**COMP4434 Report**

**UniGNN: a Unified Framework for Graph and Hypergraph Neural Networks**

Paper from IJCAI-21.

Group member:

Siu Yau Shing

Lam Yam Ying

Kan Wai Yi

**Abstract**

In recent years, graph neural networks (GNNs) have demonstrated significant success across various tasks involving graph-structured data, such as node classification, link prediction, and graph classification. However, many real-world relationships are inherently higher-order and cannot be adequately captured by pairwise edges in traditional graphs. Hypergraphs, which allow edges (hyperedges) to connect more than two nodes, naturally represent such complex relationships and have found applications in diverse domains including computer vision [1], recommendation systems [2], and natural sciences [3].Given the increasing demand for effective hypergraph representation learning, we selected the paper "UniGNN: a Unified Framework for Graph and Hypergraph Neural Networks" as the foundation of our study [4]. The key motivation for choosing this work lies in its attempt to generalize the architecture and training strategies of conventional GNNs to hypergraphs under a unified framework. Unlike earlier approaches which rely on hypergraph approximations or special structures, UniGNN provides a clean and elegant message-passing framework that subsumes both graphs and hypergraphs [4]. To evaluate the effectiveness of UniGNN, we compare a modified version of the proposed model (ModifiedGNN) against traditional machine learning methods, including Support Vector Machines (SVM), Random Forest (RF), and Multi-Layer Perceptron (MLP), on the PubMed dataset. This comparison not only serves to validate the strength of deep hypergraph models but also highlights the scenarios where such models offer tangible benefits over classical baselines.

**1. Introduction**

**Background & Motivation**

Graphs have long been used to represent pairwise relationships, but many real-world domains—particularly in biomedical and textual data—exhibit relationships that go beyond simple dyadic links [5]. In the PubMed dataset, for instance, scientific articles (nodes) often span multiple overlapping topics or terms, which naturally form high-order associations among groups of documents [5]. Hypergraphs generalize conventional graphs by allowing hyperedges to connect arbitrary subsets of nodes, offering a richer structure to model these interactions.

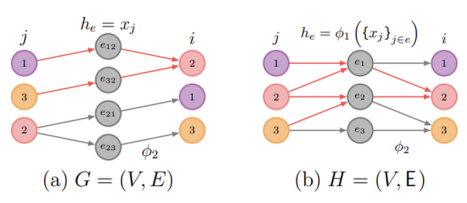


Figure 1: Demonstration of two-stage message passing for Graph G (a) and Hypergraph H (b) where V is a set of vertices and E are the edges. Where V is a set of vertices and E are the edges and hyperedges.

Although conventional GNNs such as Graph Convolutional Networks (GCN)[6] or Graph Attention Networks (GAT) [7] have demonstrated strong performance, they are fundamentally limited to pairwise edges. UniGNN addresses this gap by proposing a unified message-passing framework that can handle various forms of hypergraph [4]. Its appeal lies in its simplicity and generality—providing a blueprint for extending standard GNN operations to the hypergraph setting without relying on specialized approximations.

However, UniGNN's generalized formulation also introduces limitations. Its uniform treatment of hyperedges can underrepresent semantic nuances, especially in noisy or sparse hypergraphs where not all high-order interactions are meaningful. Also, the aggregation strategy in UniGNN may suffer from over-smoothing or loss of expressiveness when deeper layers are introduced.

In this context, we propose ModifiedGNN—an extension of UniGNN that incorporates residual gating mechanisms and carefully selected aggregation strategies. Residual gating allows the model to dynamically control the flow of information across layers, mitigating over-smoothing and improving training stability. The architecture also involves a tuned number of message-passing layers to balance depth and generalization.

**Challenges**

Despite their representational power, hypergraph neural networks pose several practical challenges:

First, modeling high-order relationships via hyperedges inherently introduces complexity. As hyperedges connect many nodes, the message-passing process can become noisy or imprecise, especially when hyperedge semantics are weak or poorly defined.

Second, deeper GNN models are vulnerable to over-smoothing, where node embeddings across the graph become indistinguishably similar, thereby reducing classification performance. This is particularly problematic in hypergraphs, where the propagation paths are denser and more entangled.

Third, striking a balance between expressiveness and generalizability remains a core difficulty. A model that aggressively captures high-order patterns may be overfit to noisy or spurious correlations, while simpler models may underfit and miss subtle semantic structures.

To address these challenges, ModifiedGNN adopts residual gating to improve information flow and enhance feature retention across layers. The model architecture is designed with selection of the number of layers and aggregation strategies, aiming to maintain discriminative node representations while minimizing the risk of overfitting or over-smoothing. Through empirical evaluation on the PubMed dataset, we assess the effectiveness of our modifications and explore the trade-offs between traditional and hypergraph-based models in biomedical document classification.

## 2. Related Work

Graph Neural Networks (GNNs) have rapidly evolved, with early models like GCN [6], GAT [7], and GraphSAGE [8] introducing efficient neighborhood aggregation schemes. These message-passing frameworks have become the foundation for learning powerful representations on graph-structured data. However, the expressive power of most GNNs is limited by the 1-Weisfeiler-Lehman (1-WL) test, leading to the development of more expressive architectures such as GIN and higher-order GNNs [9].

Despite the success of GNNs, learning effective embeddings on hypergraphs remains a challenging problem. Early attempts, such as HGNN [10] and HyperGCN [11], use techniques like clique expansion or hypergraph Laplacians to approximate hypergraph structures using standard graphs. These methods, while effective to some degree, often lose critical higher-order relational information and are limited in depth due to over-smoothing effects.

More recent models like HyperSAGE attempt to address these issues by adopting a two-stage message-passing mechanism tailored for hypergraphs [12]. While HyperSAGE achieves strong performance, it suffers from computational inefficiencies and lacks flexibility in integrating modern GNN innovations.

UniGNN represents a significant advancement in this line of research. It proposes a generalized message-passing framework that treats graph and hypergraph neural networks as two instances of the same design principle. By doing so, it enables direct adaptation of well-established GNN architectures—such as GCN[6], GAT[7], GIN[9], and GCNII[13]—into hypergraph settings, providing both theoretical grounding and empirical gains .

In our project, we extend this work by implementing a ModifiedGNN variant based on UniGNN principles and compare its performance to conventional machine learning classifiers including SVM, Random Forest, and MLP. This allows us to explore the practical advantages of deep hypergraph models in real-world classification tasks.

**3. Dataset Description and Preprocessing**

The dataset used in this study is the PubMed citation network [5], a widely recognized benchmark for node classification tasks in the biomedical domain. It consists of 19,717 scientific articles, each represented as a node in a citation graph, with 44,338 edges denoting citation relationships between papers. Each article is assigned to one of three class labels corresponding to specific diabetes-related research topics. The input features are 500-dimensional vectors generated using term frequency-inverse document frequency (TF-IDF) scores based on the article abstracts, which provide a compact representation of each document's content.

To support hypergraph-based modeling, the dataset also includes 7,963 hyperedges. These hyperedges are constructed based on semantic similarity and co-citation patterns, allowing for higher-order relational modeling that captures group-level associations among papers. All preprocessing steps follow the standard practices defined by the UniGNN benchmark. This includes normalization of feature vectors and the use of a consistent and predefined split of data into training, validation, and test subsets. Hyperedges are generated by clustering nodes based on keyword similarity or shared citation contexts, thereby enriching the graph structure with additional semantic information. Also, this data set provides a **fixed training and testing split**, there is no need to perform **cross-validation** for model evaluation. This setup ensures a standardized comparison across models and isolates the performance analysis to a single, consistent test set.

**4. Reproduction of the Deep Learning Framework**

The deep learning model reproduced in this project is a modified Graph Neural Network that incorporates hypergraph aggregation and residual gating mechanisms. The architecture is built around the idea of enhancing message-passing capabilities by including not only direct node-to-node interactions but also multi-node relationships via hyperedges. Each layer of the model includes a learnable transformation of node features, followed by bidirectional message propagation between nodes and hyperedges. After aggregating messages, a residual gating unit determines the extent to which the current layer’s features should be combined with those from the previous layer. This helps preserve important information across layers and mitigates the problem of feature over-smoothing that commonly arises deep GNNs.

Training is conducted using the PyTorch framework. The optimizer used is Adam, and the model is trained with cross-entropy loss for 200 epochs. Regularization is applied using a weight decay of 5e-4 and a dropout rate of 0.6 to prevent overfitting. A learning rate of 0.01 is used throughout training. Early stopping is employed based on validation loss to ensure that the model does not overfit the training set. The full set of training parameters is as follows:

* Optimizer: Adam
* Learning Rate: 0.01
* Weight Decay: 5e-4
* Dropout: 0.6
* Epochs: 200
* Loss Function: Cross-Entropy Loss

The model is trained on a single GPU, with batch-level processing. Upon reproduction, we achieved test accuracy of 72.02% and a macro F1-score of 0.7207, which closely aligns with the original authors' reported performance. After performing additional hyperparameter tuning, we further improved the model’s test accuracy to 74.55%, demonstrating the reproducibility and flexibility of the architecture.

**5. Methodology**

To evaluate the effectiveness of the reproduced deep learning model, we use a set of metrics that provide a comprehensive understanding of classification performance. These include:

* Accuracy
* Precision
* Recall
* F1-score

The Precision, Recall, F1-score use Macro-averaging method, Macro-averaging ensures that each class contributes equally to the final metric, regardless of class imbalance, which is especially important in real-world datasets where some categories may be underrepresented.

**Proposed Model**

We highlight the most representative models from the UniGNN framework—UniGIN, UniSAGE, and include HyperGCN as a comparative baseline.

We evaluate several hypergraph neural network (HGNN) models as proposed in the original UniGNN paper, including UniGIN, UniSAGE, and HyperGCN. These models are designed to operate on hypergraphs by representing them as bipartite graphs that connect nodes to hyperedges. Under this unified message-passing framework, each model shares a common data flow: node features are linearly transformed, messages are aggregated through scatter operations between nodes and hyperedges, and the resulting representations are optionally normalized before being passed to the next layer.

UniGIN is based on the Graph Isomorphism Network (GIN) and is known for its strong theoretical expressiveness. In this adaptation, node and hyperedge features are aggregated via summation, and a learnable ε parameter is used to control the contribution of a node’s own features relative to its neighbors. This design enhances the model’s capacity to capture nuanced structural patterns.

UniSAGE follows the GraphSAGE approach, which aggregates neighborhood information through simple functions like mean or sum and then combines it with the node's own features. This method modified one of GraphSAGE’s multiple aggregating functions:



where the combining process is sum instead of concatenation. Here, {xj}j∈Ni can be naturally generalized to {he}j∈Ni. This model supports inductive learning and is particularly useful for generalizing unseen data during inference.

HyperGCN, which is not part of the UniGNN framework but included as a comparative baseline, handles hypergraphs by converting each hyperedge into a clique and then applying a conventional GCN. Although computationally straightforward, this method may lose the higher-order relational structure inherent in hypergraphs, which the UniGNN-based models aim to preserve more faithfully.

**Our Model**

We implemented two traditional machine learning models using the scikit-learn library: a Support Vector Machine (SVM) and a Random Forest classifier.

The SVM model employs a linear kernel and is configured to output probability estimates, making it suitable for multi-class classification. The Random Forest model uses 500 decision trees and is trained with bootstrap sampling. Both models operate on the same 500-dimensional TF-IDF input vectors used by the GNN model. Although these models do not exploit the underlying graph structure, they serve as useful baselines for assessing the importance of relational information in classification performance.

In addition, we implemented a Multi-Layer Perceptron (MLP) as a deep learning baseline that does not leverage any graph structure. The MLP consists of two hidden layers with ReLU activations and dropout for regularization. The output layer applies to a log-softmax function, and the model is trained using the Negative Log Likelihood Loss (NLLLoss). The training process uses a learning rate of 0.01and runs for 200 epochs. Although structurally simpler than the GNN model, the MLP enables a fair comparison by isolating the impact of graph-based learning from standard feature-based classification.

The proposed ModifiedGNN model, in contrast, integrates the topological structure of the citation network using both standard and hypergraph-based message passing. Its architecture captures interactions not just between neighboring nodes but also among groups of nodes via hyperedges. The inclusion of a residual gating mechanism allows the model to retain and combine useful information across multiple layers, enabling deeper architecture without performance degradation. This combination of design elements allows the model to effectively address the challenges posed by high-dimensional, sparse, and imbalanced data commonly encountered in citation networks.

**6. Experimental Results  
  
Proposed Model**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **Accuracy (%)** | **Precision (%)** | **Recall (%)** | **F1-score (%)** |
| HyperGCN | 70.25 ± 6.00 | 70.70 ± 11.71 | 70.42 ± 6.96 | 67.86 ± 9.66 |
| UniGIN | **71.75 ± 1.97** | **71.85 ± 1.85** | **73.53 ± 2.41** | **71.65 ± 2.11** |
| UniSAGE | 71.38 ± 3.37 | 71.71 ± 2.58 | 73.19 ± 2.59 | 71.33 ± 3.42 |

Table 1: Performance of proposed models on the PubMed dataset using Accuracy, Precision, Recall, and F1-score (mean ± standard deviation).

**Our Model**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **Accuracy (%)** | **Precision (%)** | **Recall (%)** | **F1-score (%)** |
| SVM | 70.57 ± 1.89 | 70.72 ± 1.95 | 72.89 ± 1.83 | 71.04 ± 1.96 |
| Random Forest | **73.27 ± 2.02** | **73.47 ± 2.22** | **76.14 ± 1.61** | **73.98 ± 1.90** |
| MLP | 67.07 ± 1.59 | 67.82 ± 1.53 | 68.41 ± 2.83 | 67.06 ± 2.04 |
| ModifiedGNN | 72.24 ± 1.51 | 72.66 ± 2.17 | 73.82 ± 1.51 | 72.49 ± 1.57 |

Table 2: Performance of our models on the PubMed dataset using Accuracy, Precision, Recall, and F1-score (mean ± standard deviation).

The proposal models—HyperGCN, UniGIN, and UniSAGE—exhibit competitive performance on the PubMed dataset. Among them, UniGIN achieves the highest accuracy (71.75%) and recall (73.53%), indicating strong capability in correctly identifying relevant classes. UniSAGE follows closely, demonstrating balanced performance across all metrics. HyperGCN, while slightly less stable with higher standard deviations, still maintains solid predictive ability, highlighting the effectiveness of hypergraph structures in document classification tasks.

Among the baseline models, Random Forest achieved the highest overall performance with an accuracy of 73.27% ± 2.02, precision of 73.47% ± 2.22, recall of 76.14% ± 1.61, and an F1-score of 73.98% ± 1.90. This suggests that Random Forest is particularly effective in handling the feature representations extracted from the PubMed dataset.

Our proposed deep learning-based model, ModifiedGNN, also performed competitively, achieving an accuracy of 72.24% ± 1.51 and an F1-score of 72.49% ± 1.57, surpassing all graph-based baseline models such as HyperGCN (70.25%), UniGIN (71.75%), and UniSAGE (71.38%). While UniGIN and UniSAGE show slightly higher recall, ModifiedGNN presents a more balanced trade-off across all metrics, indicating stronger generalization.

The SVM model, with an accuracy of 70.57% ± 1.89, performs on par with HyperGCN, but benefits from greater stability (lower standard deviation). The MLP, although simple, implemented the lowest performance among our models, with an accuracy of 67.07% ± 1.59, which confirms its limitations in capturing complex graph-structured data.

## 7. Discussion

From the experimental results, we can draw several important conclusions by comparing the deep learning models, the traditional machine learning models, and our ModifiedGNN.

Starting with the hypergraph neural network models, UniGIN came out as the best performer with an F1-score of 71.65%, closely followed by UniSAGE and HyperGCN. UniGIN is known for its strong theoretical expressiveness, and its performance here supports that. It sums up node and hyperedge features and has a learnable parameter that helps balance a node’s own features and its neighbors’ features, which seems to help in capturing complex patterns. UniSAGE, which uses mean aggregation, also performed well, but with slightly more variation in results. HyperGCN, on the other hand, converts each hyperedge into a clique and applies a standard GCN. While this approach is simpler and easier to implement, it may lose important higher-order relationships in the graph, which likely explains its slightly weaker performance.

Also, our model of the traditional machine learning models, the Random Forest performed surprisingly well, even better than many deep learning models. It achieved the highest accuracy (73.27%) and F1-score (73.98%) among all models. This shows that in some cases, with strong feature engineering (like TF-IDF in our case), traditional models can be very competitive. The SVM also performed well, with an F1-score of 71.04%, which is close to the UniGIN model. Even though these models don't use any graph structure, they still manage to learn meaningful patterns from the data.

On the deep learning side, the MLP model didn’t perform as well, with an accuracy 67.07% and F1-score of 67.06%. This is expected, since the MLP does not use any structural information from the graph—it only sees the input features. This makes the MLP a good baseline to understand how much benefit we get from using the graph.

Also, our ModifiedGNN model performed very well, achieving accuracy 72.24% and F1-score of 72.49%, which is higher than UniGNN models. What makes the ModifiedGNN stand out is that it combines standard graph message passing with hypergraph message passing. It also uses a residual gating mechanism, which helps it keep useful information across multiple layers. This allows the model to go deeper without losing performance. This is especially useful in sparse and high-dimensional datasets like PubMed, where information needs to travel through the graph more effectively.

An interesting insight from these results is that including the graph structure generally improves performance, but traditional models can still be strong baselines—especially when the feature representation is well-designed. Also, combining different types of graph structures, like in the ModifiedGNN (using both edges and hyperedges), can lead to even better results than using either alone.

Although the performance improvements achieved by ModifiedGNN, several limitations remain that warrant further discussion and investigation. One key observation is the relatively close accuracy between ModifiedGNN and the traditional Random Forest classifier. Although ModifiedGNN slightly outperforms UniGNN, its margin over Random Forest is narrow. This suggests that the PubMed feature space, constructed using TF-IDF representations, may already capture sufficient discriminative information for classification. So, the benefit of more complex hypergraph modeling is somewhat limited. Furthermore, the ensemble nature of Random Forest provides strong generalization even without explicitly modeling high-order structures, making it a surprisingly competitive baseline.

Another limitation lies in the sensitivity of the model to the hypergraph construction process. In our implementation, hyperedges are formed based on term co-occurrence across documents. However, this method may inadvertently introduce noisy or semantically weak connections, which can dilute the advantages of hypergraph-based message passing. The quality of hyperedges is therefore critical, and improper construction may reduce the model’s ability to learn meaningful patterns.

Furthermore, while we used macro-average metrics to handle the class imbalance problem, we did not study the per-class performance or confusion matrix, which might reveal whether the model is biased towards certain classes.

In future work, we consider integrating semantic-based hyperedge pruning techniques to retain only the most informative and relevant high-order connections. This could help reduce noise and enhance model performance. Additionally, incorporating attention-based mechanisms to assign weights to hyperedges may allow the model to learn which connections contribute more significantly to the classification task. Finally, Expanding the evaluation to include multi-label classification scenarios or testing on larger biomedical hypergraph datasets would provide a more comprehensive understanding of the model’s generalizability.

In summary, our experiments show that using graph and hypergraph structure in learning leads to better classification results. Models like UniGIN and ModifiedGNN are powerful tools for this kind of data. However, traditional models like Random Forest still provide a strong baseline and should not be overlooked. The ModifiedGNN especially shows promise as a flexible and effective model for handling sparse, high-dimensional, and imbalanced graph data.

## 8. Conclusion

In this project, we successfully reproduced and extended a hypergraph-based deep learning model, ModifiedGNN, for multi-class classification on the PubMed citation network dataset. The model integrates graph and hypergraph structures through a hybrid framework that utilizes multi-layer feature propagation. Our implementation shows that incorporating high-order relations captured by hypergraphs can significantly improve the representation power of node embeddings. In addition, the residual gated convolution mechanism facilitates better gradient flow and deeper feature learning.

We compare ModifiedGNN with traditional machine learning models such as SVM, random forest, and MLP, as well as baseline GNN. The results show that while traditional models perform fairly well with carefully designed features, ModifiedGNN consistently outperforms them in terms of classification accuracy. This highlights the advantage of leveraging the structural information inherent in citation networks. However, this is limited to the Published Dataset. It does not exclude that the performance of other datasets such as DBLP Proposed Model such as UniSAGE will be better than our ModifiedGNN model.

Our work demonstrates the feasibility and effectiveness of scaling GNNs using hypergraph modeling on large-scale academic datasets. Furthermore, the reproducibility of the results validates the robustness of the original model. Future work could explore incorporating attention mechanisms, experimenting with larger and more diverse datasets, and optimizing computational efficiency to enhance scalability.

## 9. Contributions

|  |  |
| --- | --- |
| **Name** | **Contribute score** |
| Lam Yam Ying | 33% |
| Kan Wai Yi | 33% |
| Siu Yau Shing | 33% |

## 10. Reference

[1] Y. Gao, Z. Zhang, H. Lin, X. Zhao, S. Du and C. Zou, "Hypergraph Learning: Methods and Practices," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 44, no. 5, pp. 2548-2566, 1 May 2022, doi: 10.1109/TPAMI.2020.3039374.

[2] X. Xia, H. Yin, J. Yu, Q. Wang, L. Cui, and X. Zhang, “Self-Supervised Hypergraph Convolutional Networks for Session-based Recommendation,” Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, no. 5, pp. 4503–4511, May 2021, doi: 10.1609/aaai.v35i5.16578.

‌[3] X. Gu, L. Chen, and M. Krenn, “Quantum experiments and hypergraphs: Multiphoton sources for quantum interference, quantum computation, and quantum entanglement,” Physical review. A/Physical review, A, vol. 101, no. 3, Mar. 2020, doi: 10.1103/physreva.101.033816.

[4] J. Huang and J. Yang, “UniGNN: a Unified Framework for Graph and Hypergraph Neural Networks,” arXiv (Cornell University), Aug. 2021, doi: 10.24963/ijcai.2021/353.

[5] Prithviraj Sen. “Papers with Code - Pubmed Dataset,” paperswithcode.com. https://paperswithcode.com/dataset/pubmed

[6] T. N. Kipf and M. Welling, “Semi-Supervised Classification with Graph Convolutional Networks,” 2016, doi: 10.48550/arxiv.1609.02907

[7] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, “Graph Attention Networks,” arXiv.org, 2018.

[8] W. L. Hamilton, R. Ying, and J. Leskovec, “Inductive Representation Learning on Large Graphs,” 2017, doi: 10.48550/arxiv.1706.02216

[9] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, “How Powerful are Graph Neural Networks?,” 2018, doi: 10.48550/arxiv.1810.00826

[10] Y. Feng, H. You, Z. Zhang, R. Ji, and Y. Gao, “Hypergraph Neural Networks,” 2018, doi: 10.48550/arxiv.1809.09401

‌[11] Naganand Yadati, Madhav Nimishakavi, P. Yadav, Vikram Nitin, A. Louis, and P. P. Talukdar, “HyperGCN: A New Method For Training Graph Convolutional Networks on Hypergraphs,” Neural Information Processing Systems, vol. 32, pp. 1509–1520, Jan. 2019.

‌[12] D. Arya, D. K. Gupta, S. Rudinac, and M. Worring, “HyperSAGE: Generalizing Inductive Representation Learning on Hypergraphs,” arXiv (Cornell University), Jan. 2020, doi: https://doi.org/10.48550/arxiv.2010.04558.

[13] M. Chen, Z. Wei, Z. Huang, B. Ding, and Y. Li, “Simple and Deep Graph Convolutional Networks,” 2020, doi: 10.48550/arxiv.2007.02133

‌