# Improved Gaussian Mixture Model for Feature Classification Based on Hypergraph Structure

1st Zhiwei Zhao
*Dalian University of Technology*
*School of Computer Science and Technology*
Dalian, China
1329065761@qq.com

2nd Qiang Zhang
*Dalian University of Technology*
*School of Computer Science and Technology*
Dalian, China
zhangq@dlut.edu.cn

3rd Kai Lin
*Dalian University of Technology*
*School of Computer Science and Technology*
Dalian, China
link@dlut.edu.cn

***Abstract*—In recent years, interest in deep learning methods based on nonlinear structures for natural language processing has significantly increased. At the same time, hypergraph convolutional neural networks have started to replace graph convolutional neural networks, allowing the integration of implicit information with the complexity of the processed data. However, these hypergraph-based methods face challenges, such as the inability to capture hidden relationships and the inaccurate calculation of feature similarity between structures. A recent study proposes a hypergraph structure based on probability distributions. This approach constructs hypergraphs around multi-label correspondences and introduces a convolutional update to the framework. The study's approach includes two main components: the hypergraph construction model based on the refined EM algorithm and the adaptive dynamic convolution model. The refined EM algorithm calculates feature similarity distance to dynamically create the hypergraph structure, enhancing structural integrity at each layer. Furthermore, the study introduces a dynamic convolution module consisting of two phases: vertex convolution and hyperedge convolution. This module combines the features between the vertices and hyperedges. The study evaluates the proposed method using the Cora and Meld datasets, demonstrating its superior performance compared to contemporary state-of-the-art methods. Additionally, supplementary experiments demonstrate the method's efficacy and resilience across diverse data distributions.**

***Index Terms*—hypergraph, hybrid Gaussian model, dynamic convolution**

## I. Introduction

Graph structures, as extensions of linear relational representations, provide a powerful method to interpret and represent relationships between data using graph-theoretic methods [1]. Graph structures have had a significant impact on reinforcement learning and play a crucial role in the efficiency of representation learning. Traditional graph structures have limitations in capturing feature relationships, especially in multi-dimensional features, leading to suboptimal experimental results [2]. To address these limitations, a more advanced structure is needed to support representation learning in neural network models [3]. Researchers are turning to graph theory to extend simple graph structures to hypergraph structures, which can represent complex data relationships more effectively. As shown in Figure 1, unlike traditional graph structures, hypergraphs can connect multiple vertices, allowing for better representation of higher-order relationships in the data and more accurate descriptions of complex data structures and interconnections. Although the research field of hypergraphs

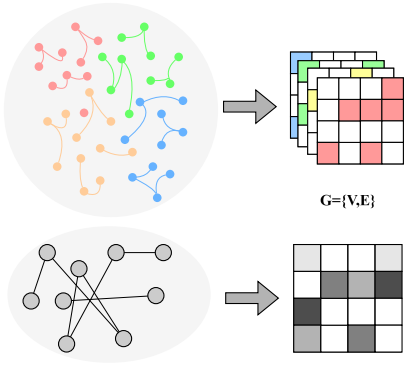



Fig. 1. The comparison between graph and hypergraph.

is relatively new, their high adaptability in complex multi-dimensional data environments has already proven valuable in various applications. Combining hypergraph and neural network models can enhance the ability to handle relationships between data [3]. To address this issue, Jiang et al. proposed the dynamic hypergraph construction model DHGCN [4]. This model is divided into two parts, constructing and updating the network to improve the accuracy of the hypergraph structure. It outperforms existing methods on the Cora citation network and microblogging dataset and is robust across different data distributions. However, DHGCN focuses on improving model accuracy through dynamic convolution under the hypergraph structure, which ignores implicit attributes n the feature space. Additionally, using a uniform similarity projection space for feature mapping in the hypergraph structure loses relationships

DOI reference number: 10.18293/SEKE2024-157

between objects and weakly correlated labels.

To address the above shortcomings, this paper proposes an improved hybrid Gaussian model to compute the similarity distribution. This model works with a hypergraph structure to dynamically convolve and update the multidimensional probability distribution [3]. In the similarity distribution calculation module, the hypergraph structure is constructed by iteratively obtaining the multidimensional probability distribution mapping using an improved expectation maximization algorithm. The dynamic convolution module computes adaptive weight transformation matrices to perform node convolution and hyperedge convolution, respectively. Compared to the DHGNN method, our approach embeds implicit relations into object features more effectively [4]. Additionally, adaptively changing the hyperedge weights ensures better integration of this embedding information into DHGN and global feature information. In this study, we apply our model to data with implicit relationships and perform label classification on the Cora citation dataset and the MELD multi-modal dataset.

The contributions of this paper are as follows:

1) We propose a hypergraph construction method based on an improved Gaussian mixture model to compute the similarity distribution. This method employs a sampling-optimized EM algorithm to compute the distribution of mappings from object features to the label space, embedding implicit relationships into the features.
2) We propose a probability distribution-based adaptive updating method, which uses probability distributions to compute the weights of hyperedges and update the hypergraph structure via vertex convolution and hyperedge convolution modules. This ensures that the hypergraph structure can fuse the implied relationships into the global features.
3) We conducted experiments on label classification using textual citation data. Our method demonstrates superior performance compared to current state-of-the-art approaches and exhibits enhanced resilience across various data distributions.

## II. Related Work

### A. Graph-based Deep Learning

Graph Neural Networks (GNNs) are neural network models designed for graph data. Their main purpose is to gain a deeper understanding of the nodes and edges in the graph structure to analyze and process graph data efficiently [5]. Graph Convolutional Networks (GCNs) are an important variant of GNNs. Through multi-layer convolutional operations, GCNs can gradually extract and integrate the local and global features of the nodes while preserving the structural information of the graph. This process achieves the learning and predictive representation of the entire graph [6].

Daniel et al. (2018) first proposed an adversarial attack model on attribute graphs and investigated challenging poisoning and causal attacks [7]. In the same year, Chandler Zuo explored the regularization effect of the Fast Gradient Sign Method (FGSM) based on IG-FGSM within the framework of the Generalized Linear Model (GLM). The IG-FGSM algorithm model was proposed based on this work [8]. Thomas et al. proposed a scalable semi-supervised learning method based on graph convolutional networks [9]. Joan et al. explored the possibility of performing convolutional operations on signals defined in more general domains without the need for translation groups [10].

### B. Hypergraph Learning

Hypergraphs, as a generalization of graphs, have significant advantages [11] over simple graph structures in characterizing complex relationships and expressing higher-order features. Combining hypergraphs with neural network models can enhance the ability to manage relationships between data. William et al. (2017) proposed an inductive representation learning method, Graph SAGE, which samples features of neighboring nodes and aggregates them into central node features [12]. In the same year, Cao et al. proposed a new hypergraph neural network model, HGNN, which effectively handles higher-order relationships and better captures complex dependencies between nodes [13]. Building on the hypergraph neural network model, Jiang et al. proposed the dynamic hypergraph construction (DHGCN) method [14]. This method divides the model into two parts, enhancing the accuracy of the hypergraph structure through network construction and updating.

## III. Method

In this section, we will introduce the proposed improved probability distribution hypergraph neural network model in detail. The network framework is shown in Fig. 3.1. The model consists of two parts: probability distribution hypergraph construction and adaptive hypergraph convolution. We will first introduce these two parts.

### A. Probability distribution hypergraph construction

Given a feature embedding set $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]$, where $\mathbf{x}_i (i = 1, 2, \ldots, n)$ denotes the feature of the $i$th sample. In a hypergraph, vertex $u$ denotes a sample, and hyperedge $e$ denotes a collection of samples. Thus, a hypergraph can be expressed as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ denotes the vertex set and $\mathcal{E}$ denotes the set of hyperedges.

We introduce a probability distribution space to represent the multidimensional label distribution, characterizing the implied object properties. The feature similarity of the object is calculated using the Euclidean distance formula as follows:

$$r_{ic} = \frac{\pi_c N\left(x_i \mid \mu_c, \Sigma_c\right)}{\sum_{k=1}^{K} \pi_k N\left(x_i \mid \mu_k, \Sigma_k\right)} \tag{1}$$

Here, the specific feature variables of the $i$th and $j$th samples in the matrix $\mathbf{X}$ are represented in the $r$th sample feature variable of the $i$th sample, and $R$ represents the total number of features.

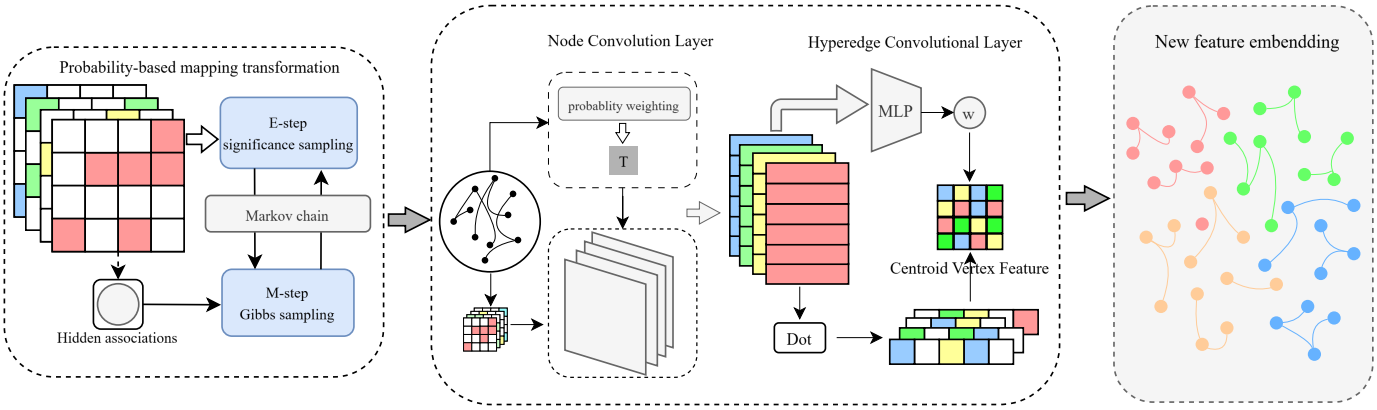



Fig. 2. Overview of Our DHGNN Framework. The first part of the framework describes the process of hypergraph construction for the central vertex and its neighborhood. It uses a GMM algorithm, improved by the sampling method, to iteratively generate object-label mapping relationships. This method incorporates implicit relationships into the node features and generates the hypergraph structure based on probability distributions. In the second part, the vertex features in the hypergraph are aggregated into one-dimensional hypergraph features through vertex convolution. Meanwhile, the features in the neighboring hyperedges are aggregated into central vertex features through hyperedge convolution. After performing these operations on all vertices of the current layer feature embedding, we obtain a new feature embedding. This new embedding will be used to construct a new hypergraph structure, as shown in the third frame.

Next, we use a hybrid Gaussian model optimized for the sampling method to compute a matrix of object feature probability distributions for each label cluster. This model is trained using a modified Expectation Maximisation (EM) algorithm [15]. Using importance sampling to approximate the expectation computation in the E-Step reduces the computational complexity of the model. Due to the complexity of the original target probability distribution, we use the normal distribution to approximate it. The specific sampling formula is as follows:

$$E(f) = \int f(x)\frac{p(x)}{q(x)}q(x)\,dx \approx \frac{1}{N}\sum_{n=1}^{N}\frac{p(x)}{q(x)}f(x) \qquad (2)$$
$$x^n \sim q(z)$$

where the importance weight is $w^n = \frac{p(x^n)}{q(x^n)}$. At this point, a weighted importance sampling method is used to minimize the difference between the approximate probability distribution and the original probability distribution:

$$E(f) \approx \sum_{n=1}^{N}\frac{w^n}{\sum_{n=1}^{N} w^m}f(x^n) \qquad (3)$$

The distribution of importance weights uses a normal distribution as an approximation. At each iteration, the probability distribution obtained from the previous sample is used as a trajectory to compute the importance weights for the current iteration.

To compute the implied parameters in the probability distributions, our maximization formula for $\theta$ in the M-step can be converted into a full log-likelihood expectation calculation using the following formula:

$$Q(\theta, \theta^{old}) = \int p(Z|X,\theta^{old})\ln p(Z|X,\theta)\,dZ \qquad (4)$$

where $Z$ is sampled using specific samples, and the computation of $p(Z|X,\theta)$ is as follows:

$$p(Z|X,\theta) = \frac{Z^x(1-Z)^{1-x}\cdot\frac{\alpha}{\sqrt{2\pi}\sigma}e^{-\frac{(Z-1/2)^2}{2\sigma^2}}}{\int_0^1 Z^x(1-Z)^{1-x}\cdot\frac{\alpha}{\sqrt{2\pi}\sigma}e^{-\frac{(Z-1/2)^2}{2\sigma^2}}\,dZ} \qquad (5)$$

Here, the Markov Chain Monte Carlo (MCMC) method is used to sample the new round of iterations using the old values. This sampling method effectively samples the multi-dimensional target distribution in a multivariate environment. It circumvents the drawback of not being able to directly calculate the posterior distribution. Additionally, using multiple iterations helps reduce the errors generated by sampling.

In the first iteration of hidden variables, a random sample is obtained as $\Theta = (\theta_1^1, \theta_2^1, \ldots, \theta_n^1)$, where the joint probability distribution $P(\theta_1^1|\theta_2^1, x_1, x_2, x_3, \ldots, x_n)$ is used. We simplify the above formula by converting the joint posterior distribution into a conditional probability formula:

$$\begin{aligned} &p(\theta_1^1|\theta_2^1, x_1, x_2, x_3, \ldots, x_n) \\ &\quad\propto p(\theta_1^1, \theta_2^1|x_1, x_2, x_3, \ldots, x_n) \\ &\quad\propto p(x_1, x_2, x_3, \ldots, x_n|\theta_1^1, \theta_2^1)p(\theta_1^1, \theta_2^1) \end{aligned} \qquad (6)$$

Therefore, the Q function in the improved M-step can be expressed as:

$$Q(\theta, \theta^{old}) = \int p(\theta_1, \theta_2, \ldots, \theta_n, \theta^{old})\ln p(\theta_1, \theta_2, \ldots, \theta_n, \theta)\,d\theta_1 d\theta_2 \ldots d\theta_n \qquad (7)$$

On this basis, we iteratively train using the standard EM algorithm to obtain the object's probability distribution matrix. Finally, the probability distribution matrix and the hypergraph Laplacian matrix are used as inputs to generate the label cluster-based probability distribution hypergraph.

### B. Adaptive hypergraph convolution

The adaptive hypergraph convolution method consists of two parts: the adaptive vertex convolution module and the hyperedge convolution module. The adaptive vertex convolution module creates a transformation matrix through multidimensional probability distributions to induce vertex features to approximate high-likelihood label clusters and form hyperedge features. It then continues to convolve the hyperedge features to aggregate hyperedges with similar probability distributions near the central vertex.

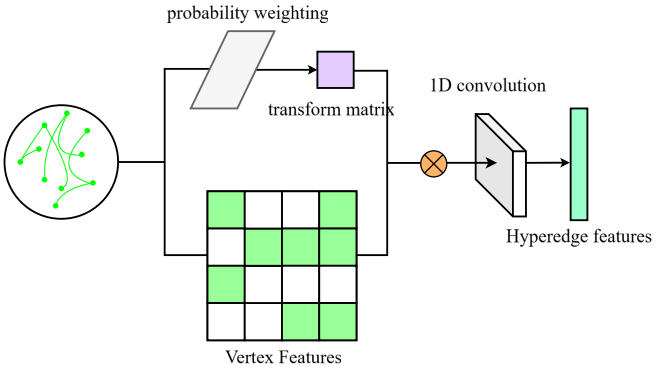



Fig. 3. Adaptive Vertex Convolution Module. For $k$ vertices, the $k \times k$ transformation matrix is calculated by combining the probability distribution matrices of the vertices. After being processed by the multilayer perceptron, the transformed vertex feature matrix is multiplied by the transformation matrix to obtain the transposed and weighted vertex feature matrix. Then, one-dimensional convolution is performed to obtain the one-dimensional hyperedge features.

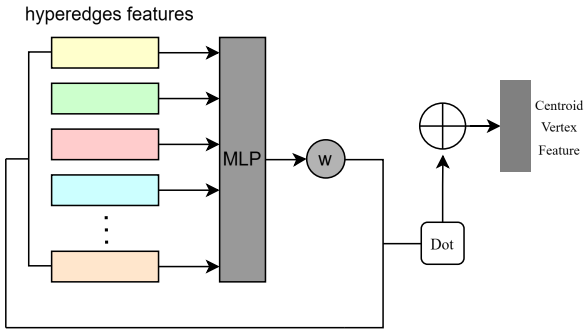



Fig. 4. The hyperedge features are aggregated using a self-attentive mechanism. The adaptive weights $w$ are computed from the hyperedge features through an MLP. The average of the adaptive weights of the hyperedge features is then computed as the central vertex feature.

*1) Adaptive vertex convolution:* Most popular vertex convolution methods today use pre-computed transformation matrices to perform convolution operations on vertices, which are an extension of pooling schemes. These methods are better suited for explicit object-label mapping relationships that do not contain implicit attributes. To better adapt to these implicit relationships and one-to-many object-label mappings, we propose learning the transformation matrix through the multidimensional probability distributions of the object-label clusters. This matrix is used to weight and rank the vertices, as shown in Figure 3.

During the convolution process, we use a multilayer perceptron (MLP) combined with conditional probability to generate the transformation matrix $\mathrm{T}_p$. The transformed features are then compressed using 1-D convolution [16]. The formula is as follows:

$$\begin{aligned} \mathrm{T}_p &= MLP(p(k)\mathbf{X}_u) \\ p(k) &= \frac{e^{z_k}}{\sum_{j=1}^{K} e^{z_j}} \end{aligned} \tag{8}$$

$$\mathbf{x}_e = \mathrm{conv}(\mathbf{T} \cdot MLP(\mathbf{X}_u)) \tag{9}$$

*2) Hyperedge convolution:* As shown in Fig. 4, the hyperedge convolution module aggregates one-dimensional hyperedge features into central vertex features, thereby converting one-to-many object-label mappings into one-to-one mappings. Specifically, the weight scores of the hyperedge features are generated using an MLP. The one-to-one central vertex features are then obtained by performing weighted averaging based on these weight scores. The formula is as follows:

$$\begin{aligned} \omega &= \mathrm{softmax}(b + \sum x_e * P_e) \\ x_u &= \sum_{i=0}^{|Adj_u|} \omega^i x_e^i \end{aligned} \tag{10}$$

## IV. Applications and Experiments

In this section, we apply our model to citation networks with intrinsic graph structures. Specifically, for data with intrinsic graph structures, we sample $k$ vertices in the 1st-order neighborhood of $u$. These $k$ vertices also form a hyperedge in $\mathrm{Adj}(u)$.

### A. Experiments in categorizing citation datasets

An open-source text dataset is selected for experimental validation of the theoretical methods proposed in this chapter. Controlled experiments are conducted on the textual dataset to compare the improved GMM hypergraph construction method and the probabilistic hypergraph convolution module proposed in this chapter against the conventional hypergraph convolution model. The effectiveness and superiority of the two approaches are verified on the textual dataset and data containing hidden high-latitude relations through different ablation experiments.

*1) Cora dataset:* The Cora dataset is used for categorizing scholarly literature, mainly in machine learning and natural language processing research. The nodes in the knowledge network represent papers, and the edges connecting the nodes represent citation relations. Each node includes information such as title, author, and more. Based on this information, the dataset categorizes papers into one or more categories, such as artificial intelligence, data mining, and target recognition research.

*2) Experiment setup:* We set specific experimental metrics for the label classification task on the Cora dataset. For the citation label classification task, we use test metrics including Accuracy (ACC) and Loss Rate (LOSS), while we define LR as the training set size of the dataset. The experiments compare our method with GCN model, HGCN model and DHGCN model based on K-means method and the results are presented in graphical form.

*3) Result:* We compared our method with the latest graph/hypergraph-based neural network methods on the Cora dataset. The experimental results are shown in Table 1, where our method outperforms state-of-the-art methods by 0.6% to 2% when using 1% to 8% randomly sampled data as the training set. The results show that the hypergraph structure is more competitive compared to the graph structure when the dataset is small. Additionally, the introduction of the probability distribution enhances the hypergraph structure's ability to express implicit relationships.

TABLE I
PERFORMANCE COMPARISON USING DIFFERENT METHODS ON CORA

| LR | #train | HGCN | DHGCN | DHGCN(ours) |
|---|---|---|---|---|
| 1 | 27 | 74.8 | 79.4 | 80.8 |
| 2 | 54 | 75.4 | 80.3 | 80.9 |
| 4 | 108 | 78.8 | 80.7 | 81.7 |
| 6 | 162 | 79.0 | 80.9 | 82.9 |
| 8 | 216 | 79.6 | 80.9 | 82.6 |
| 10 | 270 | 79.9 | 81.6 | 81.3 |

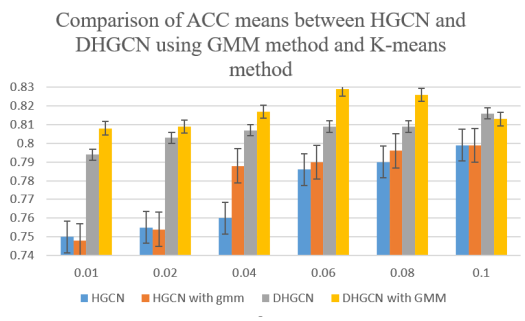



Fig. 5. Comparison of ACC means between HGCN and DHGCN using GMM method and K-means method.

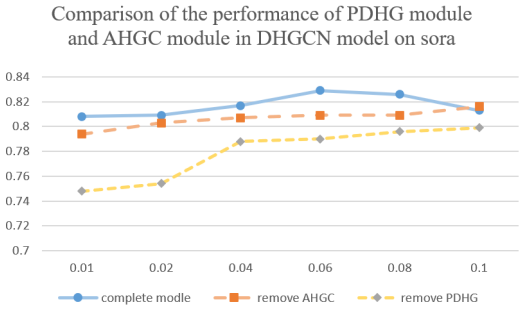



Fig. 6. Performance comparison of PDHG and AHGC modules in DHGCN model on sora dataset.

*4) Ablation experiments:* To demonstrate our method's effectiveness, this paper conducts multiple sets of ablation experiments under the same experimental conditions. The ablation experiments compare the performance of the hypergraph representation using the K-means method with that of the DHGCN model, both with and without the PDHG and AHGC modules. Our method is also evaluated under the same experimental conditions. In the case where the AHGC module is removed, the model employs the standard convolutional module on the Cora dataset. The results of the ablation experiments using the K-means method are presented in Fig. 5, while the results of experiments with the AHGC and PDHG modules removed are depicted in Fig. 6. The experimental data demonstrate that our method achieves superior mapping results compared to the traditional K-means method. This improvement leads to a 2% increase in ACC over the traditional DHGCN model. Compared to the traditional HGCN model, the DHGCN model incorporating the AHGC module demonstrates faster convergence and improved label classification accuracy. The inclusion of the AHGC module leads to a 4% enhancement in label classification accuracy. Furthermore, when the dataset size reaches 6%, the model's performance peaks with a 3% improvement.

### B. Experiments in Sentiment Analysis of Multi-modal Datasets

In addition to the citation network experiments, we evaluated our model on a more complex task, namely sentiment analysis on multi-modal datasets. Multi-modality is an important feature of social media. We use hypergraphs to model higher-order relationships between different modalities. The multi-modal hypernodes jointly represent the correlations between vertices. In our experiments, we use the MELD dataset to evaluate our hypergraph model.

*1) MELD dataset:* MELD (Multi-modal Emotion Line Dataset) is a multi-modal emotion recognition dataset derived from the classic English TV series "Friends". The MELD dataset consists of 1,400 dialogue snippets with a total of 13,000 segments. Each dialogue segment is labeled with seven emotions including anger, disgust, sadness, joy, neutrality, surprise, and fear.

*2) Experiment setup:* We constructed three hyperedge sets for each of the three modalities and merged these sets into one multi-modal hyperedge set. The number of vertices contained in each cluster is 64. We select 2 nearest clusters and a labeled cluster from the GMM clustering as the neighboring hyperedge set for each vertex. We compare our model with recent multi-modal sentiment prediction methods, Dialogue RNN method [17]; Dialogue GCN method [18]; MMGCN method [19].

*3) Result:* The experimental results of the different methods on the MELD dataset are shown in Table 2, where each model obtains an average F1 weight score on the MELD dataset. It can be seen that our method improves the performance by 7% to 8% over the Dialogue RNN and Dialogue GCN methods, and by 5% compared to MMGCN.

*4) Ablation experiments:* To demonstrate the state-of-the-art performance of our method, we conducted a series of

TABLE II
PERFORMANCE COMPARISON USING DIFFERENT METHODS ON MELD

| Method | ACC |
|---|---|
| Dialogue RNN | 57.11 |
| Dialogue GCN | 58.23 |
| MMGCN | 60.84 |
| our method | 65.12 |

ablation experiments using the K-means clustering method and compared the performance of our clustering method on the dataset, as shown in Fig. 7. The experimental data demonstrate that our method achieves superior mapping results compared to the traditional K-means method, with an experimental ACC that is 4% better than the traditional DHGCN model. Furthermore, the model performance reaches its optimal level when the number of convolutional layers reaches 8.

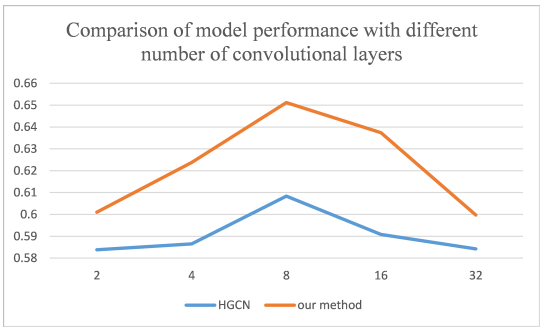



Fig. 7. Comparison of ACC means between HGCN and DHGCN using GMM method and K-means method.

## V. CONCLUSIONS

In this study, we propose enhancements to the hybrid Gaussian model for multi-mapping sample classification. Our improvements involve replacing the uniform mapping space with probability distributions for hypergraph representation and optimizing the hypergraph structure for input data features. We also introduce a dynamic hypergraph convolution module based on multidimensional probability distributions to dynamically prune and update the structure, remove erroneous associations and low-confidence relationships, and express high-latitude associations among the data, while also uncovering logical relationships between hidden features.

After thoroughly validating the reliability and effectiveness of the model using the Sora dataset, we found that our approach achieves similar or superior performance compared to state-of-the-art methods. Additionally, we have investigated the validity of the PDHG module and the AHGC module.

In future work, it may be beneficial to explore improved and more interpretable hypergraph construction methods.

## REFERENCES


[1] Li, Z., Liu, F., Yang, W., et al. "A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects." *IEEE Trans Neural Netw Learn Syst*, vol. 33, no. 12, pp. 6999-7019, 2022.

[2] Zhang, Z., Cui, P., Zhu, W. "Deep Learning on Graphs: A Survey." *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 1, pp. 249-270, 2022.

[3] Feng, Y., Haoxuan, Y., Zizhao, Z., et al. "Hypergraph Neural Networks." 2018.

[4] Jiang, J., Yuxuan, W., Yifan, F., et al. "Dynamic Hypergraph Neural Networks." *PROCEEDINGS OF THE TWENTY-EIGHTH INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE*, pp. 2635-2641, 2019.

[5] Zhou, J., Ganqu, C., Shengding, H., et al. "Graph neural networks: A review of methods and applications." *AI Open*, pp. 157-81, 2020.

[6] Li, J., Tao, X., Liang, C., et al. "Adversarial Attack on Large Scale Graph." 2020.

[7] Alom, M. Z., Paheding, S., Tarek-M, T., et al. "Handwritten Bangla Digit Recognition Using Deep Learning." 2017.

[8] Zuo, C. "Regularization Effect of Fast Gradient Sign Method and its Generalization." Ithaca: Cornell University Library, arXiv.org, 2018.

[9] Li, J., Yongfeng, H., Heng, C., et al. "Semi-Supervised Hierarchical Graph Classification." 2022.

[10] Bruna, J., Mallat, S. "Invariant scattering convolution networks." *IEEE Trans Pattern Anal Mach Intell*, vol. 35, no. 8, pp. 1872-1886, 2013.

[11] Ramos, M. V. M., Queiroz, R. J. G. B. "Formalization of simplification for context-free grammars." 2015.

[12] Ahmed, N. K., Rossi, R. A., Zhou, R., et al. "Inductive Representation Learning in Large Attributed Graphs." 2017.

[13] Zhang, H., Xin, L., Lidong, B. "Video-LLaMA: An Instruction-tuned Audio-Visual Language Model for Video Understanding." arXiv.org, 2023.

[14] Liu, L., Feng, H., Xuan, L., et al. "Multi-view Contrastive Learning Hypergraph Neural Network for Drug-Microbe-Disease Association Prediction." In: *IJCAI International Joint Conference on Artificial Intelligence*, pp. 4829-4837, Macao, China, 2023.

[15] Ramos, M. V. M., Queiroz, R. J. G. B. "Formalization of simplification for context-free grammars." 2015.

[16] Zaremba, W., Sutskever, I., Vinyals, O. "Recurrent Neural Network Regularization." 2014.

[17] Majumder, N., Poria, S., Hazarika, D., et al. "DialogueRNN: An Attentive RNN for Emotion Detection in Conversations." Honolulu: 2019, pp. 6818-6825.

[18] Ghosal, D., Majumder, N., Poria, S., et al. "DialogueGCN: A graph convolutional neural network for emotion recognition in conversation." arXiv, 2019.

[19] Hu, J., Yuchen, L., Jinming, Z., et al. "MMGCN: Multimodal Fusion via Deep Graph Convolution Network for Emotion Recognition in Conversation." Ithaca: Cornell University Library, arXiv.org, 2021.