Wavelet-Enhanced Euclidean-Hyperbolic Graph Convolutional Networks for Text Classification

Anonymous ACL submission

Abstract

Graph convolutional networks (GCNs) have been successfully applied to text classification tasks. However, existing GCN-based methods fail to fully utilize the representational advantages of tree-like structures in hyperbolic space and struggle to capture hierarchical hypernym-hyponym relationships between words. Additionally, text graph construction heavily relies on structural information from fixed corpora. To address these limitations, this study proposes a wavelet-enhanced Euclidean-hyperbolic graph convolutional network (EHGCN) for text classification. The method establishes complementary semantic enhancement across multiple dimensions through frequency-domain analysis and Euclidean-hyperbolic cross-space topology restructuring. The frequency-domain perspective captures text fine-grained features via multi-scale semantic decoupling of word vectors, while the Euclidean-hyperbolic semantic topology constructs cross-space text structures and integrates heterogeneous features from cross-space graph convolution to achieve text representations combining local semantics with hierarchical dependencies. Experiments on five benchmark datasets (R8, R52, MR, Ohsumed, TREC) show that EHGCN achieves a 1.89% average accuracy improvement over mainstream methods. Compared to task-specific LLM-based models (CAPR, COT), EHGCN demonstrates a 7.21% average performance gain.

1 Introduction

Text classification involves analyzing the features of a given text and comparing them with the shared characteristics of known categories, ultimately assigning the text to the most semantically similar class. For example, categorizing news articles into sports, entertainment, or technology (Li et al., 2024; Daud et al., 2023), or classifying customer reviews as positive, negative, or neutral (Taherdoost and Madanchian, 2023). Compared to traditional deep learning methods, graph convolutional neural network (GCN)-based approaches overcome the limitations of modeling local sequential word dependencies (Ai et al., 2025). For instance, conventional methods rely on convolutional or recurrent operations to extract local contextual features (Kim, 2014; Liu et al., 2016), whereas GCNs aggregate multi-hop neighborhood information to simultaneously model fine-grained semantic units (e.g., words, phrases) and structured cross-document relationships (Lei et al., 2021; Yao et al., 2019), significantly enhancing classification robustness in complex scenarios.

However, GCN-based text classification remains highly dependent on the quality of underlying word embeddings. Traditional methods typically employ pre-trained word vector models (e.g., Word2Vec (Mikolov et al., 2013), GloVe (Pennington et al., 2014)) to map vocabulary into low-dimensional Euclidean spaces, capturing shallow semantic associations through statistical co-occurrence. Yet, these embeddings exhibit critical drawbacks: (1) high-frequency noise (e.g., random co-occurring word pairs or spelling errors) can induce semantic drift (Newell et al., 2019; Schnabel et al., 2015); (2) single-scale global statistics fail to disentangle fine-grained word semantics (e.g., polysemy such as "cell" in biology vs. telecommunications) (Levy and Goldberg, 2014). Directly constructing text graphs from such embeddings may introduce noisy edges or weaken hierarchical relationship modeling, ultimately constraining GCN's reasoning capabilities.

Moreover, while traditional GCN-based methods effectively exploit global text information (Wang et al., 2024), their reliance on constructing word-word or word-document graph structures poses scalability challenges for large-scale datasets (Wu et al., 2023). Although techniques like noise filtering and graph centrality reduction can prune redundant edges (Yang et al., 2022b), or sliding

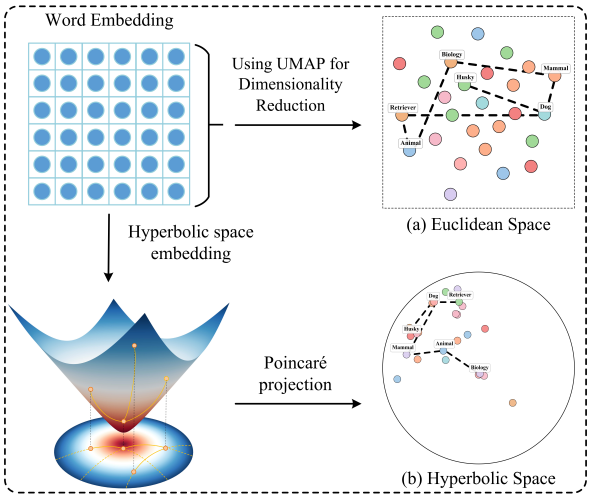


Figure 1: When projecting the hierarchical relationship "Biology-Animal-Mammal-Dog-Retriever-Husky" using Uniform Manifold Approximation and Projection (UMAP) into a low-dimensional plot, proximity to the origin indicates higher hierarchical levels. This demonstrates that the hyperbolic space (b) presents a clearer hierarchical structure compared to the Euclidean space (a).

windows can establish word-level graphs independent of document relationships (Huang et al., 2019), most graph construction methods still operate on fixed corpora. These graphs are typically built from word co-occurrences in known documents, limiting their generalizability to unseen texts.

Additionally, conventional GCN-based methods embed texts in Euclidean space and perform classification via translation, scaling, or nonlinear transformations. However, Euclidean geometry has inherent limitations in representing tree-like or hierarchical data structures (Chami et al., 2019). Natural language data often inherently exhibits hierarchical patterns (Dhingra et al., 2018; Nickel and Kiela, 2017), as illustrated in Figure 1, which contrasts hyperbolic and Euclidean embeddings of word hierarchies. Consequently, Euclidean space-based methods may inadequately capture such structural information. Fortunately, hyperbolic space naturally embeds hierarchical relationships, and hyperbolic embeddings have proven effective for text classification (Zhu et al.; Xu et al., 2022). Notably, Chami et al. (Ganea et al., 2018) first integrated hyperbolic embeddings into GCNs, achieving promising results in node classification and link prediction. Motivated by these advancements, we propose integrating hyperbolic space into GCN-based text classification.

To address these challenges, we propose a Wavelet-enhanced Euclidean-Hyperbolic Graph Convolutional Network (EHGCN). First, to mitigate noise sensitivity and semantic ambiguity in traditional word embeddings, we project raw word vectors into the frequency domain. Using wavelet basis functions, we decompose embeddings into multi-scale subbands, separating high-frequency noise from low-frequency semantic components, thereby enhancing noise robustness and multi-level semantic representation. Second, we introduce a novel graph sparsification method that constructs dual semantic graphs in both Euclidean and hyperbolic spaces, pruning weakly related edges to improve information propagation efficiency. Finally, we fuse heterogeneous features from both spaces to comprehensively model global similarities and hierarchical dependencies.

The main contributions of this work are threefold:

(1) Hierarchical decomposition of word vectors in a document using wavelet bases, suppressing noise interference and fusing multi-scale information, realizes deep parsing of text semantics in order to be able to further extract detailed information in the text.

(2) Constructing text graph structure through the similarity of heterogeneous space between words, replacing the traditional complex approach of multiple matrix union. Free from the dependence of traditional methods on fixed corpora, it supports online document prediction and reduces memory consumption. To the best of our knowledge, this is the first graph construction method that does not rely on the fixed corpus level at all.

(3) Embedding Euclidean and hyperbolic spaces into our text categorization model allows capturing global similarities and hierarchical relationships between words more comprehensively, compensating for the limitations of single-space representations.

2 Related Work

Recent advances in multi-scale feature modeling and graph-structured learning have opened new perspectives for text classification. Wavelet analysis, with its multi-scale signal processing capabilities, has been widely applied to enhance textual feature representations. For instance, (Chamorro-Padial and Rodríguez-Sánchez, 2020) proposed dimensionality reduction of document-term matrices via discrete wavelet transforms, while diffusion wavelet techniques were adopted to con-

struct multi-scale word co-occurrence graphs for addressing short-text sparsity issues (Jain and Mahadeokar, 2014). Additionally, hybrid models based on Haar wavelets were designed to map term frequency matrices onto hierarchical semantic features for hypernymy-hyponymy relationship modeling (Dönmez and Aslan, 2021). However, existing wavelet-based methods primarily rely on surface-level statistical features and lack effective integration with deep semantic embeddings.

In the domain of graph convolutional networks (GCNs), research efforts have focused on optimizing graph construction strategies. Spatial-domain GCNs, which update node representations through neighbor aggregation (Veličković et al., 2019), demonstrate superior efficiency compared to spectral-domain approaches (Kipf and Welling, 2016). To capture non-contiguous semantic relationships, heterogeneous graphs have been built using TF-IDF, PPMI (Yao et al., 2019), and document sliding windows (Li et al., 2023), or by integrating multi-perspective features (e.g., semantic, syntactic, and contextual) (Liu et al., 2020). Despite improved global modeling capabilities, these methods suffer from high computational overhead due to complex multi-metric graph construction, limiting their scalability for large-scale text applications.

Hyperbolic space, with its exponential volume growth property, provides inherent advantages for hierarchical text modeling. Compared to flat Euclidean embeddings (Mikolov et al., 2013), hyperbolic representations encode tree-like semantic structures through geodesic distances (Zhu et al.), validated in hierarchical attention networks (Zhang and Gao, 2021) and hyperbolic graph convolutions (Chami et al., 2019). Nevertheless, existing hyperbolic models predominantly treat semantic hierarchies in isolation, with insufficient exploration of joint optimization between hyperbolic and Euclidean space features.

The key distinction of this work lies in our proposed framework. It integrates text feature enhancement, graph structure optimization, and heterogeneous space fusion. Through collaborative modeling of heterogeneous geometric features, our framework unifies the representation of hierarchical semantics and local contextual dependencies in text analysis.

3 Model Architecture

This section introduces the core methodology of EHGCN. The model architecture is illustrated in Figure 2, and the fundamental concepts used in this section are detailed in Appendix C.

3.1 Text feature enhancement based on multilevel discrete wavelet decomposition

We denote the given document collection as $D = \{S_1, S_2, \ldots, S_n\}$, where S_i represents an individual sentence. Each sentence S_i consists of multiple words:$S_i = [w_1^i, w_2^i, \cdots, w_k^i]$. For simplicity, we remove the sentence-specific indices of words and refer to the set of unique words in document D as the document word corpus $W = [w_1, w_2 \ldots, w_m]$, where m denotes the number of distinct words in D. Assuming each word embedding vector has a dimensionality of $w_i \in \mathbb{R}^d$(here, initialized using 300-dimensional GloVe embeddings[1]), the raw feature matrix of the corpus can be expressed as:

$$E = [w_1, w_2, \ldots, w_m]^T \in \mathbb{R}^{m \times d} \tag{1}$$

For each word vector, we perform a j-level wavelet decomposition using the bior-3.5(Cohen et al., 1992) wavelet basis, with wavelet coefficients given by:

$$\begin{aligned} l \approx [&0.0469, -0.1407, 0.1094, 0.6029, \\ &0.2669, -0.0782, -0.0169, 0.0268] \\ h \approx [&-0.0268, -0.0169, 0.0782, -0.2669, \\ &0.6029, -0.1094, -0.1407, -0.0469] \end{aligned} \tag{2}$$

Then, the decomposition for the first layer and the j-th layer is:

$$\begin{cases} a_l^1 = \text{downsample}(w_i * l) \in \mathbb{R}^{d/2} \\ a_h^1 = \text{downsample}(w_i * h) \in \mathbb{R}^{d/2} \end{cases} \tag{3}$$

$$\begin{cases} a_l^j = \text{downsample}(a_{j-1} * l) \in \mathbb{R}^{d/2^j} \\ a_h^j = \text{downsample}(a_{j-1} * h) \in \mathbb{R}^{d/2^j} \end{cases} \tag{4}$$

The final enhanced text features are denoted as $E' = [w_1', w_2', \ldots, w_m']^T \in \mathbb{R}^{m \times d'}$, where $d' = d(1 - 1/2^j)$. Figure 3 compares the energy distribution between the original and wavelet-enhanced features (three-level decomposition). The enhanced features exhibit three key characteristics: (1) increased low-frequency energy (A) strengthens global semantic representation of topics and

[1] http://nlp.stanford.edu/data/glove.6B.zip

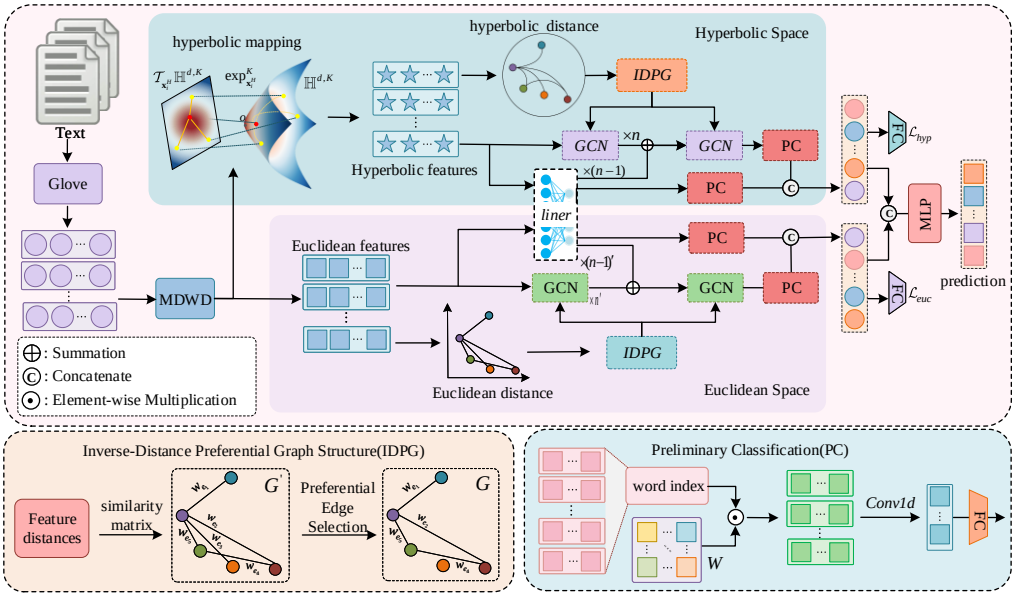


Figure 2: EHGCN Model Framework Diagram

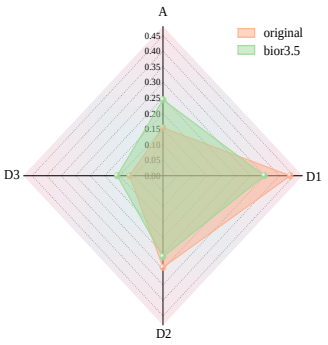


Figure 3: Comparison of energy distribution

paragraph logic, (2) reduced high-frequency energy (D1) suppresses short-term noise interference (NS and Surendran, 2019), and (3) elevated mid-low-frequency energy (D3) improves context coherence and long-range dependency modeling. This multi-scale energy allocation mechanism consequently provides more discriminative multi-level feature representations for complex text classification tasks.

3.2 Inverse-Distance Preferential Graph Structure

The construction of common text graphs primarily involves three matrices: Word-word matrix $A_1 \in \mathbb{R}^{m\times m}$, Word-sentence matrix $A_2 \in \mathbb{R}^{m\times n}$, Sentence-sentence matrix $A_3 \in \mathbb{R}^{n\times n}$, where m and n denote the number of unique words and sentences in the corpus. For A_1 , word-edge relationships are typically fixed, often computed using point-wise mutual information (PMI) to determine weights between words (Yao et al., 2019). However, this approach requires recalculating A_1 when new corpora are added. A_2 generally uses TF-IDF (term frequency-inverse document frequency) to establish edge weights (Li et al., 2023; Gao et al., 2024), but storing A_2 demands significant memory resources for large corpora. A_3 is usually an identity matrix or built using sentence similarity, yet it relies on the target text for matrix construction, limiting online prediction capability.

To address these issues, IDPG requires only the A_1 matrix. The edge weights between words are computed using the inverse of distance-based similarity between enhanced word features E' , defined as:

$$a_{ij} = \begin{cases} 1/\left(\sqrt{\sum_{k=1}^{d}(x_{ik}-x_{jk})^2+\varepsilon}\right) & \text{if } i \neq j \\ \max(a_{ij}) & \text{if } i = j \end{cases} \quad (5)$$

$$A_h = \begin{cases} 1/\operatorname{arccosh}\left(\langle w_i', w_j'\rangle_L + \varepsilon\right) & \text{if } i \neq j \\ \max\limits_{j\neq i} a_{ij} & \text{if } i = j \end{cases} \quad (6)$$

Here, A_d and A_h denote the similarity weight matrices between all words in Euclidean and hyperbolic spaces, respectively, where $x_{i,k}$ represents the k-th dimension parameter of the i-th word. After normalization via Equation 19, the K neighbors with the highest weights for each node are selected to form the matrices A_d' and A_h'. This single-matrix sparse storage reduces memory overhead and scales efficiently for large corpora. Additionally, the inverse-distance-based selection miti-

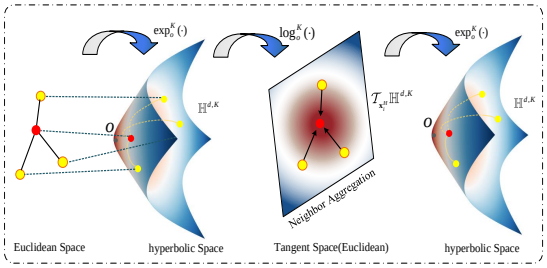


Figure 4: Schematic Diagram of Hyperbolic Neighborhood Aggregation

gates noise from weakly correlated neighbors during message propagation.

3.3 Hyperbolic Neighborhood Aggregation

Hyperbolic space is a smooth Riemannian manifold with constant negative curvature (Benedetti and Petronio, 1992). This paper employs the Lorentz model due to its superior numerical stability (Nickel and Kiela, 2018).Neighborhood aggregation is a critical step in GCNs, as shown in Figure 4, which captures neighborhood structure and features. Consider a node x_i on matrix A_h that aggregates neighbors $x_i \in N(i)$ with learnable weight parameters $w_h \in \mathbb{R}^{m\times m}$. The neighborhood aggregation for the node is formulated as:

$$(x_i^H)' = \exp_{x_i^H}^K \left(\sigma^K \left(\sum_{j\in N(i)} A'_h[i,j] \cdot \log_{x_i^h}^K (x_j^H) \cdot w_{ij}^H \right) \right) \quad (7)$$

where $\log_o^K(\cdot)$ and $\exp_o^K(\cdot)$ denote the logarithmic map and exponential map, respectively. The term $\sum_{j\in N(i)} A'_h[i,j] \cdot \log_{x_i^h}^K (x_j^H) \cdot w_{ij}^H$ represents the aggregation of neighboring features for node x_i via mean pooling. Notably, this aggregation operation is performed in the local tangent space of the central node x_i^H. σ^K denotes the non-linear activation function in hyperbolic space:

$$\sigma^K(x^H) = \exp_o^K \left(\sigma \left(\log_o^K (x^H) \right) \right) \quad (8)$$

3.4 Heterogeneous Space Feature Fusion and Classification Prediction

Heterogeneous space feature fusion integrates hyperbolic space (non-Euclidean geometry) and Euclidean space features for text analysis, including a preliminary classification module and a heterogeneous feature fusion mechanism.

Preliminary Classification: Let the pretrained word vector representation be $F \in \mathbb{R}^{n\times d_0}$(defaulting to Euclidean space or tangent space). The aggregated sentence-level representation is defined as: $S = [F_1^i, F_2^i, \ldots, F_m^i,] \in \mathbb{R}^{m\times d_0}$, where m denotes the number of words in the current sentence.

$$Y = \text{Conv1D}(\sigma(S \odot W), K) \cdot M \quad (9)$$

where $W \in \mathbb{R}^{m\times d_0}$, and $\odot$ denotes the element-wise multiplication between S and the learnable matrix W . After applying the activation function σ and 1D convolution ($K = 1$), a d_0-dimensional feature representation is formed. This is further processed by $M \in \mathbb{R}^{d_0\times m}$, where c is the number of classes and b is the bias vector.

Feature Fusion and Classification Prediction: First, the Euclidean features Y_1, Y_2 and hyperbolic hierarchical features Y_3, Y_4 from the preliminary classification are concatenated along the feature dimension as $Y^j = [Y_1^j, Y_2^j, Y_3^j, Y_4^j,] \in \mathbb{R}^{4m}$, where j denotes the j-th text sentence. The concatenated features are then fed into an MLP to produce the final prediction$Z^j \in \mathbb{R}^c$.

4 Experiment

4.1 Dataset

We evaluate our method on five datasets from different domains and sizes. Brief data statistics are listed below (see more details in Appendix A). (1) R8[2]: Classifies documents from Reuters news wires into 8 categories. (2) R52: Similar to R8 but divided into 52 categories. (3) Ohsumed[3](Ohs): Medical literature corpus from the MEDLINE database. (4)MR(Tang et al., 2015): Single-sentence review dataset for binary sentiment classification. (5)TREC(Li and Roth, 2002):TREC question classification dataset containing 6 categories.

4.2 Baseline and Experimental Settings

We compare the EHGCN model with three categories of baseline models: sequence-based deep learning models, word embedding-based models, and graph-based representation learning models, as detailed below:

Sequence-based Deep Learning Models: CNN (Kim, 2014): Uses convolutional kernels to extract local text features. LSTM (Liu et al., 2016): Models sequential dependencies via Long Short-Term Memory (LSTM) units. Bi-LSTM (Huang et al.,

[2] http://www.daviddlewis.com/resources/testcollections/reuters21578/

[3] http://disi.unitn.it/moschitti/corpora.htm

2015): Captures bidirectional contextual information.

Word Embedding-based Models: PV-DBOW (Le and Mikolov, 2014): Predicts random words in documents using paragraph vectors while ignoring word order. fastText (Joulin et al., 2016): Leverages bag-of-words and subword embeddings. LEAM (Wang et al., 2018): Incorporates label embeddings with attention mechanisms.

Graph-based Representation Learning Models: TextGCN (Yao et al., 2019): Constructs a document-word heterogeneous graph and jointly learns document/word representations through graph convolutions. HyperGAT (Ding et al., 2020): Builds hyper-edges and iteratively updates node and edge features in graph convolutions. TensorGCN (Liu et al., 2020): Merges additional node-edge weights for semantic, syntactic, and sequential relations. CGA2TC (Yang et al., 2022b): A graph-based text classification framework. Text-MGNN (Gu et al., 2023): Models multi-granularity relationships on a tripartite graph (word-document-topic). MHGAT (Jin et al., 2024): Captures word positions and multi-element information for document classification.

The model initializes word vectors with 300D GloVe embeddings, enhances features via 3-level wavelet decomposition, and sets hyperbolic curvature to 1. The inverse-distance neighbor selection retains top-3 weights. Training uses a batch size of 32 and Adam optimizer (learning rate 0.001) for up to 100 epochs, with early stopping triggered if validation loss plateaus for 10 epochs. Reported results are averaged over 10 independent runs.

4.3 Text Classification Performance

We compare EHGCN with the baseline models, and the experimental performance is summarized in Table 1. The table shows that EHGCN achieves superior performance on all five datasets. Specifically, we draw the following conclusions:

Graph-based methods generally outperform bag-of-words or sequence models (e.g., LSTM, FastText) as they integrate corpus-level co-occurrence information through global relational graphs, effectively modeling long-range semantic dependencies. These methods exhibit stronger robustness in data-sparse or structurally complex tasks.

The EHGCN model further optimizes the limitations of traditional graph-based methods: it enhances feature granularity through multi-level wavelet decomposition, models hierarchical relationships via hyperbolic space embeddings, and improves generalization capability through dynamic graph construction (IDPG), achieving significant advantages on the MR and TREC datasets. Specifically, the MR test set contains numerous unseen words, and EHGCN's independence from fixed corpus characteristics strengthens its new document prediction capability. TREC's hierarchical intent relationships are effectively captured through tree-like semantic encoding in hyperbolic space. However, on the Ohsumed dataset, MHGAT achieves higher accuracy than EHGCN, This reflecting EHGCN's shortcomings in positional information modeling and domain-specific structure representation.

Model	R8	R52	Ohs	MR	TREC
CNN	95.17	87.59	58.44	77.75	93.62
LSTM	96.09	90.48	41.13	77.33	93.01
Bi-LSTM	96.31	90.54	49.27	77.68	93.32
PV-DBOW	85.87	78.29	46.65	61.09	80.36
fastText	96.13	92.81	57.70	75.14	91.29
LEAM	93.31	91.84	58.58	76.95	89.21
TextGCN	97.07	93.56	68.36	76.74	91.40
Hyper-GAT	97.97	94.98	69.90	78.32	93.55
TensorGCN	98.04	95.05	70.11	77.91	-
CGA2TC	97.76	94.47	70.62	77.80	-
Text-MGNN	97.39	94.20	70.00	77.46	-
MHGAT	97.65	94.78	**72.88**	78.09	-
EHGCN	**98.25**	**95.67**	69.27	**84.95**	**99.20**

Table 1: Test accuracy (%) of different models on five different datasets. Bold numbers indicate the best-performing models.

In summary, EHGCN validates the effectiveness of its improvement strategies across multiple tasks, particularly achieving performance breakthroughs on datasets with high baseline accuracy (R8 and R52). Meanwhile, MHGAT's advantages in processing long domain-specific texts offer complementary insights for future research.

4.4 Ablation Study

We study the impact of various improved components in the model on text classification tasks. "W/O MDWD" denotes removing the multi-level wavelet decomposition-based text feature enhancement module. "W/O Pre-Words" indicates eliminating the neighbor preference selection strategy in the graph structure. "PMI (Pref-Words)" represents constructing the text graph using the PMI method with neighbor preference selection. "PMI

Model	R8	R52	MR	TREC
EHGCN(ours)	**98.45**	**95.67**	**84.95**	**99.20**
w/o MDWD	97.57	94.31	83.78	98.60
w/o Pref-Words	97.72	94.74	77.41	98.20
PMI(Pref-Words)	97.53	94.19	75.97	97.80
PMI(NonPref-Words)	97.40	93.30	75.91	76.22
w/o Hyperbolic	97.85	95.40	75.75	98.80
w/o Euclidean	97.35	94.98	75.66	99.00

Table 2: Impact of model on classification accuracy (%).

(NonPref-Words)" refers to building the text graph via the PMI method without neighbor preference selection. "W/O Hyperbolic" signifies removing the hyperbolic space feature representation, while "W/O Euclidean" means discarding the Euclidean space feature representation. As shown in Table2, these components demonstrate diverse effects, and we observe that removing or modifying any component severely negatively impacts the model.

4.5 Hyperparameter Experiment

During the experimental process, we identify three critical hyperparameters that significantly influence predictive performance. These include the number of neighbor node information K, the type of wavelet basis, and the wavelet decomposition level J. In this section, we analyze the impact of varying these hyperparameters on the overall model performance.

4.5.1 The impact of neighbor node K

Experiments on R8 and MR datasets evaluate the impact of neighborhood sampling range (K-value) on model performance. As shown in Figure 5: (1) Accuracy exhibits a positive correlation with K, peaking at $K \approx 15$; (2) Beyond this threshold, accuracy gradually declines, approaching the full-connection baseline. This suggests that insufficient neighbors limit semantic modeling, while excessive neighbors introduce noise from redundant weak edges. Optimal neighborhood filtering balances semantic integrity and noise robustness. Additionally, reduced edge counts lower memory consumption, demonstrating the model's efficiency.

4.5.2 The impact of wavelet basis and decomposition levels

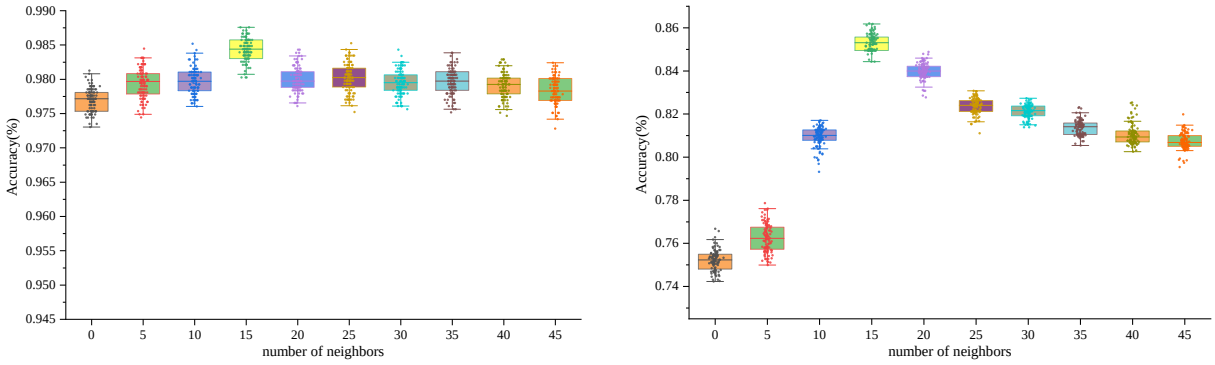
Experiments on the R8 and MR datasets evaluate the impact of different wavelet bases (haar, db4, sym8, bior3.5) (Zhang, 2019) and three decomposition levels (2, 3, 4). For decomposition level selection: when the decomposition level is set to 1, the signal splits only into low-frequency (A) and high-frequency (D1) components, failing to capture multi-scale semantic features; levels (≥ 5) lead to exponential expansion of feature dimensions and amplified high-frequency noise, causing information redundancy (Jr et al., 2018). The experimental results (Figure 6) reveal: (1) As shown in Figures 6a, 6b, and 6c, the bior3.5 basis achieves the highest proportion of low-frequency component A, providing stronger global semantic representation; (2) Figures 6d and 6e demonstrate that all wavelet bases enhance classification performance; (3) Optimal performance is achieved with 3 decomposition levels.

Model	R8	R52	MR
EHGCN(ours)	**98.45**	**95.67**	84.95
CARP	75.10	73.05	83.94
COT	90.48	91.24	**89.37**

Table 3: Test accuracy of large language models and EHGCN on three different datasets

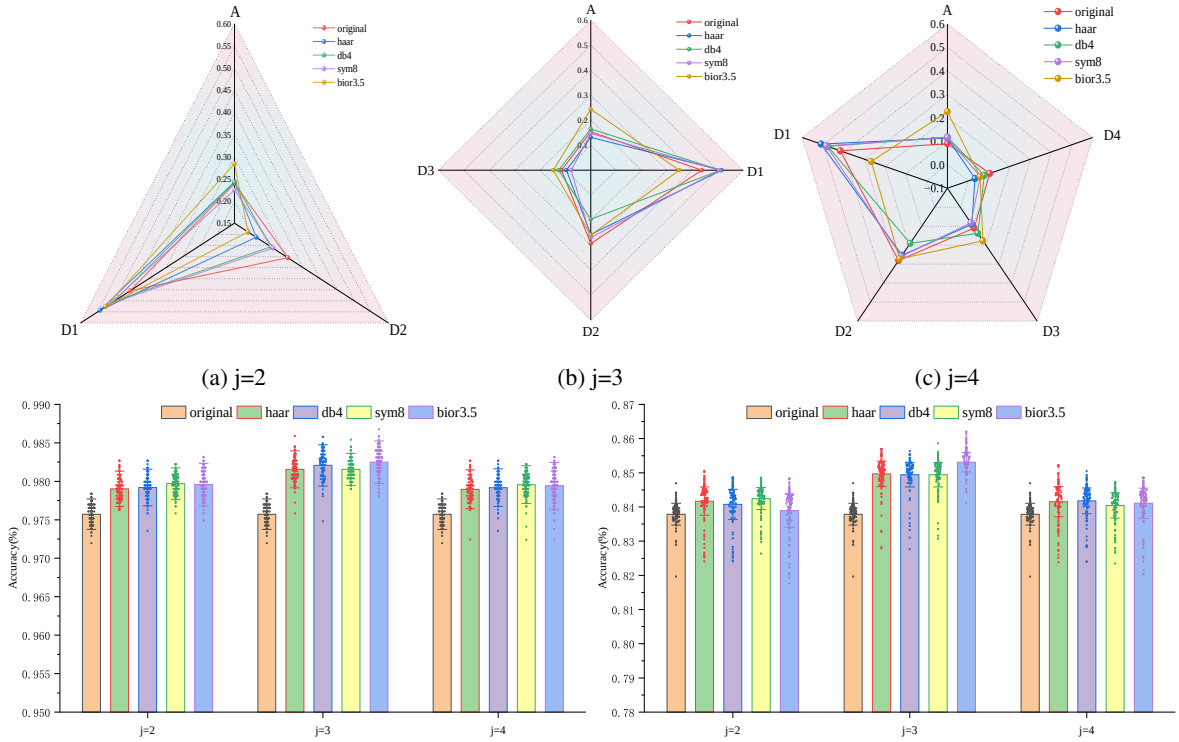
4.6 Comparison with large language models

EHGCN is compared with CARP (Sun et al., 2023) (using LLaMA2-7B (Touvron et al., 2023) as the backbone model, with training results sourced from (Liu et al., 2024)) and COT (Kojima et al., 2022) (using GPT-3 (Ouyang et al., 2022) as the backbone network, with training results from (Sun et al., 2023)) on the R8 and R52 datasets. As shown in Table 3, EHGCN outperforms CARP by 31.08% and COT by 8.80% on R8, while achieving a 22.62% and 4.43% lead over CARP and COT, respectively, on R52. These results demonstrate that the graph convolutional architecture, which models document topological relationships, exhibits significant advantages in specific text classification tasks. However, EHGCN underperforms COT (89.37%) on the MR sentiment dataset (84.95%), which may be related to the graph structure's limited ability to capture semantics in short texts. Future research may explore heterogeneous architectures that integrate EHGCN's lightweight design with the semantic comprehension capabilities of large language models (LLMs).



(a) R8 (b) MR

Figure 5: The impact of neighbor count in graphs on classification performance. Description: The results of each edge are based on 100 experiments, visualized through a combined method of box plots and scatter elements to present the accuracy distribution across different datasets and numbers of neighbors.



(a) j=2 (b) j=3 (c) j=4

(d) R8 (e) MR

Figure 6: Comparative effectiveness figure of different wavelet bases and decomposition levels. Note: Each result in (d) and (e) is averaged over 100 experimental trials.

5 Conclusion

This paper proposes a novel text classification model, EHGCN. The model enhances semantic representation through multi-scale feature decomposition and constructs global semantic association graphs in Euclidean and hyperbolic spaces to improve prediction capability for new documents. Furthermore, it integrates heterogeneous spatial features to fuse contextual and hierarchical semantics. Experiments on benchmark datasets validate the model's effectiveness and superiority.

Limitations

Dynamic modeling for domain adaptation: The current graph structure construction relies on general corpus environments and does not explicitly incorporate domain knowledge (e.g., on the Ohsumed medical dataset, classification performance remains relatively insufficient due to missing hierarchical relationships among specialized terms). Additionally, the dynamic optimization mechanism does not consider semantic sparsity commonly observed in low-resource languages,

which may introduce cross-domain bias.

Hyperbolic curvature configuration: Although the current model adopts a fixed hyperbolic curvature value ($C = 1$) (Yang et al., 2022a) to align with mainstream methods, it does not validate the impact of dynamic curvature adjustment on modeling heterogeneous semantic hierarchies (Fu et al., 2021).

Acknowledgments

References

Wei Ai, Jianbin Li, Ze Wang, Yingying Wei, Tao Meng, and Keqin Li. 2025. Contrastive multi-graph learning with neighbor hierarchical sifting for semi-supervised text classification. *Expert Systems with Applications*, 266. 25.

Riccardo Benedetti and Carlo Petronio. 1992. *Lectures on hyperbolic geometry*. Springer-Verlag, Heidelberg, Berlin.

James W Cannon, William J Floyd, Richard Kenyon, and Walter R Parry. 1997. Hyperbolic geometry. *Flavors of geometry*, 31. 2.

Ines Chami, Zhitao Ying, Christopher Ré, and Jure Leskovec. 2019. Hyperbolic graph convolutional neural networks. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pages 4868–4879.

Jorge Chamorro-Padial and Rosa Rodríguez-Sánchez. 2020. Text categorisation through dimensionality reduction using wavelet transform. *Journal of Information & Knowledge Management*, 19. 04.

Albert Cohen, Ingrid Daubechies, and J C Feauveau. 1992. Biorthogonal bases of compactly supported wavelets. *Communications on pure and applied mathematics*, 45. 5.

Shahzada Daud, Muti Ullah, Amjad Rehman, Tanzila Saba, Robertas Damaševičius, and Abdul Sattar. 2023. Topic classification of online news articles using optimized machine learning models. *Computers*, 12. 01.

Bhuwan Dhingra, Christopher Shallue, Mohammad Norouzi, Andrew Dai, and George Dahl. 2018. Embedding text in hyperbolic spaces. In *In Proceedings of the Twelfth Workshop on Graph-Based Methods for Natural Language Processing*, pages 59–69.

Kaize Ding, Jianling Wang, Jundong Li, Dingcheng Li, and Huan Liu. 2020. Be more with less: Hypergraph attention networks for inductive text classification. In *In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 4927–4936.

İlknur Dönmez and Zafer Aslan. 2021. Document sentiment classification using hybrid wavelet methodologies. *Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi*, 36. 2.

Xingcheng Fu, Jianxin Li, Jia Wu, Qingyun Sun, Cheng Ji, and Senzhang Wang. 2021. Ace-hgnn: Adaptive curvature exploration hyperbolic graph neural network. In *IEEE international conference on data mining*, pages 111–120.

Octavian-Eugen Ganea, Gary Bécigneul, and Thomas Hofmann. 2018. Hyperbolic neural networks. In *Advances in neural information processing systems 31*, pages 5345–5355.

Yue Gao, Xiangling Fu, Xien Liu, and Ji Wu. 2024. Deeply integrating unsupervised semantics and syntax into heterogeneous graphs for inductive text classification. *Complex & Intelligent Systems*, 10.

Yongchun Gu, Yi Wang, Heng-Ru Zhang, Jiao Wu, and Xingquan Gu. 2023. Enhancing text classification by graph neural networks with multi-granular topic-aware graph. *IEEE Access*, 11.

Lianzhe Huang, Dehong Ma, Sujian Li, Xiaodong Zhang, and Houfeng Wang. 2019. Text level graph neural network for text classification. In *In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 3444–3450.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging. arXiv:1508.01991.

Vidit Jain and Jay Mahadeokar. 2014. Short-text representation using diffusion wavelets. In *In Proceedings of the 23rd International Conference on World Wide Web*, pages 301–302.

Yilun Jin, Wei Yin, Haoseng Wang, and Fang He. 2024. Capturing word positions does help: A multi-element hypergraph gated attention network for document classification. *Expert Systems with Applications*, 251.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification. In *In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 427–431.

Sylvio Barbon Jr, Gabriel Fillipe Centini Campos, Gabriel Marques Tavares, Rodrigoaugusto Igawa, Mario L Jr, and Rodrigo Capobianco Guido. 2018. Detection of human, legitimate bot, and malicious bot in online social networks based on wavelets. In *ACM Transactions on Multimedia Computing, Communications, and Applications*, pages 1–17.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *In Proceedings of the 2014 conference on empirical methods in natural language processing(EMNLP)*, pages 1746–1751.

Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. arXiv:1609.02907.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Advances in neural information processing systems 35*, pages 22199–22213.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on International Conference on Machine Learning*, pages 1188–1196.

Fangyuan Lei, Xun Liu, Zhengming Li, Qingyun Dai, and Senhong. Wang. 2021. Multihop neighbor information fusion graph convolutional network for text classification. *Mathematical Problems in Engineering*, 1.

Omer Levy and Yoav Goldberg. 2014. Neural word embedding as implicit matrix factorization. In *Proceedings of the 28th International Conference on Neural Information Processing Systems*, pages 2177–2185.

Xianghua Li, Xinyu Wu, Zheng Luo, Zhanwei Du, Zhen Wang, and Chao Gao. 2023. Integration of global and local information for text classification. *Neural Computing and Applications*, 35.

Xiaoyu Li, Lingbo Han, and Zhengfeng Jiang. 2024. Deep learning-based news text classification algorithm research. *IEEE Access*, 12.

Xin Li and Dan Roth. 2002. Learning question classifiers. In *In COLING 2002: The 19th International Conference on Computational Linguistics*.

Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2016. Recurrent neural network for text classification with multi-task learning. In *In Proceedings of the twenty-fifth international joint conference on artificial intelligence*, pages 2873–2879.

Xien Liu, Xinxin You, Xiao Zhang, Ji Wu, and Ping Lv. 2020. Tensor graph convolutional networks for text classification. In *In Proceedings of the AAAI conference on artificial intelligence*, pages 8409–8416.

Yonghao Liu, Lan Huang, Fausto Giunchiglia, Xiaoyue Feng, and Renchu Guan. 2024. Improved graph contrastive learning for short text classification. In *In Proceedings of the AAAI conference on artificial intelligence*, pages 18716–18724.

Stephane G Mallat. 1989. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE transactions on pattern analysis and machine intelligence*, 11. 7.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *In Proceedings of the 27th International Conference on Neural Information Processing Systems*, pages 3111–3119.

Edward Newell, Kian Kenyon-Dean, and Jackie Chi Kit Cheung. 2019. Deconstructing and reconstructing word embedding algorithms. arXiv:1911.13280.

Maximilian Nickel and Douwe Kiela. 2017. Poincaré embeddings for learning hierarchical representations. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6341–6350.

Maximillian Nickel and Douwe Kiela. 2018. Learning continuous hierarchies in the lorentz model of hyperbolic geometry. In *Proceedings of the 35th International Conference on Machine Learning, PMLR*, pages 3779–3788.

Ameena NS and Subu Surendran. 2019. Identification of malicious bots in twitter using wavelets. In *In Proceedings of International Conference on Recent Trends in Computing, Communication Networking Technologies*, pages 1–7.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, pages 27730–27744.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Tobias Schnabel, Igor Labutov, David Mimno, and Thorsten Joachims. 2015. Evaluation methods for unsupervised word embeddings. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 298–307.

Xiaofei Sun, Xiaoya Li, Jiwei Li, Fei Wu, Shangwei Guo, Tianwei Zhang, and Guoyin Wang. 2023. Text classification via large language models. In *In Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 8990–9005.

Hamed Taherdoost and Madanchian Madanchian. 2023. Artificial intelligence and sentiment analysis: A review in competitive research. *Computers*, 12(2). 37.

Jian Tang, Meng Qu, and Qiaozhu Mei. 2015. Pte: Predictive text embedding through large-scale heterogeneous text networks. In *In Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1165–1174.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, rajjwal PBhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023. Llama 2: Open foundation and fine-tuned chat models. arXiv:2307.09288.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9. 11.

Petar Veličković, William Fedus, William L Hamilton, Pietro Liò , Yoshua Bengio, and R Devon Hjelm. 2019. Deep graph infomax. *International Conference on Learning Representations (ICLR)*.

Guoyin Wang, Chunyuan Li, Wenlin Wang, Yizhe Zhang, Dinghan Shen, Xinyuan Zhang, Ricardo Henao, and Lawrence Caarin. 2018. Joint embedding of words and labels for text classification. In *In Proceedings of the 56th annual meeting of the association for computational linguistics*, pages 2321–2331.

Kunze Wang, Yihao Ding, and Soyeon Caren Han. 2024. Graph neural networks for text classification: A survey. *Artificial Intelligence Review*, 57. 190.

Tiandeng Wu, Qijiong Liu, Yi Cao, Yao Huang, Xiaoming Wu, and Jiandong Ding. 2023. Continual graph convolutional network for text classification. In *In Proceedings of the AAAI conference on artificial intelligence*, pages 13754–13762.

Yishi Xu, Dongsheng Wang, Bo Chen, Ruiying Lu, Zhibin Duan, and Mingyuan Zhou. 2022. Hyperminer: Topic taxonomy mining with hyperbolic embedding. In *Advances in Neural Information Processing Systems 35*, pages 31557–31570.

Menglin Yang, Min Zhou, Zhihao Li, Jiahong Liu, Lujia Pan, Hui Xiong, and Irwin King. 2022a. Hyperbolic graph neural networks: A review of methods and applications. arXiv:2202.13852.

Yintao Yang, Rui Miao, Yili Wang, and Xin Wang. 2022b. Contrastive graph convolutional networks with adaptive augmentation for text classification. *Information Processing & Management*, 59. 4.

Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Graph convolutional networks for text classification. In *In Proceedings of the AAAI conference on artificial intelligence*, pages 7370–7377.

Chengkun Zhang and Junbin Gao. 2021. Hype-han: Hyperbolic hierarchical attention network for semantic embedding. In *In Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pages 3990–3996.

Dengsheng Zhang. 2019. *Wavelet Transform*. Computer Science, Cham, Switzerland.

Yudong Zhu, Di Zhou, Jinghui Xiao, Xin Jiang, Xiao Chen, and Qun Liu. Hypertext: Endowing fasttext with hyperbolic geometry. In *In Findings of the Association for Computational Linguistics: EMNLP 2020, pages=1166-1171, year=2020,*.

A Details on Datasets and Pre-processing

The experimental preprocessing pipeline follows the steps in (Yao et al., 2019), including text cleaning, tokenization, stop word removal, and filtering of low-frequency words that occur fewer than five times (except for the MR dataset). The summarized statistics of the preprocessed datasets are shown in Table 4.

B Method Supplementation and Visualization

B.1 Training Objective

The total loss function $\mathcal{L}_{total}$ of the EHGCN model comprises three components: the Euclidean space classification loss $\mathcal{L}_{euc}$, the hyperbolic space classification loss $\mathcal{L}_{hyp}$, and the final prediction loss $\mathcal{L}_{final}$ for Z^j. irst, the features from both hyperbolic and Euclidean spaces are passed through fully connected layers and a softmax function to predict class labels. The cross-entropy loss is used as the classification loss. Taking the Euclidean space feature classification as an example, the loss is defined as:

$$\hat{y}_j = \text{softmax}\left([Y_3^j, Y_4^j] \cdot W_{fc} + b\right) \tag{10}$$

$$\mathcal{L}_{euc} = -\sum_j y_j \log(\hat{y}_j) \tag{11}$$

where W_{fc} and b denote the weight and bias parameters. The losses $\mathcal{L}_{final}$ and $\mathcal{L}_{total}$ are defined as follows:

$$\mathcal{L}_{final} = -\frac{1}{b}\sum_{j=1}^{b} \log\left(\frac{\exp(-Z_{l_j}^j)}{\sum_{i=1}^{c}\exp(-Z_{l_j}^j)}\right) \quad ,(k = 1, 2, \ldots, c) \tag{12}$$

$$\mathcal{L}_{total} = \mathcal{L}_{final} + \mathcal{L}_{euc} + \mathcal{L}_{hyp} \tag{13}$$

where b denotes the batch size, and l_j represents the ground-truth class label of the j -th text sentence in the batch. The model parameters are optimized via the final total loss $\mathcal{L}_{total}$.

B.2 Pseudo-code for Batch Learning

In summary, the model can be summarized as Algorithm 1.

Dataset	#Docs	#Train	#Test	#Words	#Classes	Average Length
R8	7674	5485	2189	7688	8	65.72
R52	9100	6532	2568	8892	52	69.82
Ohsumed	7400	3357	4043	14157	23	135.80
MR	10662	7108	3554	18764	2	20.39
TREC	5952	5452	500	8783	6	11.05

Table 4: Dataset Details

Algorithm 1 Wavelet-Enhanced EuclideanHyperbolic Graph Convolutional Networks for Text Classification

Input: Corpus W and dataset D

1.$E_d = (W \leftarrow \text{glove}) \leftarrow$ MDWD, $E_h = E_d \leftarrow$ Hyperbolicspace, $A'_d \leftarrow$ IDPG, $A'_h \leftarrow$ IDPG.

2.For $epoch \leftarrow 1, 2, \ldots, e$ do

3. $h_1 = \text{gcn}_h^1(A'_h, E'_h), e_1 = \text{gcn}_e^1(A'_d, E'_d)$, $h_2 = h_1 \oplus \text{liner}(E_h), e_2 = e_1 \oplus \text{liner}(E_d)$.

4. $e_3 = \text{gcn}_d^2\left((\text{liner}(E_d) \oplus e_1), A'_d\right)$, $e_4 = \text{gcn}_d^3(e_3, A'_d)$.

5. $Y_1 = liner(E_d) \leftarrow (PC), Y_2 = e_4 \leftarrow PC$.

6. $Y_3 = \text{liner}(E_h) \leftarrow \text{PC}$, $h_3 = \text{gcn}^2(A'_h, (h_2 \oplus h_1)), Y_4 = h_3 \leftarrow \text{PC}$.

7. $Z = [Y_1.Y_2, Y_3, Y_4]$,$MLP(Z) \leftarrow$ Prediction Classification.

8. Euclidean Space Classification Loss $\mathcal{L}_{euc}$ Computation, Hyperbolic Space Classification loss $\mathcal{L}_{hyp}$ is computed, and the classification loss $\mathcal{L}_{final}$ for feature Z is calculated.The total loss $\mathcal{L}_{total}$ is computed.

9. Parameters are updated via stochastic gradient ascent to maximize $\mathcal{L}_{total}$.

10. End for.

11.Obtain the final trained model.

Output: Predicted labels for each tagged document.

B.3 visualization

We visualize the four models in Figure 7 on the MR test set using the t-SNE tool (Van der Maaten and Hinton, 2008). The models include: EHGCN, EHGCN w/o Hyperbolic, EHGCN w/o Euclidean, and W/O Pref-Words. It is visually evident that the EHGCN model learns more differentiated embeddings for categorized documents compared to other models and achieves closer intra-class distances.

C Related Concepts

In this section, we summarize the core concepts and corresponding formulas involved in the model. Specifically, we first introduce the fundamental concepts and decomposition workflow of multilevel discrete wavelet decomposition; then describe the relevant concepts and formulas of the Graph Convolutional Network (GCN) model used in our framework; and finally elaborate on the basic definitions of hyperbolic space, including the exponential map and logarithmic map.

C.1 Multilevel Discrete Wavelet Decomposition

Multilevel Discrete Wavelet Decomposition (MDWD) is a feature processing method based on recursive filter banks(Mallat, 1989). It decomposes raw features through low-frequency and high-frequency filter groups across multiple levels, partitioning features at each layer into low-frequency approximation features (reflecting global trends) and high-frequency detail features (capturing local abrupt changes). The low-frequency and high-frequency filters are determined by the wavelet function, with common choices including the Haar wavelet, bior-3.5 wavelet, sym8 wavelet, and db4 wavelet(Zhang, 2019).

Let the original feature be $X \in \mathbb{R}^d$, where d denotes the dimensionality. Let X_l^i and X_h^i represent the low-frequency and high-frequency sub-features generated at the i-th layer through the low-frequency and high-frequency filters, respectively. At the $(i+1)$-th layer, MDWD employs a low-frequency filter $l = \{l_1, l_2, \ldots, l_k\}$ and a high-frequency filter $h = \{h_1, h_2, \ldots, h_k\}$, where $k \ll d$, The filters l and h satisfy the following properties:

$$\begin{cases} h_i = (-1)^i \cdot l_{k+1-i} \\ \sum_{i=1}^k l_i^2 + \sum_{i=1}^k l_i^2 = 1 \end{cases} \tag{14}$$

Perform a convolution operation on the low-

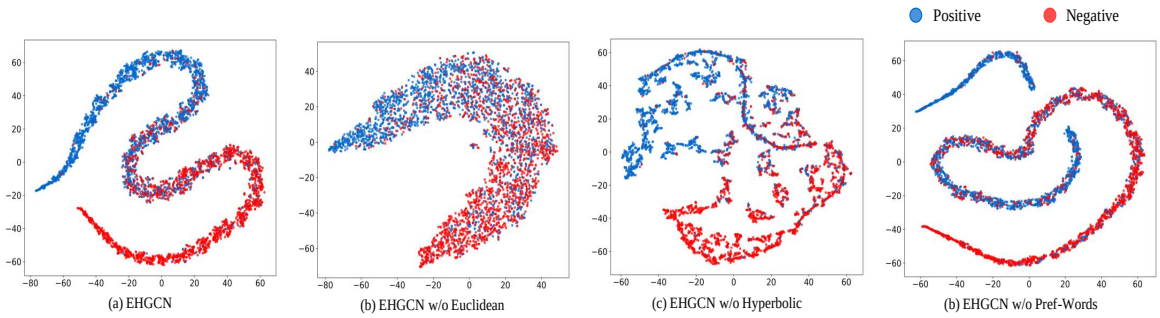


Figure 7: The t-SNE visualization of test set document embeddings on MR.

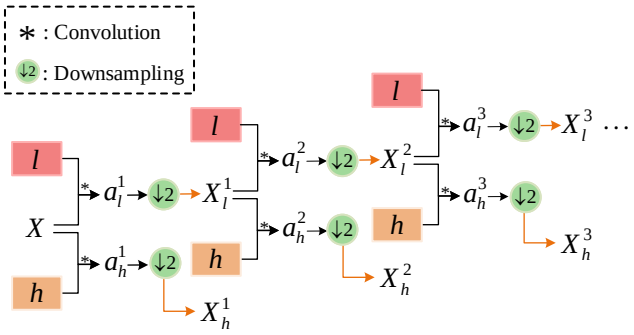


Figure 8: Framework Description of MWDN

frequency sub-features from the previous layer:

$$a_l^{i+1}(n) = \sum_m^K x_l^i(n+m-1) \cdot l_k \qquad (15)$$

$$a_h^{i+1}(n) = \sum_m^K x_l^i(n+m-1) \cdot h_k \qquad (16)$$

where $x_l^i(n)$ is the n-th element of the low-frequency sub-feature at the i -th layer, and x_l^0 represents the input feature. The intermediate variable components are downsampled by $1/2$.

$$a_l^i = \{a_l^i(1), a_l^i(2), ...\} \qquad (17)$$

$$a_h^i = \{a_h^i(1), a_h^i(2), ...\} \qquad (18)$$

The i-th level low-frequency and high-frequency components $x_l^i(n)$ and $x_h^i(n)$ can be generated. As shown in Figure8, the i-th layer decomposition result of x is $X^i = [X_h^1, X_h^2, \dots X_h^i, X_l^i]$. The three-layer decomposition feature of X is then represented as $X^3 = [X_h^1, X_h^2, X_h^3, X_l^3]$.

C.2 GCN Model

Formally, let $G = (V, \varepsilon)$ represent a graph, where $V(n = |v|)$ and ε denote the node set and edge set, respectively. Each node is self-connected (i.e., has a self-loop). The initial node features are represented by $X \in \mathbb{R}^{n \times d}$, where n is the number of nodes and d is the feature dimensionality. To enable nodes to aggregate neighborhood information, we introduce an adjacency matrix $A \in \mathbb{R}^{n \times n}$, where A_{ij} represents the correlation coefficient between nodes v_i and v_j , with $A_{ii} = 1$. We normalize the adjacency matrix to ensure uniform information propagation and numerical stability:

$$\tilde{A} = D^{-\frac{1}{2}} A D^{\frac{1}{2}} \qquad (19)$$

where D is the degree matrix, with $D_{ij} = \sum_j A_{ij}$. At the K-th convolutional layer, node embeddings are computed as:

$$H^{(k)} = \sigma(\tilde{A} H^{(k-1)} W_k) \qquad (20)$$

where $k \in \{1, 2, \dots, h\}$, h is the number of convolutional layers, σ is the activation function, $W_k \in \mathbb{R}^{d \times d}$ is a trainable weight matrix, and $H^{(0)} = X$.

C.3 Hyperbolic Geometry Representation and Domain Aggregation

Hyperbolic space is a smooth Riemannian manifold with constant negative curvature(Benedetti and Petronio, 1992). Common models of hyperbolic space include five types: the Lorentz model (also known as the hyperboloid model), the Klein model, the Jemisphere model, the Poincaré ball model, and the Poincaré half-plane model(Cannon et al., 1997). This paper adopts the Lorentz model due to its superior numerical stability(Nickel and Kiela, 2018). In Table5, $\mathbb{H}^{d,k}$ is the first-order local approximation of the manifold at x, and the Minkowski inner product is positive definite on $\mathbb{H}^{d,k}$. Let $x \in \mathbb{H}^{d,k}$, $u \in T_x\mathbb{H}^{d,K}$ with $\langle u, u \rangle_{\mathcal{L}} = 1$, then the following conclusions can be derived:

Hyperbolic Distance: Let $x \in \mathbb{H}^{d,k}$ and $y \in \mathbb{H}^{d,k}$ be two points on the aforementioned manifold.

Symbol	Illustrate
$\langle \cdot, \cdot \rangle_{\mathcal{L}} : \mathbb{R}^{d+1} \times \mathbb{R}^{d+1} \to \mathbb{R}$	Minkowski inner product
$\mathbb{H}^{d,k}$	d-dimensional hyperbolic manifold with constant negative curvature $-1/k$ $(k > 0)$
$T_x\mathbb{H}^{d,K}$	Tangent space at point x
$\|v\|_{\mathcal{L}} = \sqrt{\langle v, v \rangle_{\mathcal{L}}}$	Norm of $v \in T_x\mathbb{H}^{d,K}$
$o := \{\sqrt{K}, 0, \ldots, 0\} \in \mathbb{H}^{d,K}$	Origin (North Pole) in Lorenz model

Table 5: Basic symbol definitions of the Lorenz model

The distance function between them is defined as:

$$d_{\mathcal{L}}^{K}(x, y) = \sqrt{K} \operatorname{arccosh}\left(-\frac{\langle x, y \rangle_{\mathcal{L}}}{K}\right) \quad (21)$$

Exponential Map and Logarithmic Map: The non-Euclidean geometric properties of hyperbolic space (e.g., negative curvature, nonlinear geodesics) make it challenging to directly perform standard neural network operations (e.g., addition, matrix multiplication) on the manifold. By applying the logarithmic map, points in hyperbolic space can be projected to the corresponding tangent space (Euclidean space), enabling the use of well-established Euclidean geometric operations (e.g., linear transformations, attention mechanisms) within the tangent space. The results are then mapped back to hyperbolic space via the exponential map. Therefore, the exponential map and logarithmic map fundamentally bridge hyperbolic and Euclidean geometries, overcoming the inherent limitations of native hyperbolic space operations while retaining their hierarchical modeling advantages. For $x \in \mathbb{H}^{d,k}$,$v \in T_x\mathbb{H}^{d,K}$ with $v \neq 0$, and $y \in \mathbb{H}^{d,k}$ with $y \neq x$, the exponential map and logarithmic map are defined as:

$$\exp_x^K(v) = \cosh\left(\frac{|v|_{\mathcal{L}}}{\sqrt{K}}\right) x + \sqrt{K} \sinh\left(\frac{|v|_{\mathcal{L}}}{\sqrt{K}}\right) \frac{v}{|v|_{\mathcal{L}}} \quad (22)$$

$$\log_x^K(v) = d_{\mathcal{L}}^{K}(x, y) \quad (23)$$

Mapping from Euclidean Space to Hyperbolic Space: Let $x^{0,E} \in \mathbb{R}^d$ denote the input Euclidean feature. Since $\langle (0, x^{0,E}), 0 \rangle = 0$, and $\langle (0, x^{0,E}) \rangle$ lies in $T_o\mathbb{H}^{d,K}$, where $v \in T_x\mathbb{H}^{d,K}$, the mapping of $x^{0,E}$ to hyperbolic space $x^{0,H}$ is defined as:

$$x^{0,H} = \exp_0^K\left[\cosh\left(\frac{\|x_0^E\|}{\sqrt{K}}\right) \cdot 0 + \sqrt{K} \sinh\left(\frac{\|x_0^E\|}{\sqrt{K}}\right) \cdot \frac{x_0^E}{\|x_0^E\|}\right] \quad (24)$$

Hyperbolic Linear Transformation: Hyperbolic linear transformations require the use of exponential and logarithmic maps to complete the transformation process. Linear transformations primarily involve multiplying input vectors with matrices and applying bias translation operations to the vectors. Let x^H be a point in hyperbolic space projected onto the tangent space $T_o\mathbb{H}^{d,K}$, and $W \in \mathbb{R}^{d' \times d}$ be the learnable weight matrix. The hyperbolic matrix multiplication is defined as:

$$W \otimes^K x^H = \exp_o^K\left(W \log_o^K(x^H)\right) \quad (25)$$

where $\log_o^K(\cdot) \in \mathbb{H}^{d,K}$ and $\exp_o^K(\cdot) \in \mathbb{H}^{d',K}$. To preserve the hierarchical relationships and geometric invariance of hyperbolic space and avoid distortion during hyperbolic bias translation, first set $b \in T_o\mathbb{H}^{d',K}$, then translate x^H to the tangent space $T_{x^H}\mathbb{H}^{d',K}$. The hyperbolic bias addition is defined as:

$$x^H \oplus^K b = \exp_{x^H}^K\left(P_{o \to x^H}^K(x^H)\right) \quad (26)$$