

A Tutorial on Hypergraph Neural Networks: An In-Depth and Step-by-Step Guide Part 2. Input Features and Structures





Soo Yong Lee*



Yue Gao



Alessia Antelmi



Mirko Polato



Kijung Shin



Part 2. Inputs





The slides are available at <u>https://sites.google.com/view/hnn-tutorial</u>

Inputs of Hypergraph Neural Network

- Typical inputs of hypergraph neural networks (HNNs) are hypergraph structure and node (and/or hyperedge) feature vectors.
- The quality of the inputs can be critical for the effectiveness of HNNs.



Inputs of Hypergraph Neural Network (cont.)

- Key questions regarding HNN inputs are:
 - **Q1**) How can hypergraph structures be represented?
 - **Q2)** What input features can be used for nodes and hyperedges?



Inputs of Hypergraph Neural Network (cont.)

- Key questions regarding HNN inputs are:
 - **Q1**) How can hypergraph structures be represented?
 - **Q2**) What input features can be used for nodes and hyperedges?



Q1) Representing Hypergraph Structures

- We introduce five different ways to represent hypergraph structures:
 - Clique-, adaptive-, star-, line-, and tensor-expansions.



Q1) Representing Hypergraph Structures: Clique

- Clique expansion of a hypergraph is a homogeneous, pairwise graph.
 - Each hyperedge is converted into a clique of its member nodes.
 - Optionally, self-loops can be added.



Feng et al., Hypergraph Neural Networks, AAAI 2019

Q1) Representing Hypergraph Structures: Clique

- Clique expansion of a hypergraph is a homogeneous, pairwise graph.
 - Often, edges are weighted by the (normalized) count of hyperedges
 - Normalization considers the size of each hyperedge.
 - Feng et al. (2019) further weighed edges with learnable parameters.



Feng et al., Hypergraph Neural Networks, AAAI 2019

Q1) Representing Hypergraph Structures: Adaptive

- Adaptive expansion of a hypergraph is a homogeneous, pairwise graph.
 - Each hyperedge is converted into pairwise edge(s) via learnable rules.
 - Qian et al. (2023) created and weighed edges based on node features.



Q1) Representing Hypergraph Structures: Star

- **Star expansion** of a hypergraph is a bipartite, pairwise graph.
 - Each hyperedge is converted into a new node.
 - Each hyperedge (i.e., new node) and each of its member nodes are joined by a pairwise edge.



Chien et al., You Are Allset: A Multiset Function Framework for Hypergraph Neural Networks, ICLR 2022

Q1) Representing Hypergraph Structures: Line

- Line expansion of a hypergraph is a homogeneous, pairwise graph.
 - Each pair of a hyperedge and its node is converted into a new node.
 - Two new nodes are joined by a pairwise edge if they share a hyperedge or a node.



Q1) Representing Hypergraph Structures: Line

- Note that line expansion is different from a (k-)line graph
- *k*-Line graph of a hypergraph is a homogeneous, pairwise graph.
 - Each hyperedge is converted into a new node.
 - Two new nodes are joined by a pairwise edge if their corresponding

hyperedges share at least k common nodes.



Q1) Representing Hypergraph Structures: Tensor

- A hypergraph can be represented as a **tensor** *A*.
 - The order of the tensor equals the maximum hyperedge size
 - The dimensionality of each mode equals the node count.
 - Each tensor entry is non-zero if there exists a hyperedge containing



Wang et al. T-HyperGNNs: Hypergraph Neural Networks via Tensor Representations. IEEE TNNLS 2024.

Q1) Representing Hypergraph Structures (cont.)

- These methods are either **reductive** or **non-reductive**.
 - **Reductive** methods may incur information loss after transformation
 - However, they provide simple and straightforward graph structures.



Q1) Representing Hypergraph Structures (cont.)

- These methods are either **reductive** or **non-reductive**.
 - Non-reductive methods incur no information loss.
 - However, they are often more complex and, thus, difficult to handle.



Inputs of Hypergraph Neural Network (cont.)

- Key questions regarding hypergraph neural network (HNN) inputs are:
 - **Q1**) How can hypergraph structures be represented?
 - **Q2)** What input features can be used for nodes and hyperedges?



Q2) Node and Hyperedge Features

- Typical **input features** for HNNs are categorized into:
 - external, structural, and identity features.



Q2) Node and Hyperedge Features: External

- **External features** are additionally given information that is not directly derived from the input hypergraph structure.
- They complement structural information reflected in hypergraphs.



Q2) Node and Hyperedge Features: External (cont.)

- External node features in popular benchmark datasets include:
 - 1) bag-of-words vectors for article nodes, 2) visual object embeddings for image nodes, and 3) the counts of atoms for molecule nodes.



Q2) Node and Hyperedge Features: Structural

- **Structural features** are derived from the input hypergraph structure.
 - They typically capture structural proximity or similarity among nodes.
 - Structural features are either *local* or *global*.



Q2) Node and Hyperedge Features: Structural (cont.)

- Local structural features capture node-hyperedge membership.
 - Liu et al. (2024) leveraged the rows of the incidence matrix as part of their node features.



Q2) Node and Hyperedge Features: Structural (cont.)

- Global structural features capture proximity or similarity among nodes beyond direct connections.
- Zhang et al. (2020) and Do et al. (2024) leveraged random walks to capture such proximity and similarity, respectively.



Zhang et al., Hyper-SAGNN: A Self-attention Based Graph Neural Network for Hypergraphs, ICLR 2020 Do et al., Unsupervised Alignment of Hypergraphs with Different Scales, KDD 2024

Q2) Node and Hyperedge Features: Identity

- Identity features are uniquely assigned to each node and hyperedge.
- They encourage HNNs to distinguish different nodes and hyperedges.



Q2) Node and Hyperedge Features: Identity (cont.)

- Ji et al. (2020) "learned" identity features using an embedding layer.
 - Each node and hyperedge has a feature vector that is learned during training.



Q2) Node and Hyperedge Features (cont.)

- External, structural, and identity features offer distinct advantages:
 - External features can be highly informative, leading to performance gains.
 - However, obtaining high-quality external features can be costly.



Q2) Node and Hyperedge Features (cont.)

- External, structural, and identity features offer distinct advantages:
 - Structural features are readily obtainable from the input hypergraph.
 - However, their computation can be costly for large hypergraphs.



Q2) Node and Hyperedge Features (cont.)

- External, structural, and identity features offer distinct advantages:
 - Identity features are readily obtainable without much computational cost.
 - However, they may not be applicable to inductive settings where test hypergraphs are different from training hypergraphs.



Part 2 Summary

- Typical inputs of hypergraph neural networks (HNNs) are hypergraph structure and node (and/or hyperedge) feature vectors.
- The quality of the inputs can be critical for the effectiveness of HNNs.



Part 2 Summary (cont.)

- Key questions regarding HNN inputs are:
 - **Q1**) How can hypergraph structures be represented?
 - **Q2)** What input features can be used for nodes and hyperedges?



Part 2 Summary (cont.)

- We introduce five different ways to represent hypergraph structures:
 - Clique-, adaptive-, star-, line-, and tensor-expansions.



Part 2 Summary (cont.)

- We introduce three different types of **input features** for HNNs:
 - external, structural, and identity features.

