Adapting GNNs for Document Understanding: A Flexible Framework With Multiview Global Graphs

Zhuojia Wu, Qi Zhang, Duoqian Miao, Xuerong Zhao, and Kaize Shi

Abstract—**Graph neural networks (GNNs) have recently gained attention for capturing complex relations, prompting researchers to explore their potential in document classification. Existing studies serving this purpose fall into two directions: inductive learning focusing on personalized context relations within documents and transductive learning targeting the global distribution relations among documents in a corpus. Both directions extract distinct types of beneficial structural information and yield encouraging outcomes. However, due to the incompatibility of underlying graph structures and learning settings, developing an enhanced model that effectively integrates local and global relational learning within existing frameworks is challenging. To address this issue, we propose a new GNN-based document representation learning framework that incorporates multiview global graphs at both the word and document levels, focusing on learning the diverse global distribution information of texts at different granularities. Additionally, a contextual encoder derives the initial representations of document nodes from the updated representations of word nodes, integrating personalized context relations into document representations during this process. Finally, we tailor a node representation learning strategy for the multiview global graphs, called the multiview graph sampling and updating module, which allows our framework to operate efficiently during training without being constrained by the scale of the global graph. Experiments indicate that our framework generally enhances performance by integrating both global and local relational learning. When combined with large-scale language models, our framework achieves state-of-the-art results for GNN-based models across multiple datasets.**

Received 5 December 2023; revised 4 June 2024; accepted 10 September 2024. Date of publication 21 October 2024; date of current version 3 April 2025. This work was supported in part by the National Key Research and Development Program of China “Key Special Project on Cyberspace Security Governance” under Grant 2022YFB3104700; in part by the National Natural Science Foundation of China under Grant 61976158, Grant 62376198, Grant 62006172, and Grant 62076182; and in part by the Shanghai Baiyulan Talent Plan Pujiang Project under Grant 23PJ1413800. *(Corresponding author: Duoqian Miao.)*

Zhuojia Wu, Qi Zhang, and Duoqian Miao are with Tongji University, Shanghai 200092, China (e-mail: wuzhuojia@tongji.edu.cn; zhangqi_cs@tongji.edu.cn; dqmiao@tongji.edu.cn).

Xuerong Zhao is with Shanghai Normal University, Shanghai 201418, China (e-mail: xrzhao@tongji.edu.cn).

Kaize Shi is with the Data Science and Machine Intelligence Laboratory, University of Technology Sydney, Sydney, NSW 2007, Australia (e-mail: Kaize.Shi@uts.edu.au).

Digital Object Identifier 10.1109/TCSS.2024.3468890

Index Terms—**Document classification, graph neural network (GNN), inductive learning, multiview global graph, representation learning, transductive learning.**

I. Introduction

Document classification plays a pivotal role in various practical applications on social media [1], such as sentiment analysis [2], [3] and news detection/filtering [4], [5], [6], automatically assigning documents to semantic labels. The core process of document classification lies in learning discriminative document representations that encapsulate key information. With the rise of deep learning, various neural network models, including convolutional neural networks (CNN) [7], [8], long short-term memory (LSTM) [9], [10], [11], and multilayer perceptrons (MLPs) [12], [13] have been widely applied to learn document representations. Recently, due to their advantages in handling long-distance dependencies and facilitating parallel computation, Transformers [14], [15] have been extensively utilized to construct large-scale language models (LLMs). These LLMs [16], [17], [18] undergo pretraining on extensive text datasets, thereby embedding a rich of general knowledge, making them the preferred choice seeking enhanced performance for NLP tasks. However, despite achieving impressive performance in document classification through the pretraining and fine-tuning approach, LLMs impose a significant computational burden due to their extensive parameter count and training costs.

Recently, various efforts have been directed at developing advanced networks that mine latent inter- and intradoc (document) relations to generate more informative document representations, aiming to enable models trained only on limited downstream task data to compete with LLMs. GNNs [19], [20] have attracted attention for their ability to capture complex relations and structural information. Existing GNN-based models for document representation learning are categorized into two distinct frameworks: 1) *learning intradoc local context relations*, such as syntax and semantics; and 2) *learning interdoc global distribution relations*, such as co-occurrence and clustering. Both frameworks have yielded promising results, as these types of relations have proven to be highly valuable [21], [22], [23], [24]. For instance, sentiment analysis is highly sensitive to nuances in context and expression [25], while news detection/filtering focuses on word co-occurrence and clustering [26]. Unfortunately, due to the incompatibility in underlying

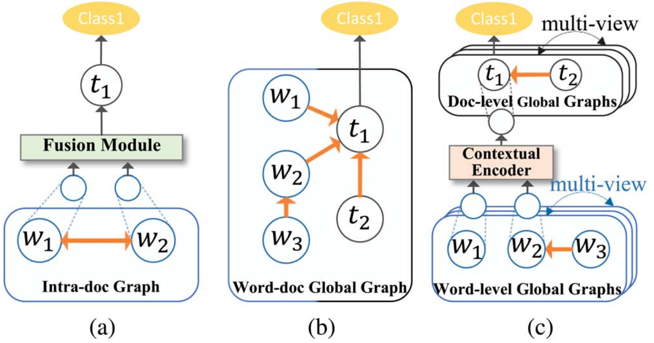


Fig. 1. Illustrations of three GNN-based frameworks for document representation learning. w and t represent the word and document nodes, respectively. Orange arrows signify information flow in the graph. Assuming t_1 is the document to be predicted, consisting of words w_1 and w_3, these include: (a) existing GNN-based inductive framework; (b) existing GNN-based transductive framework; and (c) our novel GNN-based framework. Specifically, in (a), word nodes can only get context information within the document; in (b), nodes only receive global information from the entire corpus; and in (c), our framework simultaneously captures global and context information by integrating multiview global graphs and a contextual encoder.

graph construction and learning settings, it is impractical to develop a more generalized model that simultaneously learns local context relations and global distribution relations under these existing frameworks.

Specifically, GNN-based models that focus on learning intradoc local context relations follow the principle that “Every Document Owns Its Structure” [22], with the general framework illustrated in Fig. 1(a). This approach involves constructing an independent intradoc graph for each document, wherein each node corresponds to a word within the document, and edges between nodes indicate long-range word dependencies, such as syntax [22]. Context information is propagated along the edges to update the representations of word nodes, which are then integrated through a Fusion Module to derive the representation of the document [22], [27]. The advantage of this framework guarantees that the well-trained model can effectively generalize to documents with unknown internal relations, i.e., inductive learning. However, the independent nature of intradoc graphs for each document poses challenges in transmitting corpus-level global information across documents.

Conversely, GNN-based models that focus on learning global distribution [20], [28], [29], [30], [31] relations typically construct a heterogeneous word-doc global graph, which includes nodes for all words and documents (including test documents) in the corpus, as illustrated in Fig. 1(b). Edges within this graph are established based on statistical global relations, such as word frequency and co-occurrence [20]. During training, word and document nodes are updated synchronously, integrating global information from neighboring nodes. The advantage of this framework is that it utilizes a diffusion mechanism to update the representations of test documents, thereby facilitating a clearer distinction of their classes, i.e., transductive learning. However, because the word-doc global graph merely represents simple inclusion relations between words and documents, the modeling of document internal context is overlooked [32].

To address the above limitations, we propose a novel GNN-based framework for document representation learning, as depicted in Fig. 1(c). This framework comprises two levels of homogeneous global graphs: one for words and another for documents, each encompassing all words from the vocabulary or all documents from the corpus, respectively. These two global graphs are connected through a contextual encoder, which allows a document representation to acquire distribution information from both the word- and doc-level global graphs while also learning the semantics and structural details within the document during contextual encoding. Moreover, to avoid information bias caused by a single type of global distribution relations [33], [34], we construct multiview global graphs that include multiple distribution relations of nodes under different metric perspectives, including co-occurrence, semantics, and clustering. Finally, we customize a node representation learning strategy for the proposed multiview global graphs, named multiview graph sampling and updating module. This module can independently update the representations of mini-batch specified nodes on the global graph, integrating information from neighboring nodes across different view global graphs through adaptive weighting. Thus, our framework can work well without being limited by the scale of the global graph.

Our contributions can be summarized as follows.

1) We propose a novel GNN-based framework for document representation learning, which, compared to previous approaches, can simultaneously learn local context relations and global distribution relations.
2) We construct multiview global graphs at both the word- and doc-levels, enhancing the model’s expressive capabilities while explicitly exploring the impact of different global distribution relations on document classification.
3) We design a multiview global sampling and updating module for aggregating global information on multiview global graphs in a memory-friendly and adaptive weighting manner. This module enables the global graphs to be integrated into a pipeline with any contextual encoder and trained via mini-batch gradient descent.
4) Our proposed framework outperforms existing GNN-based models and is competitive with LLMs. Moreover, when combined with LLMs, it achieves state-of-the-art performance across multiple datasets.

II. Related Work

This section summarizes existing studies on GNN-based models for document representation learning, including *inductive learning* (focusing on learning intradoc local context relations), *transductive learning* (focusing on learning interdoc global distribution relations), and *combination with LLMs*.

A. Inductive Learning

The earliest GNN-based inductive model for document representation learning was proposed by Huang et al., known as text-level GNN [21]. It was the first to suggest constructing an independent intradoc graph for each input document, allowing for improved scalability and personalized learning of intradoc context relations. Subsequent works have made further improvements based on this graph construction paradigm [21], [22], [23], [24]. One strategy of methods focuses on extracting high-order context information of each word through GNNs [22]. However, the over-smoothing problem of GCNs can often result in the reduction of these context relations to lower orders [35]. To address this, Ding et al. [23] enhanced the model's representational power by capturing the heterogeneous high-order context information of words, via hyperedges that connect an arbitrary number of nodes. Another strategy is to expand the receptive field of each word, enabling each layer of GNN to capture long-range interactions between words [35]. Furthermore, Wang et al. [36] decomposed context relations into word co-occurrence, syntactic dependency, fusion, and self-looping and utilized featurewise linear modulation to learn the edge types in the neighborhood features of words. Additionally, the quality of graphs significantly impacts representation learning. Dai et al. [27] enhanced graph quality by incorporating external knowledge into structural adjustments and fusing diverse external knowledge sources through a graph fusion process. Nevertheless, this approach does not incorporate sentence-level dynamic word information. Piao et al. [24] suggested creating multiple sentence-level graphs within a document, where word nodes convey local syntactic messages to intrasentence neighbors and global semantic messages to cross-sentence neighbors.

B. Transductive Learning

Earlier GNN-based transductive models can be traced back to Yao et al. [20] proposal, called TextGCN, which utilized spectral GCNs to learn representations of both words and documents simultaneously. Most subsequent methods have followed this model while incorporating improvements in other areas [20], [28], [29], [30], [31], [37]. One approach is to introduce more semantic information by enriching the types of nodes [37], [38]. For instance, integrating topics as nodes in the graph enables capturing the cluster relations among words and documents [37]. Another approach is to introduce more constraint information by enriching the types of edges [29], [30], [31]. For example, to tackle the problem of context information being absent in GNNs-based models, Liu et al. [29] developed a text graph tensor, which utilizes edges between word nodes to represent semantic, syntactic, and sequential context relations. Moreover, Ragesh et al. [30] decompose the single words and documents co-occurrence graph into a combination of multiple isomorphic and heterogeneous graphs, which allows for a more detailed exploration of the potential inter/intrarelations within words and documents. To enhance the robustness of node representation, Yang et al. [31] proposed a combination of contrastive learning and adaptive augmentation to reduce noise in the text graph while preserving structural integrity through joint optimization of contrastive loss and cross-entropy loss.

C. Combination With LLMs

LLMs, such as BERT [16], possess powerful capabilities for knowledge representation. Recent studies have explored the integration of GNN-based models with LLMs to enhance document classification performance [36], [39], [40], [41], [42], [43]. It is noteworthy that GNN-based models exhibit significant differences in training approaches across various learning settings. In inductive learning, since each document corresponds to an individual intradoc graph, this allows the GNN modules to process batch graphs and update parameters via mini-batch gradient descent. This setup naturally supports joint training with LLMs in the form of multitask learning [36], [42], [43]. Moreover, Wang et al. [36] enhanced the connectivity of the graph by introducing additional edge types into the construction of intradoc graphs; and Lin et al. [43] developed a heterogeneous directed graph attention network, utilizing multilevel semantic embeddings for inferential reasoning. Both approaches attempt to jointly train with LLMs through multitask learning, significantly improving document classification performance. In transductive learning, existing methods typically construct a word-doc global graph and update all node representations in a single iteration [39], [40], [41]. This approach presents challenges when combined with LLMs due to memory constraints. As a result, existing methods attempt to utilize LLMs solely for generating document embeddings without jointly fine-tuning LLMs's parameters. Furthermore, Zhao and Song [40] employed graph data augmentation and contrastive learning to maximize the representations of identical document nodes across different graph views, thereby enhancing the expressive power of the graph decoder. We develop a memory-efficient module for node representation learning tailored for global graphs. This enables our framework to integrate LLMs for concurrent parameter learning, unrestricted by the learning setting and the scale of the global graph.

III. Proposed Framework

A. Preliminaries

1) Problem Definition: For a given corpus $\mathcal{T}$, the vocabulary $\mathcal{V}$ contains all words that occur within it. Each document $t \in \mathcal{T}$ is a sequence of words denoted as $t = \{w_1, w_2, \ldots, w_l\}$, where each word w_i belongs to $\mathcal{V}$ and the length of document t is denoted by l. For simplicity, we focus on the single-label classification problem, and each document t is assigned a ground-truth label $y_t \in \{0,1\}^{|\mathcal{C}|}$, where $|\mathcal{C}|$ is the number of classes. Furthermore, $E_w \in \mathbb{R}^{d_0^w \times |\mathcal{V}|}$ represents the pretrained (e.g., *GloVe* [44]) word embedding matrix, where d_0^w denotes the dimension of embeddings and $|\mathcal{V}|$ is the size of vocabulary, where each column $e_w \in E_w$ corresponds to the embedding of a word from the vocabulary $\mathcal{V}$.

2) Global Graph Definition: We construct two independent homogeneous global graphs at two different levels of text granularity, i.e., word and document. The word-level global graph is represented as $G_w = (V_w,\ A_w)$, where the set of nodes V_w comprises $|\mathcal{V}|$ nodes, and each node corresponds to a word in the vocabulary $\mathcal{V}$. The adjacency matrix $A_w \in \{0,1\}^{|\mathcal{V}|\times|\mathcal{V}|}$ is initialized without considering the weights of the edges but instead dynamically learns them during the training process. Similarly, the doc-level global graph derived from $\mathcal{T}$ is formalized as $G_t = (V_t,\ A_t)$, where the set V_t comprises $|\mathcal{T}|$ nodes and each node corresponds to a document in the corpus. The corresponding adjacency matrix is $A_t \in \{0,1\}^{|\mathcal{T}|\times|\mathcal{T}|}$.

3) Multiview Global Graph Definition: Multiview global graphs with $\mathcal{M}$ views at the word- and doc-level are represented as sets of graph elements $\mathcal{G}_w = \{G_w^1, G_w^2, \ldots, G_w^{\mathcal{M}}\}$ and $\mathcal{G}_t = \{G_t^1, G_t^2, \ldots, G_t^{\mathcal{M}}\}$, respectively. It should be noted that the set of nodes is shared across different views, thereby simplifying the word- and doc-level multiview global graphs to $\mathcal{G}_w = (V_w, \{A_w^1, A_w^2, \ldots, A_w^{\mathcal{M}}\})$ and $\mathcal{G}_t = (V_t, \{A_t^1, A_t^2, \ldots, A_t^{\mathcal{M}}\})$, where $A^i \neq A^j$ when $i \neq j$. Our essential idea is to update node representations by integrating global information from multiple underlying graphs, thereby creating a more comprehensive information space that can facilitate subsequent tasks.

B. Overall Structure

The overall architecture of our proposed novel framework is illustrated in Fig. 2. It is noteworthy that our framework can be adapted to various learning settings through simple architectural adjustments. In detail, two-level multiview global graphs (TL-MGGs) model for transductive document representation learning can be obtained by constructing multiview global graphs at both the word and document levels and connecting them through a contextual encoder. Furthermore, an inductive learning model, namely one-level multiview global graph (OL-MGG), is implemented, which solely includes the word-level multiview global graph. Specifically, at each level, the multiview global graph (Section IV) consists of two submodules: a multiview graph sampling and updating module (Section IV-B) and a multiview self-attention integration module (Section IV-C). The former module concurrently learns global information from global graphs in different views, which reflects potential diverse relations among words or documents (Section IV-A). The latter module then integrates the obtained global information from different views through self-learned weights, acquiring a unique representation for each word or document.

In multiview graph sampling and updating module, the nodes targeted for learning in the current batch, along with their neighboring nodes, are initially detached from multiple global graphs. This detachment operation prevents the invocation of all nodes in the global graph during each mini-batch training iteration, thereby enhancing computational efficiency. Additionally, to facilitate the use of larger batch sizes in the transductive learning, we establish a memory to store the document embedding matrix $E_t \in \mathbb{R}^{d_0^t \times |\mathcal{T}|}$, where d_0^t is the dimension of the document embedding, and $|\mathcal{T}|$ is the number of documents in the corpus. The matrix E_t is utilized to preserve the embeddings of neighboring nodes for documents in the current learning batch, thus avoiding the simultaneous learning of these neighboring node representations. To expedite model convergence, we pretrain the contextual encoder on the training set and employ its output document representations to initialize the document embedding matrix E_t.

This innovative framework offers a flexible solution for document representation learning in both transductive and inductive learning. It simultaneously integrates diverse global structural information and local contextual details. Notably, this framework excels in memory efficiency and can be combined with LLMs for joint training.

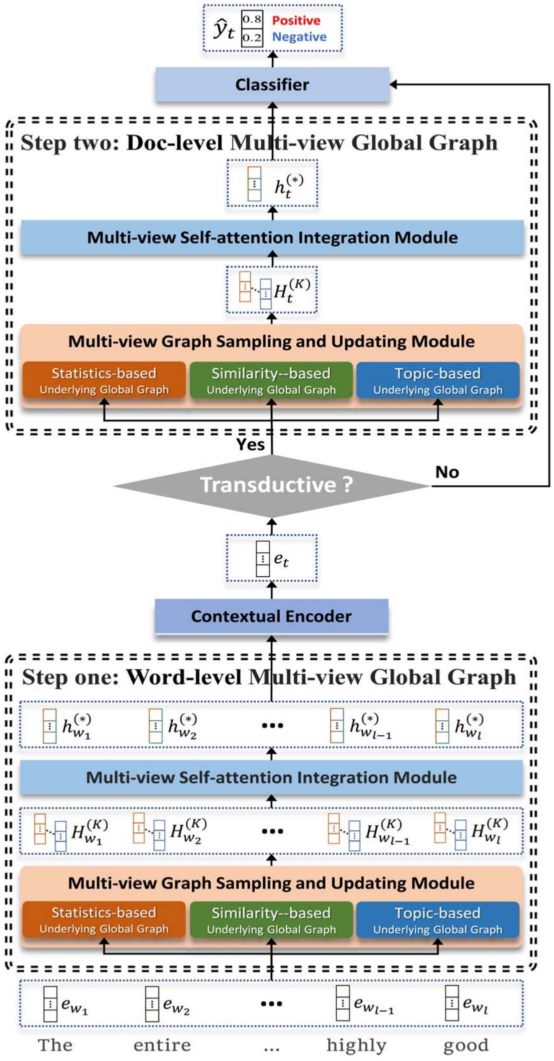

$\hat{y}_t$
0.8
0.2
Positive
Negative
Classifier
Step two: Doc-level Multi-view Global Graph
$h_t^{(*)}$
Multi-view Self-attention Integration Module
$H_t^{(K)}$
Multi-view Graph Sampling and Updating Module
Statistics-based Underlying Global Graph
Similarity--based Underlying Global Graph
Topic-based Underlying Global Graph
Yes
Transductive ?
No
e_t
Contextual Encoder
Step one: Word-level Multi-view Global Graph
$h_{w_1}^{(*)}$
$h_{w_2}^{(*)}$
•••
$h_{w_{l-1}}^{(*)}$
$h_{w_l}^{(*)}$
Multi-view Self-attention Integration Module
$H_{w_1}^{(K)}$
$H_{w_2}^{(K)}$
•••
$H_{w_{l-1}}^{(K)}$
$H_{w_l}^{(K)}$
Multi-view Graph Sampling and Updating Module
Statistics-based Underlying Global Graph
Similarity--based Underlying Global Graph
Topic-based Underlying Global Graph
e_{w_1}
e_{w_2}
•••
$e_{w_{l-1}}$
e_{w_l}
The
entire
...
highly
good

Fig. 2. Overall architecture of our novel GNN-based document representation learning framework with multiview global graphs. It is noteworthy that during inductive learning, the document representation e_t output by the contextual encoder is fed directly to the classifier for prediction. In transductive learning, e_t is further updated on the doc-level multiview global graph, integrating multiview global information between documents.

IV. Multiview Global Graph

A. Underlying Global Graphs Construction

Multiview global graphs are constructed, where distinct underlying global graphs contain diverse structural information. Specifically, we explore the potential co-occurrence, semantics, and cluster relations among words or documents through three views: *statistics-*, *similarity-*, and *topic-based views*. Subsequently, these relations are transformed into underlying global graph structural information by constructing nearest neighbor graphs [45].

The following are the methods for measuring the relations among words or documents from different views:

1) Statistics-Based View: For the word-level graph, the co-occurrence of two words in a corpus reflects their contextual relevance. The pointwise mutual information (PMI) in statistics can quantify this relation by comparing the probability of their joint distribution to the product of their marginal distributions. Therefore, we measure the relation between words w_i and w_j as follows:

$$f_w^{\text{stat}}(w_i, w_j) = \text{PMI}(w_i, w_j). \quad (1)$$

The Jaccard similarity coefficient (JSC) is a widely used statistical measure for quantifying the similarity between two sets [46], [47]. In this work, the document is naturally regarded as an unordered collection of words, where the co-occurrence of words represents the correlation between two sets. Specifically, in the doc-level graph, the relation between documents t_p and t_q is calculated as follows:

$$f_t^{\text{stat}}(t_p, t_q) = \sum_{w_i, w_j \in t_p \cap t_q} \text{PMI}(w_i,\ w_j) \quad (2)$$

where w_i and w_j are a pair of words that co-occur in both documents d_p and d_q. Formally, the PMI between two words is calculated using the sliding window strategy

$$\text{PMI}(w_i, w_j) = \log \frac{p(w_i,\ w_j)}{p(w_i)p(w_j)} \quad (3)$$

where $p(w_i,\ w_j) = \#W(w_i,\ w_j)/\#W$ and $p(w_i) = \#W(w_i)/\#W$. $\#W(w_i,\ w_j)$ is the number of times the word pair $(w_i,\ w_j)$ co-occurs within a sliding window in the entire corpus, $\#W(w_i)$ is the number of times the word w_i appears within a sliding window in the corpus, and $\#W$ is the total number of the sliding windows.

2) Similarity-Based View: The cosine similarity metric is widely used to measure the similarity between two vectors by calculating the cosine of their angle [48]. For word nodes, pretrained word embeddings contain abundant semantic information [44], and the similarity between embedding vectors further reflects the similarity of two words in the semantic space. Therefore, in the word-level graph, the relation function under this view can be defined as follows:

$$f_w^{\text{simi}}(w_i,\ w_j) = \text{Cosine}(e_{w_i},\ e_{w_j}) \quad (4)$$

where e_{w_i} and e_{w_j} are the pretrained word embeddings derived from the embedding matrix $E^w \in \mathbb{R}^{d_0^w \times |\mathcal{V}|}$.

In particular, LSTM has demonstrated remarkable capabilities in capturing the semantics of documents [49]. Consequently, we utilize the hidden-layer output vectors from the last time step of a pretrained LSTM as the document embeddings. These embeddings can subsequently be utilized to compute document similarity

$$f_t^{\text{simi}}(t_p,\ t_q) = \text{Cosine}(e_{t_p},\ e_{t_q}) \quad (5)$$

where e_{t_p} and e_{t_q} are pretrained low-dimensional document embeddings, and they are stored in the document embedding matrix $E_t \in \mathbb{R}^{d_0^t \times |\mathcal{T}|}$. Formally, the cosine is calculated by

$$\text{Cosine}(e_1,\ e_2) = \frac{e_1 \cdot e_2}{\|e_1\| \cdot \|e_2\|} \quad (6)$$

where e_1 and e_2 denote embedding vectors. The larger the computed result, the closer they are in the semantic space.

3) Topic-Based View: Unlike previous studies [38], [50], we also transform the cluster relations into the underlying graph structure. We first mine the latent clusters in the corpus by exploiting the latent Dirichlet allocation (LDA) topic model [51]. The LDA training iteration produces T clusters. For each word in the vocabulary $\mathcal{V}$ or each document in the corpus $\mathcal{T}$, it is assigned a topic probability distribution $\delta \in [0, 1]^T$. The unique cluster for the word or document is as follows:

$$c = \arg \max_{i \in 1,2,\ldots,T} \delta_i \quad (7)$$

where c denotes the unique cluster to which the word or document belongs and is identified as the element's index with the highest value in the probability distribution δ.

The nearest neighbor graph [45] is a directed graph that establishes asymmetric relations between nodes, with each node only being connected to its closest neighboring nodes, without considering the influence of other nodes. Moreover, we control the number of neighboring nodes for nodes in the word- and doc-level global graph by adjusting hyperparameters, denoted as K_w and K_t, respectively. These hyperparameters allow for adjusting the information received by each node and reducing the influence of noise information.

In both statistics- and similarity-based views, the nearest neighbor graph takes the form of a "0–1" KNN graph, and it is defined by the adjacency matrix A as follows:

$$A_{ij} = \begin{cases} 1, & v_i \in \text{KNN}(v_j) \\ 0, & \text{otherwise} \end{cases} \quad (8)$$

where $\text{KNN}(v_j)$ refers to the nodes that are most relevant to the word or document node v_j, which is determined using different relation functions [i.e., (1), (2), (4), and (5)]. The quantity of these nodes is dynamically adjusted through hyperparameters K_w and K_t.

In the topic-based view, each word or document node is connected to the top K (K_w and K_t) nodes with the highest membership degree in its corresponding cluster. The "0–1" KNN graph is defined as follows:

$$A_{ij} = \begin{cases} 1, & v_i \in \text{TopK}(c_{v_j}) \\ 0, & \text{otherwise} \end{cases} \quad (9)$$

where $\text{TopK}(c_{v_j})$ represents the top K_w words or K_t documents with the highest membership degree in cluster c_{v_j}. These words or documents are key elements that reflect the main topic of the cluster, and connecting with them maximizes information about the topics.

B. Multiview Graph Sampling and Updating Module

This module consists of two operations: 1) *multiview global graph sampling* is utilized to separate the nodes required for computation in the current batch from multiview global graphs; and 2) *global graph self-attention updating* is employed to aggregate information from neighboring nodes and update target node representations in a weights self-learning manner. The complete process is illustrated in Fig. 3.

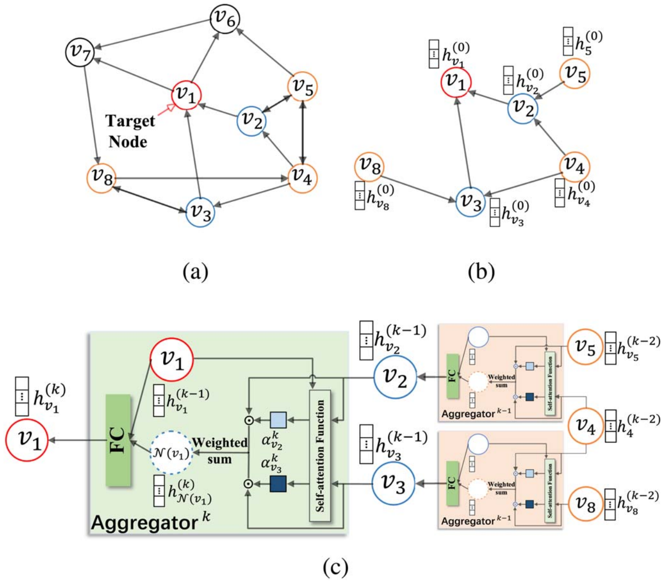


Fig. 3. Illustrations of two operations within the multiview graph sampling and updating module. Assuming a multiview global graph consists of a set of nodes $\{v_1, v_2, \ldots, v_7\}$, where v_1 is the target node to be learned in the current batch. Specifically, these include: (a) underlying global graph structure from a certain view; (b) subgraph formed by v_1 and its neighboring nodes within Kth order ($K = 2$) after sampling the global graph in (a), where v_2 and v_3 are the first-order neighboring nodes of v_1, marked in blue. v_4, v_5, and v_8 is the second-order neighboring nodes; (c). updating the representation of v_1 through K self-attention aggregators, where FC represents the fully connected layer.

1) Multiview Global Graph Sampling: The multiview global graph $\mathcal{G} = (V, \{A^1, A^2, \ldots, A^{\mathcal{M}}\})$ includes all word or document nodes, as well as multiple underlying graph structures built upon these nodes. Assume one of the underlying graph structures is illustrated in Fig. 3(a). We detach the target nodes $V_{\mathcal{B}}$ to be updated in the current batch, as well as their neighboring nodes within Kth order in different views from the corresponding underlying global graphs. Assuming one of the views, the nodes are to be detached as shown in Fig. 3(b). It is worth emphasizing that sampling node on the graph is an inverse expansion process compared to updating node representations. Specifically, in the doc-level multiview graph, the target nodes refer to the documents to be predicted in the current batch, denoted as $V_{\mathcal{B}_t} = \{t_1, t_2, \ldots, t_{|\mathcal{B}_t|}\}$, where $|\mathcal{B}_t|$ is the mini-batch size. In the word-level multiview graph, the target nodes are words contained in these documents, represented as $V_{\mathcal{B}_w} = \{w | w \in t_1 \cup t_2 \cup \ldots \cup t_{|\mathcal{B}_t|}\}$. For each view $m \in \{1, \mathcal{M}\}$, Algorithm 1 outlines this sampling process. at the initial Kth layer, the nodes in $V_{\mathcal{B}_m}^K$ are the target nodes. We then employ a sampling strategy to obtain the (K-1)th layer nodes, denoted as $V_{\mathcal{B}_m}^{K-1}$. Specifically, $V_{\mathcal{B}_m}^{K-1}$ encompasses the previous layer nodes $V_{\mathcal{B}_m}^K$ and their *first*-order neighboring nodes. This iterative process continues until the lowest layer, i.e., zeroth layer. Consequently, for each view m, a set $\{V_{\mathcal{B}_m}^0$, $V_{\mathcal{B}_m}^1, \ldots, V_{\mathcal{B}_m}^K\}$ is obtained, where $V_{\mathcal{B}_m}^{k-1}$ is comprised of all requisite nodes for updating the nodes in $V_{\mathcal{B}_m}^k$ during the forward propagation.

Algorithm 1 Reverse Sampling Algorithm for Target Nodes.

```
Input: Multi-view graph $\mathcal{G} = (V, \{A^1, A^2, ..., A^{\mathcal{M}}\})$;
       the view $m \in \{1, \mathcal{M}\}$;
       the set of target nodes $V_{\mathcal{B}}$;
       sample depth $K$
$V_{\mathcal{B}_m}^K \longleftarrow V_{\mathcal{B}}$
for $k = K$ to 1 do
  $V_{\mathcal{B}_m}^{k-1} \longleftarrow V_{\mathcal{B}_m}^k$
  for $v_m \in V_{\mathcal{B}_m}^k$ do
    $V_{\mathcal{B}_m}^{k-1} \longleftarrow V_{\mathcal{B}_m}^{k-1} \cup \mathcal{N}(v_m)$
  end for
end for
Output: { $V_{\mathcal{B}_m}^0$, $V_{\mathcal{B}_m}^1$, ..., $V_{\mathcal{B}_m}^K$ }
```

2) Global Graph Self-Attention Updating: We train a group of K individual self-attention aggregators [52] for each view, which aggregate neighboring information up to the Kth order from different views for target nodes $V_{\mathcal{B}}$. In particular, as illustrated in Fig. 3(c), the self-attention aggregator at the kth layer in each view can be formalized as follows:

$$h_{\mathcal{N}(v)}^{(k)} = f_{\text{aggregator}}^{(k)}(\theta_{ag}, \{h_u^{(k-1)}, \forall u \in \mathcal{N}(v)\}) \qquad (10)$$

where θ_{ag} is the set of trainable parameters. $v \in V_{\mathcal{B}}^k$ and the view superscript m is omitted for brevity. $\mathcal{N}(v) \in V_{\mathcal{B}}^{k-1}$ is the neighboring node set of node v, $h_u^{(k-1)}$ denotes the node u representation at the (k-1)th layer. $h_{\mathcal{N}(v)}^{(k)}$ is obtained by computing a weighted sum of representations from all neighboring nodes as follows:

$$h_{\mathcal{N}(v)}^{(k)} = \sum_{u \in \mathcal{N}(v)} \alpha_u^k \cdot h_u^{(k-1)} \qquad (11)$$

where α_u^k is the self-attention coefficient score of node u, which is computed based on the representations of nodes v and u, the self-attention function is outlined as follows:

$$\alpha_u^k = \psi\left(\mathbf{a}^{(k-1)^{\mathrm{T}}}\left[\, \mathbf{W}_\alpha^{(k-1)} h_v^{(k-1)} \mid\mid \mathbf{W}_\alpha^{(k-1)} h_u^{(k-1)} \right]\right) \qquad (12)$$

where $\mathbf{W}_\alpha^{(k-1)} \in \mathbb{R}^{d'_{k-1} \times d_{k-1}}$ represents a parameter matrix employed for linearly transforming the initial node representation, thereby augmenting the model's expressive capacity. $\mathbf{a}^{(k-1)} \in \mathbb{R}^{d'_h \times 1}$ is a trainable attention vector. ψ is the LeakyReLU activation function. We then normalize α_u^k across all neighboring nodes using the Softmax function

$$\alpha_u^k = \frac{\exp(\alpha_u^k)}{\sum_{u' \in \mathcal{N}(v)} \exp(\alpha_{u'}^k)}. \qquad (13)$$

Subsequently, the representation of node v at the kth layer, denoted as $h_v^{(k)}$, is obtained by concatenating its representation $h_v^{(k-1)}$ at the (k-1)th layer with the aggregated representation $h_{\mathcal{N}(v)}^{(k)}$ from its neighboring nodes. This operation can be formalized as follows:

$$h_v^{(k)} = \sigma(\mathbf{W}_a^{(k)}[h_v^{(k-1)} \mid\mid h_{\mathcal{N}(v)}^{(k)}]) \qquad (14)$$

where $\mathbf{W}_a^{(k)} \in \mathbb{R}^{d_k \times 2d_{k-1}}$ is the trainable parameter matrix at the kth layer. σ is the nonlinearity activation function.

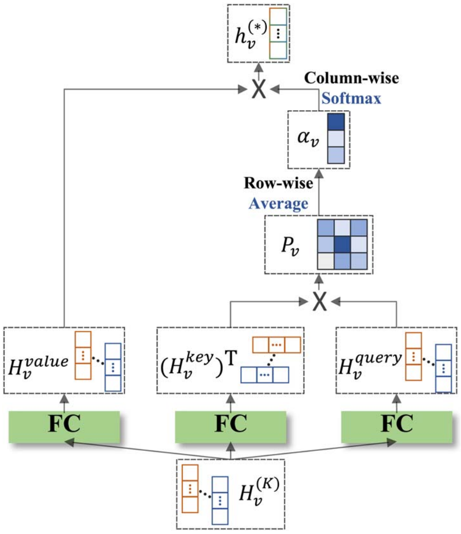


Fig. 4. Illustration of the multiview self-attention integration module, where FC represents the fully connected layer.

C. Multiview Self-Attention Integration Module

Inspired by scaled dot-product attention [14], we design a multiview self-attention integration module, as illustrated in Fig. 4, to fuse node representations from multiple views. For a target node $v \in V_{\mathcal{B}}$, we employ $H_v^{(K)} = [h_v^{(K)_1}, \ldots, h_v^{(K)_{\mathcal{M}}}]$ to represent the multiview representation tensor, which has a dimension of $d_K \times \mathcal{M}$. This tensor is derived from the multiview graph sampling and updating Module, wherein d_K represents the dimension of output node representations and $\mathcal{M}$ denotes the number of views. The self-attention integration function, which is used to calculate the weighted sum of feature representations from different views for node v, is expressed as follows:

$$h_v^{(*)} = f^{at}(\theta_{at},\ H_v^{(K)}) \tag{15}$$

where $h_v^{(*)} \in \mathbb{R}^{d_K}$ represents the unique feature representation of node v, and θ_{at} is the set of trainable parameters.

First, the input tensor $H_v^{(K)}$ is mapped to a feature representation triple $(H_v^{\text{query}}, H_v^{\text{key}}, H_v^{\text{value}})$ via three separate fully connected layer networks. Next, a pairwise matching matrix $P_v \in \mathbb{R}^{\mathcal{M}\times\mathcal{M}}$ is calculated as follows:

$$P_v = \frac{(H_v^{\text{key}})^{\text{T}}\ H_v^{\text{query}}}{\sqrt{d_K}} \tag{16}$$

where d_K represents the dimension of the feature representation H_v^{query}. $P_{v_{ij}}$ denotes the element located in the ith row and jth column of the matrix P_v, which denotes the correlation between two representation vectors $h_{v_i}^{(K)}$ and $h_{v_j}^{(K)}$. Based on P_v, the attention coefficients β_v among different views is calculated as follows:

$$\beta_v = \text{Sofmax}\left(\frac{1}{\mathcal{M}}\sum_{j=1}^{\mathcal{M}} P_{v_{ij}}\right) \tag{17}$$

where $\beta_v \in [0,1]^{\mathcal{M}}$ is a probability distribution, and the Sofmax function is defined as follows:

$$\text{Sofmax}(\beta_{v_i}) = \frac{\exp(\beta_{v_i})}{\sum_{i=1}^{\mathcal{M}}\exp(\beta_{v_i})}. \tag{18}$$

Finally, a weighted vector $h_v^{(*)}$ is computed as the final node feature representation that integrates global structural information from multiple views:

$$h_v^{(*)} = H_v^{\text{value}}\beta_v. \tag{19}$$

D. Classifier and Loss

In the transductive learning, the final document representation $h_{v_t}^{(*)}$ generated by the doc-level multiview global graph is inputted to the classifier for predicting the label $\hat{y}_t$. Specifically, this classifier is implemented as a fully connected layer network with Softmax function

$$\hat{y}_t = \text{Softmax}(\mathbf{W}_c h_{v_t}^{(*)} +\ \mathbf{b}_c) \tag{20}$$

where $\mathbf{W}_c \in \mathbb{R}^{|\mathcal{C}|\times d_K}$ and $\mathbf{b}_c \in \mathbb{R}^{|\mathcal{C}|}$ are the parameter matrix and bias vector, respectively. $|\mathcal{C}|$ represents the number of classes and d_K is the dimension of the final document representation $h_{v_t}^{(*)}$. It is worth noting that in inductive learning, the final document representation used for predicting the label is the output of the contextual encoder.

Finally, we minimize the cross-entropy loss using the mini-batch gradient descent, which is determined by the difference between the predicted values and the ground-truth values

$$\text{Loss} = -\sum_{i=1}^{|\mathcal{B}|}\sum_{j=1}^{|\mathcal{C}|} y_{t_{ij}}\log\hat{y}_{t_{ij}} \tag{21}$$

where y_t is the ground-truth label, $|\mathcal{C}|$ is the number of the classes, and $|\mathcal{B}|$ is the mini-batch size.

V. Experiments and Analysis

A. Datasets

To evaluate our proposed framework, we conduct extensive experiments on five well-known document classification datasets that have long been used as benchmarks for evaluating various GNN-based models. The list of these datasets is provided as follows.

1) *MR*[1] is a sentiment binary classification dataset specifically focused on *M*ovie *R*eview.
2) *R8*[2] is a subset of the *R*euters news dataset, which contains eight (*8*) topics for document classification.
3) *20NG*[3] is a dataset for document classification, consisting of news articles from twenty (*20*) different *N*ews *G*roups.
4) *Ohsumed*[4] is a dataset for medical document classification, covering twenty-three medical topic categories.

[1] https://github.com/mnqu/PTE/tree/master/data/mr
[2] https://www.cs.umb.edu/smimarog/textmining/datasets/
[3] http://qwone.com/jason/20Newsgroups/
[4] http://disi.unitn.it/moschitti/corpora.htm

TABLE I
SUMMARY STATISTICS OF BENCHMARK DATASETS USED IN EXPERIMENTS AND ANALYSIS

Dataset	#Docs	#Train	#Test	#Classes	Avg_Length
MR	10 662	7108	3554	2	18.12
R8	7674	5485	2189	8	64.96
20NG	18 846	11 314	7532	20	221.26
Ohsumed	7400	3357	4043	23	135.82
R52	9100	6532	2568	52	69.04

5) *R52*[5] offers a wider range of topics, specifically fifty-two (*52*) in total, compared to the R8 dataset.

The summary statistics of benchmark datasets are presented in Table I. To ensure impartiality, we follow the text preprocessing procedures outlined in the original papers of comparison models [20], [22], [24], [31].

B. Parameter Settings

All our models are trained and tested on a single Tesla V100 PCIe 32 GB GPU. Following a common practice employed by many comparison models, we randomly select 10% of documents from the training set as the validation set. We train our models using the mini-batch stochastic gradient descent (SGD) algorithm, with a batch size of 64 for R8, Ohsumed, and R52 datasets, and a batch size of 128 for MR and 20NG datasets. Ulteriorly, the initial learning rate of the Adam optimizer is 0.001 (it is 0.005 for MR dataset), and the L2 weight decay is 0.0002. The dropout rate is 0.5.

The statistics-based view uses a sliding window size of 20, but it is set to 10 for the MR dataset due to the shorter document length. The similarity-based view utilizes pretrained word embeddings with 300 dimensions from *GloVe* [44]. For the topic-based view, we set the number of potential clusters in each dataset to 50. We evaluate the impact of different numbers of neighboring nodes (K_w and K_t), ranging from 10 to 400, on the classification results, and aggregate the *second*-order neighboring information for each word or document node.

Furthermore, we explored the impact of employing different contextual encoders on performance, including WideNLP, LSTM, and the GNN-based model TextING. These networks focus on different aspects of intradoc relations. Specifically, WideNLP ignores the sequential information within documents, LSTM models the simplest sequential relations, and TextING constructs a graph for each document based on word co-occurrence relations, thereby learning the personalized context information within documents. Throughout the forward propagation process of the entire framework, we maintain consistent dimensions for word and document representations, excluding the final classifier. We provide more comprehensive parameter settings in our source code.[6]

C. Comparison Models

In addition to the existing GNN-based inductive/transductive document classification models and those that combine GNNs with LLMs, our comparison also includes average-based models that represent documents by averaging word embeddings without considering the position and order of words within documents, as well as sequence-based models that prioritize sequence modeling for documents. We have categorized all the comparison models and listed them as follows.

1) Average-Based Models:

a) *PV-DBOW* [53] is a paragraph vector model that ignores word order and utilizes logistic regression as its classifier.

b) *fastText* [54] utilizes simple averaging of *n*-gram word embeddings as the representation for a document and employs a linear classifier for classification purposes.

c) *SWEM* [55] is a word embedding model that employs the pooling strategy.

d) *WideMLP* [12] is a MLP network with a single wide hidden layer.

2) Sequence-Based Models:

a) *PV-DM* [53] is similar to PV-DBOW except that PV-DM incorporates word order information.

b) *CNN* [7] is a multilayer CNN, where we use pretrained word embeddings (i.e., *GloVe*) to initialize the word representations.

c) *LSTM* [10] is a well-known contextual encoder that consists of two layers of LSTM networks.

d) *BERT/RoBERTa* [16], [17] is large-scale pretrained models based on Transformer. They has achieved notable advancements in NLP tasks.

3) GNN-Based Inductive Models:

a) *TextGCN* [20] is proposed by Ragesh et al. [30] to achieve predictions on unseen documents by preserving pretrained word embeddings.

b) *Text-level-GNN* [21] achieves GNN-based document representation learning by building intradoc graphs for each document with global parameters sharing.

c) *TextING* [22] is similar to text-level-GNN, but it utilizes edges between word nodes to depict the co-occurrence information of words present in the corpus.

d) *HyperGAT* [23] proposes to model documents using intradoc hypergraphs and introduces hypergraph attention networks based on the dual-attention mechanism to support text representation learning on hypergraphs.

e) *HeteGCN* [30] utilizes word node embeddings from specific layer outputs as pretrained features and performs inference on unseen documents using a GCN layer equipped with a Softmax function.

f) *TextSSL* [24] constructs sentence-level graphs for each document to enable intra- and intersentence message passing, updating structures via sparse learning and Gumbel-Softmax.

g) *ContGCN+BERT/RoBERTa* [42] employs a new "all-token-any-document" paradigm to dynamically update graphs for online document classification and uses LLMs's pretrained embeddings to enhance node initialization.

h) *Text-FCG+BERT* [36] constructs a single graph for each document with typed edges representing various context relations, incorporates a gated recurrent unit (GRU) to

[5] https://www.cs.umb.edu/smimarog/textmining/datasets/

[6] https://github.com/962541229/TL-MGG-OL-MGG

TABLE II
CLASSIFICATION ACCURACY (%) OF VARIOUS GNN-BASED TRANSDUCTIVE MODELS ON FIVE DATASETS, WHERE AVERAGE- AND SEQUENCE-BASED MODELS WERE ALSO USED FOR COMPARISON

Types	Models	20NG	R8	R52	Ohsumed	MR
Average-based Models	PV-DBOW [27]	74.36 (0.18)	85.87 (0.10)	78.29 (0.11)	46.65 (0.19)	61.09 (0.10)
	fastText [23]	79.38 (0.30)	96.13 (0.21)	92.81 (0.09)	57.70 (0.49)	75.14 (0.20)
	SWEM [23]	85.16 (0.29)	95.32 (0.26)	92.94 (0.24)	63.12 (0.55)	76.65 (0.63)
	WideMLP [12]	83.31 (0.22)	97.27 (0.12)	93.89 (0.16)	63.95 (0.13)	76.72 (0.26)
Sequence-based Models	PV-DM [27]	51.14 (0.22)	52.07 (0.04)	44.92 (0.05)	29.50 (0.07)	59.47 (0.38)
	CNN [7]	-	95.70 (0.50)	87.60 (0.50)	58.40 (1.00)	-
	LSTM [10]	75.43 (1.72)	96.09 (0.19)	90.48 (0.86)	51.10 (1.50)	77.33 (0.39)
GNN-based Models (Inductive)	Text-level-GNN [21]	-	97.80 (0.20)	94.60 (0.30)	69.40 (0.60)	-
	TextING [22]	-	97.34 (0.25)	93.73 (0.47)	67.95 (0.52)	78.93 (0.65)
	HyperGAT [23]	84.65 (0.31)	96.82 (0.21)	94.15 (0.18)	66.39 (0.65)	77.36 (0.22)
	HeteGCN [30]	84.59 (0.14)	97.17 (0.33)	93.89 (0.45)	63.79 (0.80)	75.62 (0.26)
	TextSSL [24]	85.26 (0.28)	97.81 (0.14)	95.48 (0.26)	70.59 (0.38)	79.74 (0.19)
	OL-MGG-WideMLP (Ours)	**87.46** (0.41)	**97.94** (0.32)	**95.57** (0.33)	68.84 (0.42)	79.50 (0.21)
	OL-MGG-LSTM (Ours)	84.02 (0.26)	97.54 (0.28)	92.39 (1.26)	59.34 (1.24)	81.13 (0.23)
	OL-MGG-TextING (Ours)	85.19 (0.39)	97.58 (0.36)	95.03 (0.17)	**71.19** (0.42)	**82.09** (0.31)
GNN-based Models (Transductive)	TextGCN [20]	86.34 (0.31)	97.07 (0.21)	93.56 (0.34)	68.36 (0.34)	76.74 (0.23)
	SGC [28]	88.50 (0.10)	97.20 (0.10)	94.00 (0.20)	68.50 (0.30)	75.90 (0.30)
	DHTG [37]	87.13 (0.07)	97.33 (0.06)	93.93 (0.10)	68.80 (0.33)	77.21 (0.11)
	TensorGCN [29]	87.74 (0.05)	98.04 (0.08)	**95.05** (0.11)	70.11 (0.24)	77.91 (0.07)
	HeteGCN [30]	87.15 (0.15)	97.24 (0.51)	94.35 (0.25)	68.11 (0.70)	76.71 (0.33)
	CGA2TC [31]	-	97.76 (0.19)	94.47 (0.16)	70.62 (0.45)	77.80 (0.29)
	TL-MGG-WideMLP (Ours)	**88.96** (0.35)	**98.12** (0.12)	94.54 (0.31)	71.37 (0.42)	80.82 (0.19)
	TL-MGG-LSTM (Ours)	86.38 (0.12)	97.39 (0.17)	92.08 (0.24)	62.36 (0.15)	81.32 (0.25)
	TL-MGG-TextING (Ours)	87.21 (0.40)	97.92 (0.33)	93.48 (0.34)	**72.55** (0.25)	**82.88** (0.42)

Note: We run our model ten times and report the mean, with the standard deviation (±) enclosed in parentheses. The mark "-" indicates an unreported value in all existing works. The bold values indicate the best performance, while the underlined values represent the second-best performance.

enrich node sequence representations, and engages in multitask learning with LLMs.

i) *HDGAT+BERT* [43] utilizes multilevel semantic embeddings and a novel attention mechanism to effectively capture and integrate complex semantic relations, leveraging pretrained sentence embeddings for node initialization.

4) GNN-Based Transductive Models:

a) *TextGCN* [20] constructs a single global graph consisting of documents and words and learns representations of both documents and words simultaneously through the multilayer spectral GCN.

b) *SGC* [28] improves the spectral GCN by eliminating the weight matrix between nonlinear and folded consecutive layers, thereby reducing unnecessary complexity and redundant computations.

c) *DHTG* [37] constructs a unified graph incorporating word, topic, and document information. It treats document classification as a task of generating node labels, which are dynamically updated using variational inference and GCN.

d) *TensorGCN* [29] utilizes a text graph tensor to capture diverse context information and effectively integrates heterogeneous information from multiple graphs through intra- and intergraph propagation.

e) *HeteGCN* [30] integrates heterogeneous GCNs with TextGCN's individual graphs, employing cross-layer compatible graphs to merge multilayer representations as needed.

f) *CGA2TC* [31] refines text graph topology using noise and centrality-based augmentations, improving node representations through combined optimization of contrastive and classification losses.

g) *TextGCN+BERT/RoBERTa* [39] combines LLMs with TextGCN, leveraging the strengths of both to optimize node representations.

h) *TextGCL+BERT* [40] incorporates graph contrastive learning by constructing contrasting views of word- and doc-graphs, optimizing node representations with BERT's pretrained embeddings.

i) *GRTE+BERT* [41] combines GNNs with LLMs, constructing diverse graph structures to model complex textual relations and optimize node representations.

D. Performance Analysis

1) Inductive Learning: Table II presents the performance of our inductive model OL-MGG in document classification, compared with average-, sequence-, and other GNN-based models (Inductive). Specifically, OL-MGG utilizes WideMLP, LSTM, and TextING as contextual encoders to generate the initial representations of documents. These variants are referred to as OL-MGG-WideMLP, OL-MGG-LSTM, and OL-MGG-TextING, respectively. Overall, OL-MGG demonstrates superior performance across all datasets, evidencing its effectiveness in learning both the intradoc local context relations and the interdoc global distribution relations. In detail, OL-MGG-WideMLP

TABLE III
Classification Accuracy (%) of Various GNN-Based Models Combined With BERT

Types	Models	20NG	R8	R52	Ohsumed	MR
LLMs (BERT or RoBERTa)	BERT [39]	85.20 (0.31)	97.73 (0.21)	96.22 (0.34)	70.53 (0.34)	85.71 (0.31)
	RoBERTa [39]	83.44 (0.41)	96.98 (0.29)	96.07 (0.32)	70.21 (0.61)	88.96 (0.45)
Combined with LLMs (Inductive)	ContGCN + BERT [42]	89.40 (-)	98.30 (-)	96.90 (-)	73.10 (-)	86.40 (-)
	ContGCN + RoBERTa [42]	90.10 (-)	98.60 (-)	96.60 (-)	73.40 (-)	91.30 (-)
	Text-FCG + BERT [36]	90.38 (0.26)	**98.93** (0.18)	**98.14** (0.33)	73.84 (0.31)	86.89 (0.18)
	HDGAT + BERT [43]	90.50 (-)	98.21 (-)	96.68 (-)	**76.64** (-)	85.63 (-)
	OL-MGG (Ours)	87.46 (0.41)	97.94 (0.32)	95.57 (0.33)	71.19 (0.42)	82.09 (0.31)
	OL-MGG + BERT (Ours)	91.12 (0.22)	98.68 (0.36)	97.28 (0.31)	74.32 (0.39)	88.37 (0.31)
	OL-MGG + RoBERTa (Ours)	**91.43** (0.27)	98.74 (0.19)	97.02 (0.37)	74.55 (0.23)	**92.21** (0.27)
Combined with LLMs (Transductive)	TextGCN + BERT [39]	89.30 (-)	98.10 (-)	96.60 (-)	72.80 (-)	86.00 (-)
	TextGCN + RoBERTa [39]	89.50 (-)	98.20 (-)	96.10 (-)	72.80 (-)	89.70 (-)
	TextGCL + BERT [40]	90.20 (-)	98.20 (-)	96.80 (-)	73.30 (-)	86.20 (-)
	GRTE + BERT [41]	87.52 (-)	**98.72** (-)	96.25 (-)	71.70 (-)	89.69 (-)
	TL-MGG (Ours)	88.96 (0.35)	98.12 (0.12)	94.54 (0.31)	72.55 (0.25)	82.88 (0.42)
	TL-MGG + BERT (Ours)	91.26 (0.19)	98.38 (0.31)	**97.25** (0.27)	74.41 (0.31)	88.91 (0.22)
	TL-MGG + RoBERTa (Ours)	**91.55** (0.38)	98.49 (0.24)	96.89 (0.44)	**74.88** (0.45)	**91.78** (0.19)

Note: The bold values indicate the best performance, while the underlined values represent the second-best performance.

achieves the highest classification accuracy on three datasets related to the news domain: 20NG, R8, and R52. This success is attributed to its ability to capture global distribution information significant within the word-level multiview global graph, such as word co-occurrence, synonyms, and thematic connections. Conversely, OL-MGG-TextING performs best on the Ohsumed and MR datasets. The MR dataset primarily involves sentiment classification, while Ohsumed presents challenges due to its extensive use of specialized terminology and longer sentences. These factors make it difficult for average- and sequence-based models to learn the overall semantics of documents from sequences of word representations. In contrast, TextING enhances document representation by capturing personalized local context relations, providing additional distinguishable information.

Moreover, compared to baseline contextual encoders, OL-MGG series consistently demonstrates significant improvements across all datasets. Specifically, OL-MGG-WideMLP shows an average increase of 2.83% compared to WideMLP, OL-MGG-LSTM exhibits a 4.80% improvement over LSTM, and OL-MGG-TextING achieves a 1.98% enhancement relative to TextING. These results indicate that OL-MGG effectively captures the diverse global distribution information between documents, complementing networks that focus on learning indradoc relations. Overall, our model demonstrates remarkable adaptability and robust performance enhancements across various datasets.

2) Transductive Learning: Table II also displays the document classification performance of our transductive model, TL-MGG, compared to existing GNN-based models (Transductive). As a whole, TL-MGG outperforms all other models on 20NG, R8, Ohsumed, and MR datasets, while achieving the second-highest ranking on the R52 dataset. Notably, on the MR dataset, TL-MGG-TextING outperforms the CGA2TC, the latest state-of-the-art GNN-based transductive model, by 5.08% in accuracy. This fully demonstrates the advanced nature of our framework. When the context modeling process is no longer ignored, GNN-based document classification models show improved performance, particularly in downstream tasks such as sentiment analysis that heavily relies on contextual semantic and sequential information.

TensorGCN also attempts to construct multiple graphs for document representation learning, similar to TL-MGG. However, TensorGCN utilizes these graphs to capture the semantic, syntactic, and sequential relations between a document and its constituent words. In contrast, our multiview global graph is utilized for learning the global co-occurrence, semantics, and clustering relations among words or documents while leveraging contextual encoders to focus on modeling internal document context. This approach fully capitalizes on the strengths of GNNs in learning complex global relations. Consequently, TL-MGG exploits the complementarity of networks with distinct characteristics, exhibiting superior performance across four datasets compared to TensorGCN.

3) Combined with BERT: Following existing methods, we combined BERT with our models for multitask learning and adjusted the final training objective through linear interpolation, as demonstrated in the following equation:

$$\hat{y} = \lambda \hat{y}_{\text{BERT}} + (1-\lambda)\hat{y}_{\text{OL-MGG / TL-MGG}} \tag{22}$$

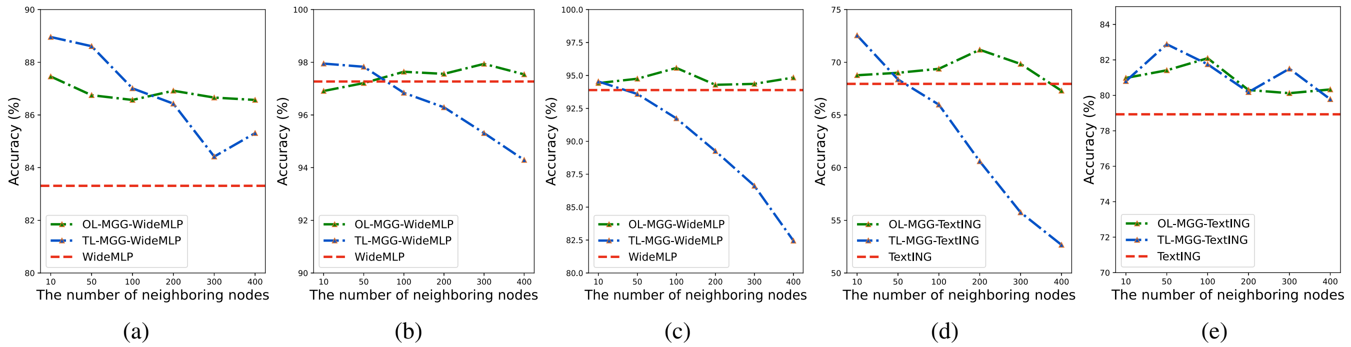
where λ is to adjust the weight ratio between the predicted outcomes of both models. Per convention [39], we set $\lambda = 0.9$.

Table III displays the comparative performance of our framework combined with LLMs across two learning settings. In inductive learning, OL-MGG + BERT/RoBERTa achieved optimal or near-optimal results across all datasets. Notably, although OL-MGG + BERT/RoBERTa performed slightly below the baseline Text-FCG + BERT on the R8 and R52 datasets, with an average deficit of only 0.52%, it outperformed Text-FCG + BERT by 1.48% on the sentiment classification

TABLE IV
ABLATION EXPERIMENTS TO ANALYZE THE EFFECTIVENESS OF VARIOUS VIEWS

Models	20NG	R8	R52	Ohsumed	MR
TextGCN [20]	86.34 (0.31)	97.07 (0.21)	93.56 (0.34)	68.36 (0.34)	76.74 (0.23)
Only statistics-based view	87.11 (0.11)	96.95 (0.33)	93.54 (0.27)	71.33 (0.28)	81.15 (0.19)
Only similarity-based view	86.66 (0.12)	96.87 (0.30)	93.43 (0.18)	70.65 (0.34)	81.64 (0.35)
Only topics-based view	86.67 (0.23)	96.71 (0.29)	93.41 (0.23)	71.21 (0.39)	80.21 (0.48)
w/o statistics-based view	87.31 (0.24)	96.94 (0.37)	93.68 (0.19)	71.35 (0.26)	81.33 (0.21)
w/o similarity-based view	88.41 (0.26)	96.96 (0.41)	94.06 (0.17)	72.12 (0.41)	81.18 (0.27)
w/o topics-based view	87.75 (0.11)	97.52 (0.28)	93.71 (0.22)	71.69 (0.25)	81.68 (0.19)
TL-MGG	**88.96** (0.35)	**98.12** (0.12)	**94.54** (0.31)	**72.55** (0.25)	**82.88** (0.42)

The bold values indicate the best performance.



(a) (b) (c) (d) (e)

Fig. 5. Effect of varying numbers of neighboring nodes on classification accuracy across different datasets, these include: (a) 20NG dataset; (b) R8 dataset; (c) R52 dataset; (d) Ohsumed dataset; and (e) MR dataset.

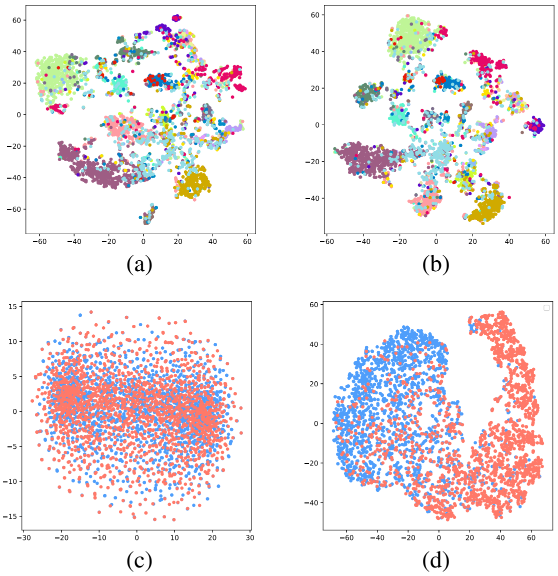
dataset MR. This can be attributed to the key role played by OL-MGG in enhancing BERT's rich language features with global information from the local corpus. In transductive learning, TL-MGG + BERT/RoBERTa yielded the best results on four datasets. This is due to our customized multiview graph sampling and updating module, which enables efficient node learning on the global graph without being limited by the graph's scale. TL-MGG can optimize parameters via mini-batch gradient descent, facilitating multitask learning with LLMs and enabling simultaneous optimization of multiple models. In contrast, other GNN-based transductive learning models employ a full-batch training approach, which requires using the global graph's adjacency matrix to update all nodes' representations in a single iteration. This approach imposes memory constraints during training, allowing only the initialization of node representations with LLMs, without fine-tuning in combination with LLMs.

E. Ablation Analysis

We conduct ablation experiments to validate the effectiveness of our constructed multiview global graph and analyze the sensitivity of different datasets to different views. The summarized results are presented in Table IV. First, we note that TL-MGG consistently outperforms scenarios with a single or dual view across all datasets, highlighting the effectiveness of the multiview global graph in utilizing complementarity and synergy among different views. Second, we find that the MR dataset is most responsive to the similarity-based view, this is due to the crucial role of context information in sentiment classification [56]. On the other hand, other datasets are more sensitive to statistics- and topic-based views, which are closely related to their task attributes, namely document topic classification. Third, it is noteworthy that our model surpasses TextGCN in the only statistics-based view scenario. This comparison is significant because both models rely solely on global statistical relations, highlighting the superiority of our proposed framework over existing GNN-based transductive frameworks.

F. Influence Analysis

The line plots in Fig. 5 demonstrate the impact of varying the number of neighboring nodes (from 10 to 400) on classification outcomes in both word- and doc-level graphs. Specifically, in transductive learning, TL-MGG (depicted by the blue line) is used to represent the features of document nodes, with the number of neighboring nodes for word nodes held constant. In inductive learning, OL-MGG (shown with a green line) illustrates the features of word nodes, while the baseline models, WideMLP and TextING (represented by a red dashed line), serve as reference points. It is apparent that OL-MGG-WideMLP and OL-MGG-TextING each surpass WideMLP and TextING in most instances, respectively. Furthermore, the alteration in the number of neighboring nodes notably affects the classification performance of TL-MGG compared to OL-MGG. This indicates caution should be exercised when establishing edges among document nodes.



(a) (b)

(c) (d)

Fig. 6. t-SNE visualization presents the representations of test documents from various models, including: (a) TextING; (b) OL-MGG-TextING; (c) TextGCN; and (d) TL-MGG-TextING. The test documents in (a) and (b) are from the Ohsumed dataset, while those in (c) and (d) are from the MR dataset. Distinct colors denote distinct classes to which the documents belong.

The observed disparity is attributed to the direct impact of document node representations on the classifier. In detail, the classifier faces a significant challenge in effectively distinguishing document classes due to the over-smoothing issue [57] arising from dense graphs. Inversely, the smoothness of neighboring word nodes' representations, resulting from their semantic proximity, has minimal impact on the overall semantics of the document. Consequently, word nodes demonstrate enhanced resilience to the over-smoothing issue.

G. Visualization Analysis

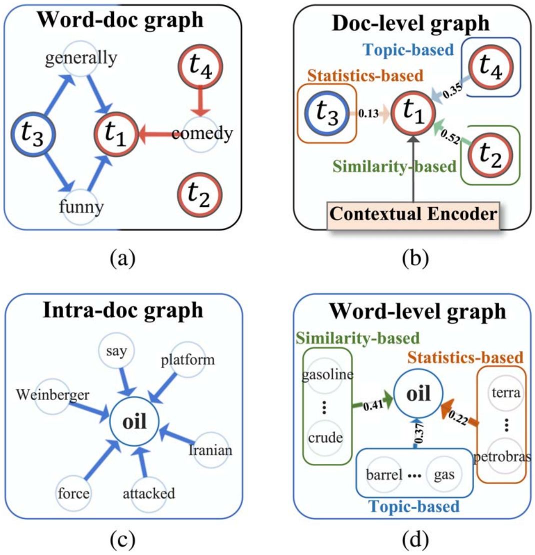
To qualitatively demonstrate the superiority of our framework, we provide visual distributions of test document representations learned by our framework and other existing representative models in different learning settings. As shown in Fig. 6, we utilize the t-SNE tool to reduce the dimensionality of test document representations that will be inputted into the classifier for all models and categorize them by their ground-truth label using distinct colors. From Fig. 6(a) and 6(b), one observes that OL-MGG-TextING outperforms TextING in learning document representations, showing greater differentiation across different classes and forming more compact clusters for documents within the same class. Fig. 6(c) and 6(d) demonstrates that TL-MGG-TextING effectively distinguishes documents with different sentiment polarities, which are mixed in TexGCN. These results demonstrate the superiority of our framework.

H. Case Study

To improve interpretability and facilitate a more intuitive understanding of the impressive results achieved by our framework, we use specific cases presented in Table V to clarify the actual neighboring nodes of those documents or words in various models under different frameworks, as well as how they are updated.

TABLE V
Collection of Documents Used in Case Study

Tabs	Documents	Dataset	Label
t_1	This comedy is *generally* quite *funny*.	MR	*Positive*
t_2	It's a very valuable film.	MR	*Positive*
t_3	It's not that *funny*, which is just *generally* insulting.	MR	*Negative*
t_4	An enjoyable feel-good family *comedy* regardless of race.	MR	*Positive*
t_5	Weinberger say force attacked Iranian *oil* platform.	R8	*Crude*



(a) (b)

(c) (d)

Fig. 7. Illustration of the neighboring nodes and information flow of the documents or words to be learned in different models, these include: (a) TextGCN; (b) TL-MGG-LSTM; (c) TextING; and (d) OL-MGG-WideMLP, where the arrows of different colors represent information from different sources, the direction of the arrow indicates the flow of information, and the weight that depicts importance is marked on the arrow. Specifically, in (a) TextGCN and (b) TL-MGG-LSTM, the different colors of document nodes correspond to different sentiment polarities.

As shown in Fig. 7, in the transductive learning, when considering t_1 as the target document, its neighboring nodes in TextGCN are illustrated in Fig. 7(a). t_1 is updated by integrating features from its neighboring nodes. Differently, in TL-MGG-LSTM, as depicted in Fig. 7(b), we first obtain the initial embedding of t_1 through a contextual encoder and then integrate features from neighboring nodes across multiple views in a weighted manner. TL-MGG-LSTM effectively captures both local contextual and diverse global structural information during training. These advantages enabled TL-MGG-LSTM to make correct predictions on the sentiment polarity of t_1, while TextGCN produced incorrect results.

In the inductive learning, assuming the target document is t_5, TextING and OL-MGG-WideMLP employ different underlying graphs to select neighboring nodes for the keyword "oil" within

t_5. In TextING, as shown in Fig. 7(c), neighboring nodes are other words within t_5 due to its underlying intradoc graph. Meanwhile, in OL-MGG-WideMLP, as depicted in Fig. 7(d), neighboring nodes are selected in the vocabulary based on their strongest correlations with "oil" regarding statistics-, similarity-, and topic-based views. Obviously, OL-MGG-WideMLP incorporates richer information for word nodes compared to TextING, resulting in a closer representation of the keyword "oil" with other keywords under the class *crude*. This allowed OL-MGG-WideMLP to make more confident predictions about the class of t_5, even when we used the simple WideMLP as the contextual encoder.

In summary, our proposed framework constructs more reasonable underlying graphs and explores multiple potential relations among words or documents from different views, resulting in improved performance on document classification.

VI. Discussion

We showcase the impressive ability of GNNs to aggregate global information for document representation learning. However, a potential limitation is that both TL-MGG and OL-MGG are based on static word embeddings and cannot fully integrate with advanced LLMs that generate context-aware and dynamic word embeddings. Therefore, our future research will focus on constructing and training dynamic graphs, including both the dynamics of node representations and underlying graph structures. While current state-of-the-art LLMs primarily focuses on extracting deep semantic information from documents, they often fall short in providing targeted global information tailored to domain-specific tasks [39]. Conversely, GNN-based models possess the potential to accomplish this objective.

VII. Conclusion

In this article, we propose a comprehensive GNN-based representation learning framework with multiview global graphs for document classification. Our framework addresses the limited ability of existing frameworks to balance the acquisition of global and local information. Moreover, it boasts diverse and scalable underlying global graph structures. In training mode, our framework utilizes the mini-batch algorithm for learning, possesses the capability to generalize to larger datasets and can be jointly trained with the most advanced LLMs model, further enhancing the accuracy of document classification. The experimental results demonstrate that the model constructed within our proposed framework achieves superior performance compared to the latest GNN-based models in both transductive and inductive learning.

References

[1] D. Lazer et al., "Social science. computational social science," *Science*, vol. 323, no. 5915, pp. 721–723, 2009.

[2] K. Chakraborty, S. Bhattacharyya, and R. Bag, "A survey of sentiment analysis from social media data," *IEEE Trans. Comput. Social Syst.*, vol. 7, no. 2, pp. 450–464, Apr. 2020.

[3] Q. Li, T. Zhang, C. L. P. Chen, K. Yi, and L. Chen, "Residual GCB-net: Residual graph convolutional broad network on emotion recognition," *IEEE Trans. Cogn. Develop. Syst.*, vol. 15, no. 4, pp. 1673–1685, Dec. 2023.

[4] A. Lao et al., "Frequency spectrum is more effective for multimodal representation and fusion: A multimodal spectrum rumor detector," in *Proc. Assoc. Adv. Artif. Intell. Assoc. Adv. Artif. Intell.* (*AAAI,*) Palo Alto, CA, USA: AAAI Press, 2024, pp. 18426–18434.

[5] G. Fei, Y. Cheng, W. Ma, C. Chen, S. Wen, and G. Hu, "Real-time detection of COVID-19 events from Twitter: A spatial-temporally bursty-aware method," *IEEE Trans. Comput. Social Syst.*, vol. 10, no. 2, pp. 656–672, Apr. 2022.

[6] L. Xiao, Q. Zhang, C. Shi, S. Wang, U. Naseem, and L. Hu, "MSynFD: Multi-hop syntax aware fake news detection," in *Proc. ACM Web Conf.,* New York, NY, USA: ACM, 2024, pp. 4128–4137.

[7] P. Kim and P. Kim, "Convolutional neural network," in *MATLAB Deep Learning: With Machine Learning, Neural Networks and Artificial Intelligence*, New York, NY, USA: Apress, 2017, pp. 121–147.

[8] Y. Kim, "Convolutional neural networks for sentence classification," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP),* Association for Computational Linguistics, 2014, pp. 1746–1751.

[9] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[10] P. Liu, X. Qiu, and X. Huang, "Recurrent neural network for text classification with multi-task learning," in *Proc. 25th Int. Joint Conf. Artif. Intell. (IJCAI)*, 2016, pp. 2873–2879.

[11] Q. Zhang et al., "Rumor detection with hierarchical representation on bipartite ad hoc event trees," *IEEE Trans. Neural Networks Learn. Syst.*, early access, Dec. 23, 2023, doi: 10.1109/TNNLS.2023.3274694.

[12] L. Galke and A. Scherp, "Bag-of-words vs. graph vs. sequence in text classification: Questioning the necessity of text-graphs and the surprising strength of a wide MLP," in *Proc. 60th Annu. Meeting Assoc. Comput. Linguistics (Volume 1: Long Papers) (ACL)*, 2022, pp. 4038–4051.

[13] Z. Gong et al., "Lite-mind: Towards efficient and robust brain representation learning," in *ACM Multimedia 2024*, New York, NY, USA: ACM, 2024. [Online]. Available: https://openreview.net/forum?id=kYhRv9cw9i

[14] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 5998–6008.

[15] Z. Wu, Q. Zhang, D. Miao, K. Yi, W. Fan, and L. Hu, "HyDiscGAN: A hybrid distributed cGAN for audio-visual privacy preservation in multimodal sentiment analysis," in *Proc. 33rd Int. Joint Conf. Artif. Intell. (IJCAI)*, K. Larson, Ed., Aug. 2024, pp. 6550–6558.

[16] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Human Lang. Technol., (Volume 1: Long Short Papers) (NAACL),* 2019, pp. 4171–4186.

[17] Y. Liu et al., "RoBERTa: A robustly optimized BERT pretraining approach," 2019, *arXiv:1907.11692*.

[18] G. Bao, Q. Zhang, D. Miao, Z. Gong, and L. Hu, "Multimodal federated learning with missing modality via prototype mask and contrast," 2023, *arXiv:2312.13508*.

[19] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learn. Representations (ICLR)*, 2016.

[20] L. Yao, C. Mao, and Y. Luo, "Graph convolutional networks for text classification," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, vol. 33, 2019, pp. 7370–7377.

[21] L. Huang, D. Ma, S. Li, X. Zhang, and H. Wang, "Text level graph neural network for text classification," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP),* Stroudsburg, PA, USA: Association for Computational Linguistics, 2019, pp. 3444–3450.

[22] Y. Zhang, X. Yu, Z. Cui, S. Wu, Z. Wen, and L. Wang, "Every document owns its structure: Inductive text classification via graph neural networks," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics (ACL)*, 2020, pp. 334–339.

[23] K. Ding, J. Wang, J. Li, D. Li, and H. Liu, "Be more with less: Hypergraph attention networks for inductive text classification," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2020, pp. 4927–4936.

[24] Y. Piao, S. Lee, D. Lee, and S. Kim, "Sparse structure learning via graph neural networks for inductive document classification," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, vol. 36, 2022, pp. 11 165–11 173.

[25] W. Medhat, A. Hassan, and H. Korashy, "Sentiment analysis algorithms and applications: A survey," *Ain Shams Eng. J.*, vol. 5, no. 4, pp. 1093–1113, 2014.

[26] S. Hingmire, S. Chougule, G. K. Palshikar, and S. Chakraborti, "Document classification by topic labeling," in *Proc. 36th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2013, pp. 877–880.

[27] Y. Dai et al., "Graph fusion network for text classification," *Knowl. Based Syst.*, vol. 236, 2022, Art. no. 107659.

[28] F. Wu, A. Souza, T. Zhang, C. Fifty, T. Yu, and K. Weinberger, "Simplifying graph convolutional networks," in *Proc. Int. Conf. Mach. Learn. (ICML),* PMLR, 2019, pp. 6861–6871.

[29] X. Liu, X. You, X. Zhang, J. Wu, and P. Lv, "Tensor graph convolutional networks for text classification," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, vol. 34, 2020, pp. 8409–8416.

[30] R. Ragesh, S. Sellamanickam, A. Iyer, R. Bairi, and V. Lingam, "HeteGCN: Heterogeneous graph convolutional networks for text classification," in *Proc. 14th ACM Int. Conf. Web Search Data Mining (WSDM)*, 2021, pp. 860–868.

[31] Y. Yang, R. Miao, Y. Wang, and X. Wang, "Contrastive graph convolutional networks with adaptive augmentation for text classification," *Inf. Process. Manage.*, vol. 59, no. 4, 2022, Art. no. 102946.

[32] G. Ciano, A. Rossi, M. Bianchini, and F. Scarselli, "On inductive–transductive learning with graph neural networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 2, pp. 758–769, Feb. 2022.

[33] K. Yao, J. Liang, J. Liang, M. Li, and F. Cao, "Multi-view graph convolutional networks with attention mechanism," *AI*, vol. 307, 2022, Art. no. 103708.

[34] C. Xu, D. Tao, and C. Xu, "Multi-view intact space learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 12, pp. 2531–2544, Dec. 2015.

[35] Y. Liu, R. Guan, F. Giunchiglia, Y. Liang, and X. Feng, "Deep attention diffusion graph neural networks for text classification," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP),* Stroudsburg, PA, USA: Association for Computational Linguistics, 2021, pp. 8142–8152.

[36] Y. Wang, C. Wang, J. Zhan, W. Ma, and Y. Jiang, "TextFCG: Fusing contextual information via graph learning for text classification," *Expert Syst. Appl.*, vol. 219, 2023, Art. no. 119658.

[37] Z. Wang, C. Wang, H. Zhang, Z. Duan, M. Zhou, and B. Chen, "Learning dynamic hierarchical topic graph with graph convolutional network for document classification," in *Proc. Int. Conf. Artif. Intell. Statist. (AISTATS),* PMLR, 2020, pp. 3959–3969.

[38] H. Linmei, T. Yang, C. Shi, H. Ji, and X. Li, "Heterogeneous graph attention networks for semi-supervised short text classification," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP),* Stroudsburg, PA, USA: Association for Computational Linguistics, 2019, pp. 4821–4830.

[39] Y. Lin et al., "BERTGCN: Transductive text classification by combining GNN and BERT," in *Proc. Findings Assoc. Comput. Linguistics (ACL-IJCNLP)*, 2021, pp. 1456–1462.

[40] Y. Zhao and X. Song, "TextGCL: Graph contrastive learning for transductive text classification," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN),* Piscataway, NJ, USA: IEEE Press, 2023, pp. 1–8.

[41] A. C. Aras, T. Alikasifoglu, and A. Koç, "Graph receptive transformer encoder for text classification," *IEEE Trans. Signal Inf. Process. Over Netw.*, vol. 10, pp. 347–359, 2024.

[42] T. Wu, Q. Liu, Y. Cao, Y. Huang, X.-M. Wu, and J. Ding, "Continual graph convolutional network for text classification," in *Proc. AAAI Conf. Artif. Intell.*, vol. 37, no. 11, 2023, pp. 13754–13762.

[43] M. Lin, T. Wang, Y. Zhu, X. Li, X. Zhou, and W. Wang, "A heterogeneous directed graph attention network for inductive text classification using multilevel semantic embeddings," *Knowl.* Based Syst., vol. 295, 2024, Art. no. 111797.

[44] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global vectors for word representation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1532–1543.

[45] D. Eppstein, M. S. Paterson, and F. F. Yao, "On nearest-neighbor graphs," *Discrete Comput. Geometry*, vol. 17, pp. 263–282, Apr. 1997.

[46] S. Niwattanakul, J. Singthongchai, E. Naenudorn, and S. Wanapu, "Using of Jaccard coefficient for keywords similarity," in *Proc. Int. Multiconf. Eng. Comput. Sci.*, vol. 1, no. 6, 2013, pp. 380–384.

[47] K. Yuan, D. Miao, W. Pedrycz, W. Ding, and H. Zhang, "Ze-HFS: Zentropy-based uncertainty measure for heterogeneous feature selection and knowledge discovery," *IEEE Trans. Knowl. & Data Eng.*, early access, Jun. 26, 2024.

[48] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013, *arXiv:1301.3781*.

[49] I. Iacobacci and R. Navigli, "LSTMEmbed: Learning word and sense representations from a large semantically annotated corpus with long short-term memories," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics (ACL)*, 2019, pp. 1685–1695.

[50] Q. Xie, J. Huang, P. Du, M. Peng, and J.-Y. Nie, "Inductive topic variational graph auto-encoder for text classification," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Human Lang. Technol. (NAACL)*, 2021, pp. 4218–4227.

[51] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *J. Mach. Learn. Res.,* vol. 3, no. Jan, pp. 993–1022, 2003.

[52] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 1025–1035.

[53] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *Proc. 31st Int. Conf. Int. Conf. Mach. Learn. -Volume 32 (ICML)*, 2014, pp. II–1188.

[54] A. Joulin, É. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," in *Proc. 15th Conf. Eur. Chapter Assoc. Comput. Linguistics (Volume 2: Short Papers)*, 2017, pp. 427–431.

[55] D. Shen et al., "Baseline needs more love: On simple word-embedding-based models and associated pooling mechanisms," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics (Volume 1: Long Papers) (ACL)*, 2018, pp. 440–450.

[56] Y. Zhang, Z. Zhang, D. Miao, and J. Wang, "Three-way enhanced convolutional neural networks for sentence-level sentiment classification," *Inf. Sci.*, vol. 477, pp. 55–64, Mar. 2019.

[57] D. Chen, Y. Lin, W. Li, P. Li, J. Zhou, and X. Sun, "Measuring and relieving the over-smoothing problem for graph neural networks from the topological view," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, vol. 34, no. 4, 2020, pp. 3438–3445.