus 1K 数据集上准确率与精准率均提升 2.9 个百分点和 1.5 个百分点、1.2 个百分点和 4.6 个百分点、1.1 个百分点和 0.7 个百分点。结果进一步表明本文方法既能更准确地判断股票整体趋势如准确率，又能更出色地预测具体类别如精准率，这是因为 GATC 在 SAGE 采样基础上增加了 MLA 机制和 GNN-GRU 融合神经网络模型，并优化了损失函数整体，提升了语义特征融合能力，包括捕获股票数据中结构信息和时序属性的文本特征。与此同时，实验结果验证了该方法在不同数据来源、数据规模和数据特征上的泛化能力，特别是基于股票数据的不同市场预测分类任务自适应能力。这是由于直接对数据进行线性处理可能导致噪声或冗余信息影响动态节点嵌入生成和特征融合不足，还增加了计算量，最终影响模型性能。综合上述分析，SAGE 动态聚合邻节点方法不但可以更高效地生成邻节点特征信息，而且可以有效地提取文本数据的关键信息，结合 MLA 机制和 GNN-GRU 网络模型可以更有效地提取和融合语义特征，提高了模型性能和泛化能力。总之，在上述数据集的实验结果和不同模型结果的对比分析充分表明了本文方法的有效性和优越性。

表6　不同模型在 3 个不同数据集的分类效果对比

| ID | 模型 | CSI 100 | | CSI 300 | | Rus 1K | |
|---|---|---|---|---|---|---|---|
| | | Acc | Prec | Acc | Prec | Acc | Prec |
| 1 | ADGL[21] | 55.6% | 58.2% | 54.6% | 54.7% | 56.6% | 57.3% |
| 2 | ADGL+SAGE | 57.1% | 58.2% | 56.3% | 56.4% | 58.8% | 58.5% |
| 3 | GATC | **60.0%** | 59.7% | 57.5% | **61.0%** | **59.9%** | 59.2% |

#### 3.2.2　优化损失函数结果分析

为提升分类效果和模型性能，本文对数据集中不同正负样本进行深入分析并从中发现数据存在类别不平衡性的问题，导致模型关注多数类别样本而忽视少数类别样本，还弱化了模型泛化和预测精度的能力。针对该问题对标准交叉熵损失进行优化，采用 Focal loss 损失函数抑制数据类别不平衡性带来的影响。损失函数优化分类效果对比见表 7，从表 7 中实验结果可以发现，GATC-MLA（ID 为 3）模型相较于 ADGL+SAGE（ID 为 1）在 CSI 100 和 CSI 300 数据集精准率各提升了 0.8 个百分点和 0.6 个百分点，Rus 1K 数据集上准确率和精准率各提高 0.2 个百分点和 0.8 个百分点；GATC（ID 为 4）模型相较于 GATC_F（ID 为 2）在 CSI 100、CSI

表7　损失函数优化分类效果对比

| ID | 模型 | CSI 100 | | CSI 300 | | Rus 1K | |
|---|---|---|---|---|---|---|---|
| | | Acc | Prec | Acc | Prec | Acc | Prec |
| 1 | ADGL+SAGE | 57.1% | 58.2% | 56.3% | 56.4% | 58.8% | 58.5% |
| 2 | GATC_F | 58.4% | 58.3% | 55.4% | 62.2% | 57.6% | 58.9% |
| 3 | GATC-MLA | 56.4% | 59.0% | 55.4% | 57.0% | 59.0% | **59.3%** |
| 4 | GATC | **60.0%** | 59.7% | 57.5% | **61.0%** | **59.9%** | 59.2% |

300和Rus 1K数据集上准确率和精准率分别提高1.6个百分点和1.4个百分点、2.1个百分点和−1.2个百分点、2.3个百分点和0.3个百分点。上述实验结果表明，本文改进损失方法，通过Focal loss引入调节因子$(1-p_t)^{\gamma}$使模型有效地分配较易分类的样本，并控制较小的权重损失，还有效地解决了数据类型不平衡性的问题，提高了模型泛化能力，提高了文本分类准确率和精准率。

3.2.3 GATC模型性能与计算资源对比分析

基于GQA和MAL机制的GATC模型性能与分类效果对比见表8，不同模型在3个不同数据集的计算资源对比见表9。根据表8和表9可知，GATC模型性能、分类效果及计算量与参数都有不同程度的提升，下面对实验结果进行分析。

（1）GQA机制

一方面，图数据中可能蕴含复杂的非线性关系，如噪声干扰影响模型性能，如股票市场数据带有噪声敏感和股价稳定性数据的特点；另一方面，模型本身结构同样影响模型效率。受文献[19]启示，将MHA改为GQA机制进行实验，实验结果见表8和表9，GATC_GQA（ID为4）和ADGL+SAGE+GQA（ID为3）预测指标均提高，它们的具体分别相较于GATC-MLA（ID为7）和ADGL+SAGE（ID为2）在CSI 300、CSI 100和Rus 1K数据集上准确率和精准率各提升了2.1个百分点和1.9个百分点、−0.7个百分点和1.1个百分点；1.6个百分点和−0.2个百分点、0.4个百分点和

表8 基于GQA和MLA机制的GATC模型性能与分类效果对比

| ID | 模型 | CSI 100 | | CSI 300 | | Rus 1K | |
|---|---|---|---|---|---|---|---|
| | | Acc | Prec | Acc | Prec | Acc | Prec |
| 1 | ADGL | 55.6% | 58.2% | 54.6% | 54.7% | 56.6% | 57.3% |
| 2 | ADGL+SAGE | 57.1% | 58.2% | 56.3% | 56.4% | 58.8% | 58.5% |
| 3 | ADGL+SAGE+GQA | 57.5% | 58.6% | 57.0% | 57.5% | 58.7% | 58.8% |
| 4 | GATC_GQA | 58.0% | 58.8% | 57.5% | 58.9% | 59.3% | 59.1% |
| 5 | ADGL+MLA | 56.0% | 58.9% | 55.1% | 55.2% | 56.8% | 57.6% |
| 6 | GATC_F | 58.4% | 58.3% | 55.4% | 62.2% | 57.6% | 58.9% |
| 7 | GATC-MLA | 56.4% | 59.0% | 55.4% | 57.0% | 59.0% | **59.3%** |
| 8 | GATC | **60.0%** | 59.7% | 57.5% | **61.0%** | **59.9%** | 59.2% |

表9 不同模型在3个不同数据集的计算资源对比

| ID | 模型 | CSI 100 | | CSI 300 | | Rus 1K | |
|---|---|---|---|---|---|---|---|
| | | GFLOPS | 参数/百万 | GFLOPS | 参数/百万 | GFLOPS | 参数/百万 |
| | 基于GQA的GATC模型计算量与参数对比 | | | | | | |
| 1 | GATC-MLA | 0.123 881 472 | 0.065 922 | 0.350 997 504 | 0.065 922 | 0.053 668 736 | 0.065 922 |
| 2 | GATC_GQA | 0.095 127 552 | 0.050 434 | 0.269 528 064 | 0.050 434 | 0.041 211 776 | 0.050 434 |
| | | 23.2% | 23.5% | 23.2% | 23.5% | 23.2% | 23.5% |
| | 基于MLA的GATC模型计算量与参数对比 | | | | | | |
| 3 | ADGL | 0.002 640 768 | 0.033 666 | 0.007 482 176 | 0.033 666 | 0.027 405 312 | 0.033 602 |
| 4 | ADGL+MLA | 0.001 150 656 | 0.017 794 | 0.003 260 192 | 0.014 818 | 0.019 269 360 | 0.023 826 |
| 5 | ADGL+SAGE | 0.012 829 440 | 0.164 226 | 0.036 350 080 | 0.164 226 | 0.133 393 280 | 0.164 226 |
| 6 | GATC_F | 0.001 470 144 | 0.018 850 | 0.004 236 128 | 0.019 170 | 0.020 216 608 | 0.024 930 |
| 7 | GATC-MLA | 0.012 829 440 | 0.164 226 | 0.036 350 080 | 0.164 226 | 0.133 393 280 | 0.164 226 |
| 8 | GATC | 0.002 471 040 | 0.031 682 | 0.007 001 280 | 0.031 682 | 0.025 128 024 | 0.030 986 |

0.4个百分点；0.3个百分点和−0.2个百分点、−0.1个百分点和0.3个百分点。并且模型计算效率大幅提升，如表9中GATC-MLA（ID为1）和GATC_GQA（ID为2）所示，在3个数据集上GFLOPS和参数值各减少了23.2%和23.5%。分析实验结果与对比可以发现，采用GQA机制可以有效地提取数据关键信息，如图数据结构与时序属性及其特征有效融合，获取了稳定性的数据特征，增强了对股票价格波动趋势、市场趋势转变等预测能力，通过GQA机制降低了计算资源占有率，提高了模型性能，最终提升了模型分类效果。

（2）MLA机制

为验证GATC方法和模型效果，进一步提升分类效果和模型性能，考虑到图结构推理方法对长周期预测任务有较突出的表现，对计算资源要求较高，但是该方法难以满足短周期实时性要求高的预测任务。针对此问题采用MLA通过一个共享潜在空间，接着联合压缩KV值为低秩潜在向量，故不仅减少了推理过程中键值的缓存大小，而且提高了在预测分类任务精度的前提下满足实时性和降低计算显存资源的要求。首先，通过低秩联合压缩技术减少了冗余计算，使模型可以高效地捕获数据特征，提高了模型效果，实验结果见表8，ADGL+MLA（ID为5）相较于ADGL（ID为1）模型在CSI 100、CSI 300和Rus 1K数据集上的准确率和精准率分别提升了0.4个百分点和0.7个百分点、0.5个百分点和0.5个百分点、0.2个百分点和0.3个百分点。其次，通过低秩联合压缩与SAGE及在GATC_F（ID为5）采用Focal loss后，对比发现模型性能与效果再提高，GATC（ID为8）相较于ADGL+MLA（ID为5）在CSI 100、CSI 300和Rus 1K数据集上的准确率和精准率分别提高了4.0个百分点和0.8个百分点、2.4个百分点和5.8个百分点、3.1个百分点和1.6个百分点。最后，GATC计算量与参数见表9，MLA在3个不同数据集上表现出了优秀的模型性能。当选择GATC模型准确率和精准率最优值时，MLA显著地降低了计算复杂度和参数量。具体而言，MLA在CSI 100数据集上运算速度和参数（百万）分别由0.012 829 44 GFLOPS降至0.002 471 04 GFLOPS、0.164 226降至0.031 682，可知各降低了约80.7%和80.7%的计算复杂度和参数量。因此，MLA机制的表现不仅优于相比较的方法，而且降低了推理过程中模型计算资源要求，提高了推理性能和分类效果。

综合上述，本文方法在上述数据集进行了实验和不同模型结果对比分析既充分表明了基于MLA机制GATC方法的有效性和优越性，又充分论证了该方法较好地降低了计算资源，并在模型性能和分类效果方面优于其他方法。

## 4　结束语

本文提出了一种基于GNN与注意力机制的文本分类模型。为有效动态聚合图数据未知邻节点的学习能力，构建了一种归纳式学习的图神经模型，通过聚合函数实现动态嵌入未知邻节点；为高效融合语义特征而提高模型性能，采用多头潜在注意力机制，通过低秩联合压缩方法减少了键值缓存，不仅降低了内存消耗，而且增强了模型性能和泛化能力。最后融合GNN和GRU网络模型，进一步获取图数据结构信息与时序语义特征融合，实现了文本分类准确率的预期目标。实验结果表明，本文所提方法既有效又相比于对比算法ADGL+MLA的分类准确率在CSI 100、CSI 300和Rus 1K数据集上至少各提高了4.0个百分点、2.4个百分点和3.1个百分点，充分证明了本文方法的优越性。未来的研究可以尝试将数据中节点与边信息的关联性特征进而更高效的语义特征进行融合，同时探索更高效DNN模型，如通过Transformer架构，提升模型性能和推理效率，最终提高文本分类的准确率。

## 参考文献：


[1] DEEPSEEK-AI, LIU A X, FENG B, et al. DeepSeek-V3 technical report[EB]. arXiv preprint，2024, arXiv: 2412.19437.

[2] DEEPSEEK-AI, GUO D Y, YANG D J, et al. DeepSeek-R1: incentivizing reasoning capability in LLMs via reinforcement learning[EB]. arXiv preprint, 2025, arXiv: 2501.12948.

[3] YAO L, MAO C S, LUO Y. Graph convolutional networks for text classification[EB]. arXiv preprint, 2018, arXiv: 1809.05679.

[4] DENG C H, YUE Z C, ZHANG Z R. Polynormer: polynomial-expressive graph transformer in linear time[EB]. arXiv preprint, 2024, arXiv 2403.01232.

[5] LUO Y K, SHI L, WU X M. Classic GNNs are strong baselines: reassessing GNNs for node classification[EB]. arXiv preprint, 2024, arXiv: 2406.08993.

[6] VELIČKOVIĆ P, CUCURULL G, CASANOVA A, et al. Graph attention networks[EB]. arXiv preprint, 2017, arXiv: 1710.10903.

[7] JIN W, DERR T, WANG Y Q, et al. Node similarity preserving graph convolutional networks[C]//Proceedings of the 14th ACM International Conference on Web Search and Data Mining. New York: ACM Press, 2021: 148-156.

[8] HAMILTON W L, YING R, LESKOVEC J. Inductive representation learning on large graphs[C]//Proceedings of the 31st International Conference on Neural Information Processing Systems. Massachusetts: MIT Press, 2017: 1024-1034.

[9] YUAN J Y, GAO H Z, DAI D M, et al. Native sparse attention: hardware-aligned and natively trainable sparse attention[J]. arXiv preprint, 2025，arXiv:2502.11089.

[10] EL-KISHKY A, WEI A, SARAIVA A, et al. Competitive programming with large reasoning models[J]. arXiv preprint, arXiv:2502.06807, 2025.

[11] WU S W, XIONG Y T, LIANG H, et al. D2-GCN: a graph convolutional network with dynamic disentanglement for node classification[J]. Frontiers of Computer Science, 2025, 19: 191305.

[12] BUSBRIDGE D, SHERBURN D, CAVALLO P. Relational graph attention networks[J]. arXiv preprint, 2019, arXiv: 1904.05811.

[13] FENG F, HE X, WANG X, et al. Temporal relational ranking for stock prediction[J]. arXiv preprint, 2018, arXiv:1809.09441.

[14] CARDOSO J V D M, PALOMAR D P. Learning undirected graphs in financial markets[C]//Proceedings of the 2020 54th Asilomar Conference on Signals, Systems, and Computers. [S. l.:s.n.], 2020: 741-745.

[15] KIM R, HOSO C, JEONG M, et al. HATS: a hierarchical graph attention network for stock movement prediction[J]. arXiv preprint, 2019, arXiv:1908.07999.

[16] HAJIRAMEZANALI E, HASANZADEH A, DUFFIELD N, et al. Variational graph recurrent neural networks[C]//Proceedings of the 33rd International Conference on Neural Information Processing Systems. [S.l.:s.n.], 2019: 10701-10711.

[17] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need[C]//Proceedings of the 31st International Conference on Neural Information Processing Systems. [S. l.: s. n.], 2017: 6000-6010.

[18] SHAZEER N. Fast transformer decoding: one write-head is all you need[J]. arXiv preprint, 2019, arXiv:1911.02150.

[19] AINSLIE J, LEEOT J, JONG M D, et al. GQA: training generalized multi-query transformer models from multi-head checkpoints[J]. arXiv preprint, 2023, arXiv:2305.13245.

[20] DEEPSEEK-AI, LIU A X, FENG B, et al. DeepSeek-V2: a strong, economical, and efficient mixture-of-experts language model[J]. arXiv preprint, 2024, arXiv:2405.04434.

[21] TIAN H, ZHANG X, ZHENG X, et al. Learning dynamic dependencies with graph evolution recurrent unit for stock predictions[J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2023, 53(11): 6705-6717.

[22] HE K M, ZHANG X Y, REN S Q, et al. Deep residual learning for image recognition[C]//Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Piscatarvay: IEEE Press, 2016: 770-778.

[23] GEVA M, SCHUSTER R, BERANT J, et al. Transformer feed-forward layers are key-value memories[C]//Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. [S.l.:s.n.], 2021: 5484-5495.

[24] LIN T Y, GOYAL P, GIRSHICK R, et al. Focal loss for dense object detection[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2017(99): 2999-3007.


[作者简介]


**曾谁飞**（1978– ），男，博士，青岛海尔电冰箱有限公司、海尔优家智能科技（北京）有限公司工程师，主要研究方向为人工智能、大模型、深度学习、神经网络、机器学习、多模态等。

**孟瑶**（1999– ），女，华东师范大学软件工程学院硕士生，主要研究方向为深度学习、自然语言理解、软件建模、风险预警、可信人工智能。

**刘静**（1964– ），女，博士，华东师范大学软件工程学院教授，主要研究方向为软件建模、风险预警、可信人工智能。