# Comparative Study of GCN and HGConv: Structure, Features, and Noise

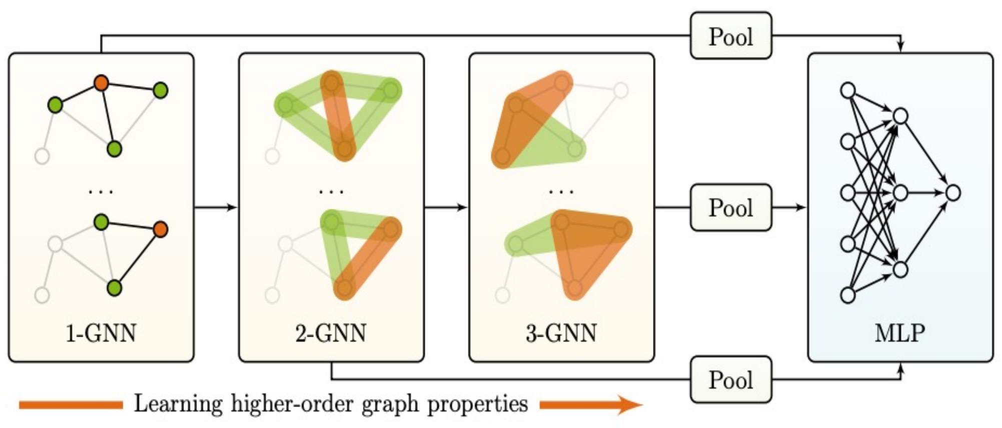
**Jaeyeol Lim[1]**

Rural Systems Engineering[1], Seoul National Unviersity

limlimlim00@snu.ac.kr

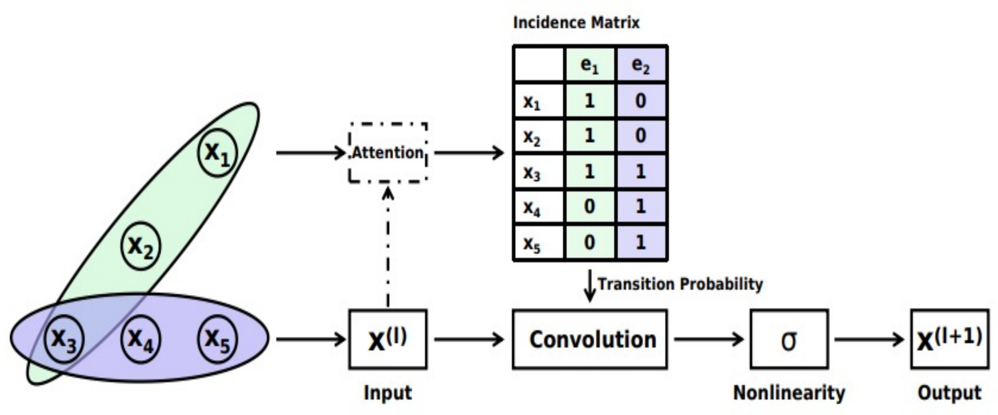
## Backgrounds

- **Graph Convolutional Network (GCN)** aggregates information from neighboring nodes via **pairwise edges**
- Known for its simplicity and interpretability, GCN performs well on **homophilic graphs** with clear structural patterns



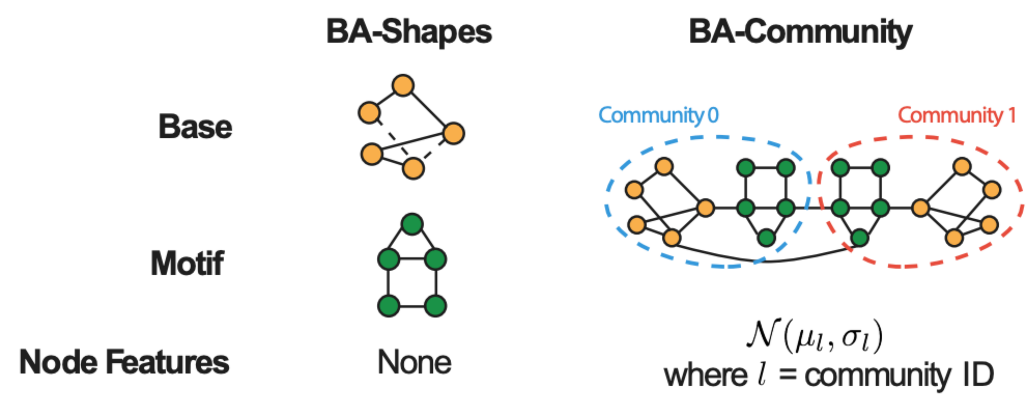

Morris, C., Ritzert, M., Fey, M., Hamilton, W. L., Lenssen, J. E., Rattan, G., & Grohe, M. (2019, July). Weisfeiler and leman go neural: Higher-order graph neural networks. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 33, No. 01, pp. 4602-4609).

- **HyperGraph Convolution (HGConv)** extends GCN by connecting multiple nodes via **hyperedges**, enabling richer interactions and higher-order relational learning
- Seen as a generalized form of GCN, it has gained attention in various applications.




Bai, S., Zhang, F., & Torr, P. H. (2021). Hypergraph convolution and hypergraph attention. *Pattern Recognition*, *110*, 107637.

- However, prior research lacks a direct and systematic comparison of GCN and HGConv in terms of:
  - **Structural generalization performance**
  - **Robustness** to feature noise, perturbation, label noise




Ying, Z., Bourgeois, D., You, J., Zitnik, M., & Leskovec, J. (2019). Gnnexplainer: Generating explanations for graph neural networks. *Advances in neural information processing systems*, *32*.

## Research Goals

- **Identify graph structures that support GCN and HGConv (1-hop, kNN)**
  - Analyze performance across varying structures
- **Evaluate the robustness of GCN and HGConv (1-hop)**
  - Test models under feature/label noise, and structural perturbations

## Methodology

- **Message Passing Architecture**
  - Graph Convolution

$$X^{l+1} = \sigma(\hat{A}X^l W)$$

  - HyperGraph Convolution

$$X^{l+1} = \sigma(D^{-\frac{1}{2}}HWB^{-1}H^T D^{-\frac{1}{2}}X^l P)$$

- **Hyperedge construction for HGConv**
  - 1-hop neighborhood grouping: Create a hyperedge for each node and its immediate neighbors

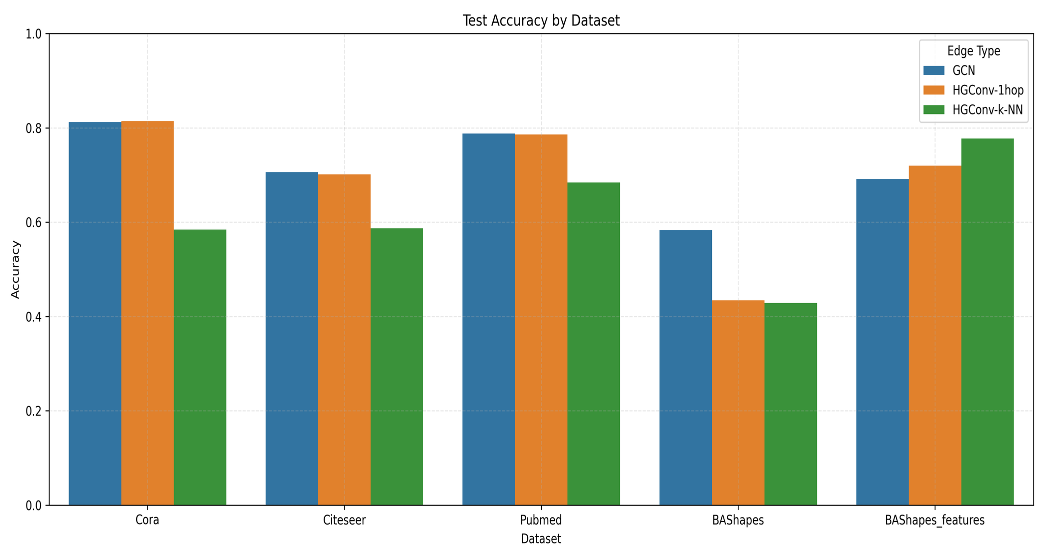
$$e_v = \{v\} \cup N(v) \quad where\ N(v) = \{u \in V | (v,u) \in E\}$$

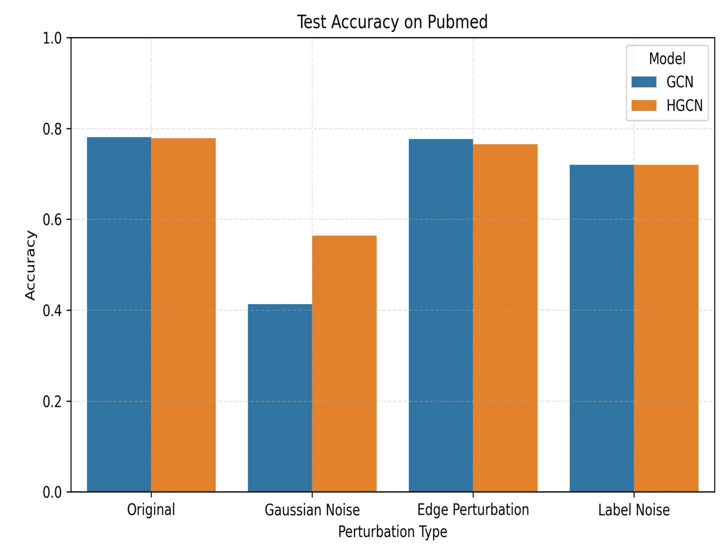
  - k-nearest neighbors grouping: Groups each node with its k most similar nodes based on cosine similarity

$$e_v = \{v\} \cup top_k(\{u \in V \setminus \{v\} \mid sim(X_v, X_u)\}$$

- **Perturbation settings for robustness evaluation**
  - Feature Noise: Gaussian Noise $N(0, \sigma^2)$ is added to input features to stimulate input corruption
  - Structural Perturbation: Randomly add or remove edges from the original graph structure to test the models' stability under topological changes
  - Label Noise: A fixed percentage of training node labels are randomly flipped to incorrect classes to assess classification robustness

## Experiments




- **Experimental Setup**
  - Constructed 2-layer models using implementations from PyTorch Geometric (PyG)
  - Evaluated on Cora, Citeseer, Pubmed, BAShapes, and a feature-augmented BAShapes dataset with class-wise shifted means
  - Gaussian noise sigma 0.1, Edge perturbation 5%, Label noise 10%
- **Key Results**
  - On BAShapes with features, the performance followed the order: HGConv-kNN > HGConv-1hop > GCN
  - Under feature noise, HGConv model (1-hop) exhibited enhanced robustness

## Concluding Remarks

- **HGConv-kNN achieves the best performance on BAShapes with features**
  - Highlights the importance of hyperedge design in structure-driven tasks
- **HGConv-1hop model exhibit enhanced robustness to feature-level noise**
  - Consistently outperform GCN in noisy environments across all datasets

## GitHub

- https://github.com/limlimlim00/FInal-Project_GCN-HGConv