NCML
Neutrosophic Computing and Machine Learning, Vol. **39**, 2025
University of New Mexico

Multi-SuperHyperGraph Neural Networks: A Generalization of Multi-HyperGraph Neural Networks

Takaaki Fujita[1] *

[1] Independent Researcher, Shinjuku, Shinjuku-ku, Tokyo, Japan. Takaaki.fujita060@gmail.com

*Correspondence: Takaaki.fujita060@gmail.com

Received: *03 01, 2025;* Accepted: *08 22, 2025*

Abstract. Graph theory provides a mathematical framework for modeling relationships among entities via vertices (nodes) and edges [1, 2]. A *hypergraph* extends this framework by allowing *hyperedges* to connect any number of vertices, thereby capturing complex multi-way interactions [3]. The *SuperHyperGraph* concept generalizes hypergraphs further through iterated power-set constructions and has recently drawn significant research interest [4, 5].

Graph Neural Networks (GNNs) propagate and aggregate node features across graph topologies via learnable message-passing to capture structural context [6–8]. Extensions such as Hypergraph Neural Networks, SuperHyperGraph Neural Networks, Multigraph Neural Networks, and MultiHyperGraph Neural Networks have likewise been explored [9, 10].

In this paper, we introduce and analyze the *Multi n-SuperHyperGraph Neural Network*, a theoretical extension of SuperHyperGraph Neural Networks built upon Multi-SuperHyperGraph structures. We expect that this framework will stimulate further advances in the study and application of GNNs.

Keywords: Graph Neural Networks (GNNs), HyperGraph, SuperHyperGraph, Multigraph Neural Networks, MultiHyperGraph Neural Networks, Hypergraph Neural Networks, SuperHyperGraph Neural Networks

1. Preliminaries

This section introduces the basic concepts and terminology required for the developments in this paper. Throughout, all sets and structures are assumed to be finite. Unless otherwise specified, the parameter n denotes a nonnegative integer.

1.1. *SuperHyperGraph*

A *hypergraph* generalizes a classical graph by introducing *hyperedges* that may connect any number of vertices, not only two. This property makes hypergraphs well suited for modeling complex multiway relationships [11–15]. A *SuperHyperGraph* extends this concept further. Recently introduced and increasingly investigated in the literature [4,5,16–18], a SuperHyperGraph is obtained by iteratively applying the powerset operator to a base vertex set, thereby embedding recursive hierarchical structures into hypergraphs [19–21]. The formal definitions are presented below.

Definition 1.1 (Powerset [22])**.** Let S be a set. The *powerset* of S, denoted $\mathcal{P}(S)$, is the collection of all subsets of S:

$$\mathcal{P}(S) = \{ A \mid A \subseteq S \}.$$

In particular, $\emptyset \in \mathcal{P}(S)$ and $S \in \mathcal{P}(S)$.

Definition 1.2 (n-th Powerset)**.** (cf. [23–26])

Let H be a set. The hierarchy of iterated powersets of H, denoted $\mathcal{P}_n(H)$, is defined inductively as

$$\mathcal{P}_1(H) := \mathcal{P}(H), \qquad \mathcal{P}_{n+1}(H) := \mathcal{P}\big(\mathcal{P}_n(H)\big), \quad n \geq 1.$$

Hence, for the first few cases one obtains

$$\mathcal{P}_2(H) = \mathcal{P}(\mathcal{P}(H)), \qquad \mathcal{P}_3(H) = \mathcal{P}(\mathcal{P}(\mathcal{P}(H))).$$

Similarly, the n-th *nonempty powerset*, denoted $\mathcal{P}^*_n(H)$, is defined recursively by

$$\mathcal{P}^*_1(H) := \mathcal{P}^*(H), \qquad \mathcal{P}^*_{n+1}(H) := \mathcal{P}^*\big(\mathcal{P}^*_n(H)\big), \quad n \geq 1,$$

where $\mathcal{P}^*(H) = \mathcal{P}(H) \setminus \{\emptyset\}$.

Example 1.3 (Feature Selection with $\mathcal{P}_2(H)$ in Machine Learning)**.** Feature selection in machine learning chooses the most relevant input variables, reducing dimensionality, improving accuracy, and enhancing model interpretability (cf. [27–30]). Let $H = \{\text{age, income, education}\}$ be a set of features for a classification task.

- The first powerset $\mathcal{P}_1(H) = \mathcal{P}(H)$ contains all possible feature subsets, e.g. $\{\text{age}, \text{income}\}$, $\{\text{education}\}$, etc. This corresponds to conventional feature selection.
- The second powerset $\mathcal{P}_2(H) = \mathcal{P}(\mathcal{P}(H))$ contains collections of such feature subsets, e.g. $\big\{\{\text{age}, \text{income}\}, \{\text{education}\}\big\}$. This can be used in ensemble feature selection, where different subsets of features are grouped together to construct meta-models.

Example 1.4 (Model Architecture Search with $\mathcal{P}_3(H)$)**.** Let $H = \{\text{CNN, RNN, Transformer}\}$ be a set of candidate neural network components.

- The first powerset $\mathcal{P}_1(H)$ enumerates all possible model architectures that select a subset of components.
- The second powerset $\mathcal{P}_2(H)$ enumerates sets of such architectures, useful for defining search spaces in AutoML.

- The third powerset $\mathcal{P}_3(H) = \mathcal{P}(\mathcal{P}(\mathcal{P}(H)))$ then represents collections of model-architecture families, enabling higher-order reasoning in meta-learning or neural architecture search frameworks.

Definition 1.5 (Hypergraph [3,31])**.** A *hypergraph* $H = (V(H), E(H))$ consists of

- a nonempty set $V(H)$ of vertices, and
- a set $E(H) \subseteq \mathcal{P}(V(H))$ of hyperedges.

This paper considers only finite hypergraphs.

Definition 1.6 (n-SuperHyperGraph)**.** (cf. [5,18]) Let V_0 be a finite base set of vertices, and define the iterated powersets

$$\mathcal{P}^0(V_0) := V_0, \qquad \mathcal{P}^{k+1}(V_0) := \mathcal{P}\big(\mathcal{P}^k(V_0)\big) \quad (k \geq 0).$$

An *n-SuperHyperGraph* is a pair

$$\mathrm{SHG}^{(n)} = (V, E),$$

where

$$V \subseteq \mathcal{P}^n(V_0), \qquad E \subseteq \mathcal{P}(V).$$

The elements of V are called *n-supervertices*, and the elements of E are called *n-superedges*. The condition $E \subseteq \mathcal{P}(V)$ ensures that every n-superedge is a subset of the n-supervertex set V, preserving the incidence relation between vertices and edges as in graphs and hypergraphs.

Example 1.7 (2-SuperHyperGraph)**.** Let the base set be

$$V_0 = \{a, b\}.$$

Then

$$P^1(V_0) = \{\emptyset, \{a\}, \{b\}, \{a, b\}\}, \qquad P^2(V_0) = \mathcal{P}\big(P^1(V_0)\big).$$

Choose the set of 2-supervertices

$$V = \Big\{ v_1 = \{\{a\}\}, \quad v_2 = \{\{b\}, \{a, b\}\} \Big\} \subseteq P^2(V_0),$$

and the set of 2-superedges

$$E = \Big\{ e_1 = \{v_1, v_2\}, \quad e_2 = \{v_2\} \Big\} \subseteq \mathcal{P}(V).$$

Thus

$$\mathrm{SHG}^{(2)} = (V, E)$$

is a 2-SuperHyperGraph in which

- v_1 and v_2 are distinct 2-supervertices drawn from $P^2(V_0)$;
- e_1 and e_2 are 2-superedges, each a subset of the supervertex set V;
- all vertices and edges lie within $P^2(V_0)$, illustrating the hierarchical construction.

1.2. *Multi n-SuperHyperGraph*

A multigraph is a graph in which multiple edges connecting the same pair of vertices are allowed, enabling edge multiplicities [32,33]. A multihypergraph is a hypergraph variant where hyperedges, each potentially connecting any number of vertices, can appear repeatedly with multiplicities [34–36]. A Multi n-SuperHyperGraph generalizes hypergraphs by iteratively lifting vertices and edges into n-th powerset hierarchies, enabling supervertex and superedge multiplicities [17].

Definition 1.8 (MultiHypergraph)**.** (cf. [34, 35]) A *multihypergraph* is a triple

$$\mathcal{H} = (V, \mathcal{E}, \mu),$$

where

- V is a finite set of *vertices*,
- $\mathcal{E}$ is a (multi)set of nonempty subsets of V, called *hyperedges*,
- $\mu\colon \mathcal{E} \to \mathbb{N}_{>0}$ is a *multiplicity function*, assigning to each hyperedge $e \in \mathcal{E}$ the number of times it appears.

Equivalently, one may regard $\mathcal{E}$ itself as a multiset, in which each hyperedge e occurs with multiplicity $\mu(e)$.

Example 1.9 (MultiHypergraph)**.** Let the vertex set be

$$V = \{v_1, v_2, v_3\}.$$

Define the multiset of hyperedges

$$\mathcal{E} = \{\, e_1, e_1, e_2, e_3, e_3, e_3 \},$$

where

$$e_1 = \{v_1, v_2\}, \quad e_2 = \{v_2, v_3\}, \quad e_3 = \{v_1\}.$$

The multiplicity function μ is given by

$$\mu(e_1) = 2, \quad \mu(e_2) = 1, \quad \mu(e_3) = 3.$$

Thus the *multihypergraph*

$$\mathcal{H} = (V, \mathcal{E}, \mu)$$

has:

- three vertices v_1, v_2, v_3,
- one hyperedge $\{v_1, v_2\}$ appearing twice,
- one hyperedge $\{v_2, v_3\}$ appearing once,
- one hyperedge $\{v_1\}$ appearing three times.

Definition 1.10 (Multi n-SuperHyperGraph)**.** (cf. [17]) Let V_0 be a finite *base set* of vertices. For each integer $k \geq 0$, define the iterated powerset

$$P^0(V_0) := V_0, \qquad P^{k+1}(V_0) := \mathcal{P}\big(P^k(V_0)\big),$$

where $\mathcal{P}(\cdot)$ denotes the standard powerset operator. A *Multi n-SuperHyperGraph* is a triple

$$\mathrm{MSHG}^{(n)} = (V, E, \mu),$$

with

$$V \subseteq P^n(V_0), \qquad E \subseteq \mathcal{P}(V),$$

and a *multiplicity function*

$$\mu : E \longrightarrow \mathbb{N}$$

assigning to each n-superedge $e \in E$ a positive integer $\mu(e)$ indicating how many parallel occurrences of e are present. Elements of V are called *n-supervertices*; elements of E are called *n-superedges*. The incidence condition $E \subseteq \mathcal{P}(V)$ makes each n-superedge a subset of the n-supervertex set.

Remark 1.11. If $\mu(e) = 1$ for all $e \in E$, then $\mathrm{MSHG}^{(n)}$ reduces to an ordinary n-SuperHyperGraph.

Example 1.12 (Multi 2-SuperHyperGraph)**.** Let the base set be $V_0 = \{a, b\}$. Then

$$P^1(V_0) = \{\emptyset, \{a\}, \{b\}, \{a, b\}\}, \qquad P^2(V_0) = \mathcal{P}\big(P^1(V_0)\big).$$

Choose the 2-supervertices

$$v_1 = \{\{a\}\}, \qquad v_2 = \{\{b\}, \{a, b\}\},$$

and set

$$V = \{v_1, v_2\} \subseteq P^2(V_0).$$

Define 2-superedges *on* V

$$e_1 = \{v_1, v_2\}, \qquad e_2 = \{v_2\},$$

so that

$$E = \{e_1, e_2\} \subseteq \mathcal{P}(V),$$

and specify the multiplicity function

$$\mu(e_1) = 2, \qquad \mu(e_2) = 1.$$

Thus $\mathrm{MSHG}^{(2)} = (V, E, \mu)$ is a Multi 2-SuperHyperGraph in which edges are subsets of the supervertex set V, with e_1 occurring twice and e_2 once.

Example 1.13 (Multi 3-SuperHyperGraph)**.** Let the base set be $V_0 = \{a\}$. Then

$$P^1(V_0) = \{\emptyset, \{a\}\}, \qquad P^2(V_0) = \big\{\emptyset, \{\emptyset\}, \{\{a\}\}, \{\emptyset, \{a\}\}\big\},$$

and

$$P^3(V_0) = \mathcal{P}\big(P^2(V_0)\big).$$

Select two 3-supervertices

$$U_1 = \{\emptyset, \{\emptyset\}\}, \qquad U_2 = \{\{\{a\}\}, \{\emptyset, \{a\}\}\},$$

and set

$$V = \{U_1, U_2\} \subseteq P^3(V_0).$$

Define 3-superedges *on* V

$$E_1 = \{U_1, U_2\}, \qquad E_2 = \{U_2\},$$

so that

$$E = \{E_1, E_2\} \subseteq \mathcal{P}(V),$$

and assign multiplicities

$$\mu(E_1) = 2, \qquad \mu(E_2) = 4.$$

Then $\mathrm{MSHG}^{(3)} = (V, E, \mu)$ is a Multi 3-SuperHyperGraph with vertices $U_1, U_2 \in P^3(V_0)$ and edges taken as subsets of V, where E_1 occurs twice and E_2 four times.

1.3. *MultiHypergraph Neural Network*

Graph Neural Networks and Hypergraph Neural Networks have been the subject of extensive research across a multitude of publications [37–39]. A Multigraph Neural Network processes multiple graph instances via parallel graph convolutional layers, then aggregates their vertex embeddings into a unified representation [40–42]. A Hypergraph Neural Network generalizes GNNs by learning on hypergraphs, capturing higher-order relationships among groups of vertices [9, 43–47]. A MultiHypergraph Neural Network generalizes this by applying hypergraph convolution to several hypergraph structures in parallel, integrating both hyperedge and vertex features into a combined embedding [48–50].

Definition 1.14 (MultiHypergraph Neural Network)**.** (cf. [51]) Let $\big\{H_m = (V, \mathcal{E}_m)\big\}_{m=1}^{M}$ be a collection of M hypergraphs over the same vertex set V. Denote by

$$H_m \in \{0,1\}^{|V| \times |\mathcal{E}_m|} \quad \text{the incidence matrix of } H_m,$$

and let $X \in \mathbb{R}^{|V| \times F}$ be the matrix of input vertex features. Define for each m the *hypergraph Laplacian*

$$\widetilde{H}_m \;=\; D_{v,m}^{-\frac{1}{2}}\, H_m\, D_{e,m}^{-1}\, H_m^{\top}\, D_{v,m}^{-\frac{1}{2}}, \tag{1}$$

where $D_{v,m}$ and $D_{e,m}$ are the diagonal degree matrices of vertices and hyperedges respectively. A *MultiHypergraph Neural Network* with L layers is defined by the layerwise propagation

$$Z_m^{(\ell+1)} = \sigma\big(\widetilde{H}_m \, Z_m^{(\ell)} \, W_m^{(\ell)}\big), \quad Z_m^{(0)} = X, \tag{2}$$

for $\ell = 0, 1, \ldots, L-1$, where each $W_m^{(\ell)}$ is a learnable weight matrix and σ an activation function. Finally, the outputs from all hypergraphs are *fused* by

$$Z = \mathrm{AGG}\big(Z_1^{(L)}, \, Z_2^{(L)}, \, \ldots, \, Z_M^{(L)}\big), \tag{3}$$

where AGG is an aggregation operator (e.g. average or concatenation). This architecture processes multiple hypergraph structures in parallel and integrates their learned representations.

Example 1.15 (MultiHypergraph Neural Network)**.** Let the base vertex set be

$$V = \{v_1, \, v_2, \, v_3\}, \quad M = 2.$$

We define two hypergraphs on V:

$$H_1 : \, \mathcal{E}_1 = \{\{v_1, v_2\}, \, \{v_2, v_3\}\}, \quad H_2 : \, \mathcal{E}_2 = \{\{v_1, v_3\}, \, \{v_2\}\}.$$

Their incidence matrices (rows v_1, v_2, v_3; columns ordered as the hyperedges above) are

$$H_1 = \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{pmatrix}, \quad H_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

Compute the degree matrices:

$$D_{v,1} = \mathrm{diag}(1, 2, 1), \quad D_{e,1} = \mathrm{diag}(2, 2), \qquad D_{v,2} = \mathrm{diag}(2, 1, 1), \quad D_{e,2} = \mathrm{diag}(2, 1).$$

The normalized Laplacians (cf. (1)) are

$$\widetilde{H}_1 = D_{v,1}^{-\frac{1}{2}} \, H_1 \, D_{e,1}^{-1} \, H_1^{\top} \, D_{v,1}^{-\frac{1}{2}} \;\approx\; \begin{pmatrix} 0.5000 & 0.3536 & 0 \\ 0.3536 & 0.5000 & 0.3536 \\ 0 & 0.3536 & 0.5000 \end{pmatrix},$$

$$\widetilde{H}_2 = D_{v,2}^{-\frac{1}{2}} \, H_2 \, D_{e,2}^{-1} \, H_2^{\top} \, D_{v,2}^{-\frac{1}{2}} = \begin{pmatrix} 0.5000 & 0 & 0.5000 \\ 0 & 1.0000 & 0 \\ 0.5000 & 0 & 0.5000 \end{pmatrix}.$$

Choose input features

$$X = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{pmatrix},$$

use weight matrices $W_m^{(0)} = I$, and the identity activation $\sigma(x) = x$. Then one propagation layer (2) yields

$$Z_1^{(1)} = \widetilde{H}_1 X \approx \begin{pmatrix} 0.5000 & 0.3536 \\ 0.7071 & 0.8536 \\ 0.5000 & 0.8536 \end{pmatrix}, \quad Z_2^{(1)} = \widetilde{H}_2 X = \begin{pmatrix} 1.0000 & 0.5000 \\ 0.0000 & 1.0000 \\ 1.0000 & 0.5000 \end{pmatrix}.$$

Finally, fuse the two outputs by averaging (3):

$$Z = \tfrac{1}{2}\big(Z_1^{(1)} + Z_2^{(1)}\big) \approx \begin{pmatrix} 0.7500 & 0.4268 \\ 0.3536 & 0.9268 \\ 0.7500 & 0.6768 \end{pmatrix}.$$

This example illustrates a concrete forward pass of a MultiHypergraph Neural Network with two hypergraph structures.

1.4. *Undirected n-SuperHyperGraph Neural Network (n-SHGNN)*

The definition of the Undirected n-SuperHyperGraph Neural Network (n-SHGNN) is presented as follows [10].

Definition 1.16 (n-SuperHyperGraph Neural Network (n-SHGNN))**.** [10] Let $H^{(n)} = \big(V^{(n)}, E^{(n)}\big)$ be an n-SuperHyperGraph over a base vertex set V_0, and let

$$H' = \big(V_0,\, E'\big)$$

be its *Expanded Hypergraph*, where

$$E' = \Big\{\, e' \subseteq V_0 \;\Big|\; e' = \bigcup_{v \in e} v, \quad e \in E^{(n)} \Big\}.$$

Let

$$X \in \mathbb{R}^{|V_0| \times d}$$

be the input feature matrix whose i-th row $x_i \in \mathbb{R}^d$ is the feature vector of base vertex $v_i \in V_0$. Define:

- The incidence matrix $H' \in \{0,1\}^{|V_0| \times |E'|}$ with entries

$$H'_{ij} = \begin{cases} 1, & v_i \in e'_j, \\ 0, & \text{otherwise.} \end{cases}$$

- The diagonal vertex-degree matrix $D_V \in \mathbb{R}^{|V_0| \times |V_0|}$ and hyperedge-degree matrix $D_E \in \mathbb{R}^{|E'| \times |E'|}$ defined by

$$(D_V)_{ii} = \sum_{j=1}^{|E'|} H'_{ij}\, w(e'_j), \quad (D_E)_{jj} = \sum_{i=1}^{|V_0|} H'_{ij},$$

  where $w(e'_j) > 0$ is a learnable weight for hyperedge $e'_j \in E'$.

- A learnable hyperedge-weight matrix

$$W \in \mathbb{R}^{|E'|\times|E'|}, \qquad \Theta \in \mathbb{R}^{d\times c},$$

  and a non-linear activation $\sigma(\cdot)$ (e.g. ReLU).

Then one layer of the n-SHGNN is given by the convolution

$$Y = \sigma\big(D_V^{-1/2}\, H'\, W\, D_E^{-1}\, H'^{\top}\, D_V^{-1/2}\, X\, \Theta\big),$$

where $Y \in \mathbb{R}^{|V_0|\times c}$ is the updated feature matrix.

Example 1.17 (Concrete Undirected 2-SuperHyperGraph Neural Network)**.** Let the base vertex set be

$$V_0 = \{1, 2, 3\}, \qquad n = 2.$$

Define an n-SuperHyperGraph

$$H^{(2)} = \big(V^{(2)}, E^{(2)}\big)$$

by choosing

$$V^{(2)} = \big\{\{1,2\},\, \{2,3\}\big\}, \quad E^{(2)} = \big\{e_1 = \{\{1,2\}\},\, e_2 = \{\{2,3\}\}\big\}.$$

Its *expanded hypergraph* is

$$H' = \big(V_0,\, E'\big), \qquad E' = \{\,\{1,2\},\, \{2,3\}\}.$$

The incidence matrix $H' \in \{0,1\}^{3\times 2}$ (rows $1, 2, 3$; cols e'_1, e'_2) is

$$H' = \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{pmatrix}.$$

Assign learnable hyperedge weights

$$w(e'_1) = 1, \quad w(e'_2) = 2,$$

so that

$$D_V = \operatorname{diag}\big(H'\, w(E')\big) = \operatorname{diag}(1,\, 3,\, 2), \qquad D_E = \operatorname{diag}\big(H'^{\top}\mathbf{1}\big) = \operatorname{diag}(2,\, 2),$$

and form

$$W = \operatorname{diag}(1, 2).$$

Let the input feature vector be

$$X = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix},$$

choose a single output channel ($\Theta = 1$) and identity activation $\sigma(x) = x$. Then one layer of the 2-SHGNN computes

$$Y = D_V^{-\frac{1}{2}}\, H'\, W\, D_E^{-1}\, H'^{\top}\, D_V^{-\frac{1}{2}}\, X \;\approx\; \begin{pmatrix} 1.0774 \\ 2.5133 \\ 2.3165 \end{pmatrix}.$$

Thus each base-vertex's new feature is a weighted, normalized aggregation of its neighbors according to the 2-SuperHyperGraph structure.

2. Results: Multi-SuperHyperGraph Neural Networks

We now present the definition of the Multi-SuperHyperGraph Neural Network. This construction extends the n-SuperHyperGraph Neural Network by employing the Multi-SuperHyperGraph framework. As this is a theoretical extension, we envision future empirical studies on real datasets to assess its effectiveness.

Definition 2.1 (Multi n-SuperHyperGraph Neural Network)**.** Let $\{\mathrm{MSHG}_m^{(n)} = (V_m^{(n)}, E_m^{(n)}, \mu_m)\}_{m=1}^M$ be a collection of M Multi n-SuperHyperGraphs over the same base set V_0, each satisfying the incidence condition $E_m^{(n)} \subseteq \mathcal{P}(V_m^{(n)})$. For $k \geq 0$ define the level-to-base *flattening* map

$$\mathrm{flat}_0(X) := X, \qquad \mathrm{flat}_{k+1}(X) := \bigcup_{Y \in X} \mathrm{flat}_k(Y).$$

For each m and each $e \in E_m^{(n)}$, set

$$\exp_n(e) := \mathrm{flat}_{\,n-1}\Big(\bigcup_{v \in e} v\Big) \;\in\; \mathcal{P}(V_0),$$

and define the *expanded hypergraph*

$$\mathsf{H}'_m \;=\; \big(V_0,\ E'_m\big), \qquad E'_m \;:=\; \{\,\exp_n(e) \mid e \in E_m^{(n)}\,\}.$$

Let $H'_m \in \{0,1\}^{|V_0| \times |E_m^{(n)}|}$ be the incidence matrix of H'_m with columns indexed by $e \in E_m^{(n)}$ (i.e., the j-th column is the indicator of $\exp_n(e_j) \subseteq V_0$). Define

$$D_{V,m} = \mathrm{diag}\big(H'_m\, W_m\, \mathbf{1}\big), \qquad D_{E,m} = \mathrm{diag}\big(H'^{\top}_m \mathbf{1}\big), \qquad W_m = \mathrm{diag}\big(\mu_m(e)\big)_{e \in E_m^{(n)}}.$$

Given input features $X \in \mathbb{R}^{|V_0| \times F}$, a Multi n-SuperHyperGraph Neural Network with L layers computes, for each m and $\ell = 0, \ldots, L-1$,

$$Z_m^{(\ell+1)} = \sigma\Big(D_{V,m}^{-\frac{1}{2}}\, H'_m\, W_m\, D_{E,m}^{-1}\, H'^{\top}_m\, D_{V,m}^{-\frac{1}{2}}\, Z_m^{(\ell)}\, \Theta_m^{(\ell)}\Big), \qquad Z_m^{(0)} = X,$$

where each $\Theta_m^{(\ell)}$ is a learnable weight matrix and σ an activation. Finally, the per-graph outputs are fused by an aggregation operator AGG (e.g., mean or concatenation):

$$Z = \mathrm{AGG}\big(Z_1^{(L)},\, Z_2^{(L)},\, \ldots,\, Z_M^{(L)}\big).$$

Example 2.2 (Concrete Multi 1-SuperHyperGraph Neural Network)**.** Let $V_0 = \{1,2,3\}$, $n = 1$, $M = 2$. Define two Multi 1-SuperHyperGraphs (with $E_m^{(1)} \subseteq \mathcal{P}(V_m^{(1)})$):

$$\mathrm{MSHG}_1^{(1)}: \ V_1^{(1)} = \big\{\{1,2\},\ \{2,3\}\big\}, \quad E_1^{(1)} = \big\{\ \{\{1,2\}\},\ \{\{2,3\}\}\ \big\}, \quad \mu_1(\{\{1,2\}\}) = 1,\ \mu_1(\{\{2,3\}\}) = 2.$$

$$\mathrm{MSHG}_2^{(1)}: \ V_2^{(1)} = \big\{\{1\},\ \{2\},\ \{3\}\big\}, \quad E_2^{(1)} = \big\{\ \{\{1\},\{2\},\{3\}\}\ \big\}, \quad \mu_2(\{\{1\},\{2\},\{3\}\}) = 1.$$

For $n = 1$, $\exp_1(e) = \bigcup_{v \in e} v$, hence the expanded hyperedges are

$$E_1' = \big\{\{1,2\},\ \{2,3\}\big\}, \qquad E_2' = \big\{\{1,2,3\}\big\}.$$

Thus the incidence matrices (rows $1,2,3$; columns enumerate $E_m^{(1)}$) are

$$H_1' = \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{pmatrix}, \qquad H_2' = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix},$$

and

$$W_1 = \mathrm{diag}(1,2), \qquad W_2 = \mathrm{diag}(1).$$

Degrees:

$$D_{V,1} = \mathrm{diag}(1,3,2), \quad D_{E,1} = \mathrm{diag}(2,2), \qquad D_{V,2} = \mathrm{diag}(1,1,1), \quad D_{E,2} = \mathrm{diag}(3).$$

With

$$X = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{pmatrix}, \quad \Theta_1^{(0)} = \Theta_2^{(0)} = I_2, \quad \sigma = \mathrm{id},$$

one layer yields

$$Z_1^{(1)} \approx \begin{pmatrix} 0.5000 & 0.2887 \\ 0.6969 & 0.9082 \\ 0.5000 & 0.9082 \end{pmatrix}, \qquad Z_2^{(1)} = \frac{1}{3} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} X = \begin{pmatrix} \frac{2}{3} & \frac{2}{3} \\ \frac{2}{3} & \frac{2}{3} \\ \frac{2}{3} & \frac{2}{3} \end{pmatrix}.$$

Fuse by averaging:

$$Z = \tfrac{1}{2}\big(Z_1^{(1)} + Z_2^{(1)}\big) \approx \begin{pmatrix} 0.5833 & 0.4777 \\ 0.6818 & 0.7875 \\ 0.5833 & 0.7875 \end{pmatrix}.$$

Example 2.3 (Concrete Multi 1-SuperHyperGraph Neural Network with $M = 3$)**.** Let $V_0 = \{1,2,3\}$, $n = 1$, $M = 3$. Define

$$\mathrm{MSHG}_1^{(1)}: \ V_1^{(1)} = \big\{\{1\},\{2\},\{3\}\big\},\ E_1^{(1)} = \big\{\ \{\{1\},\{2\}\},\ \{\{2\},\{3\}\}\ \big\},\ \mu_1 = (2,1).$$

$$\mathrm{MSHG}_2^{(1)}: \ V_2^{(1)} = \{\{1\},\{2\},\{3\}\}, \ E_2^{(1)} = \{\,\{\{1\},\{2\}\}, \ \{\{1\},\{3\}\}, \ \{\{2\},\{3\}\}\,\}, \ \mu_2 \equiv 1.$$

$$\mathrm{MSHG}_3^{(1)}: \ V_3^{(1)} = \{\{1\},\{2\},\{3\}\}, \ E_3^{(1)} = \{\,\{\{1\}\}, \ \{\{2\},\{3\}\}\,\}, \ \mu_3 = (1,2).$$

Then

$$H_1' = \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{pmatrix}, \quad H_2' = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}, \quad H_3' = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{pmatrix},$$

with $W_1 = \mathrm{diag}(2,1), \ W_2 = I_3, \ W_3 = \mathrm{diag}(1,2)$. With $D_{V,m} = \mathrm{diag}(H_m' W_m \mathbf{1}), \ D_{E,m} = \mathrm{diag}(H_m'^{\top}\mathbf{1})$, input $X = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{pmatrix}$, $\Theta_m^{(0)} = I_2, \ \sigma = \mathrm{id}$, one layer gives

$$Z_1^{(1)} \approx \begin{pmatrix} 0.5000 & 0.4082 \\ 0.6969 & 0.7887 \\ 0.5000 & 0.7887 \end{pmatrix}, \quad Z_2^{(1)} \approx \begin{pmatrix} 0.7500 & 0.5000 \\ 0.5000 & 0.7500 \\ 0.7500 & 0.7500 \end{pmatrix}, \quad Z_3^{(1)} = \begin{pmatrix} 1.0000 & 0.0000 \\ 0.5000 & 1.0000 \\ 0.5000 & 1.0000 \end{pmatrix}.$$

Averaging yields

$$Z = \tfrac{1}{3}\big(Z_1^{(1)} + Z_2^{(1)} + Z_3^{(1)}\big) \approx \begin{pmatrix} 0.7500 & 0.3027 \\ 0.5656 & 0.8462 \\ 0.5833 & 0.8462 \end{pmatrix}.$$

Example 2.4 (Concrete Multi 2-SuperHyperGraph Neural Network)**.** Let $V_0 = \{1,2\}$, $n = 2$, $M = 2$. Define

$$\mathrm{MSHG}_1^{(2)}: \ V_1^{(2)} = \{v_1 = \{\{1\}\}, \ v_2 = \{\{2\},\{1,2\}\}\}, \quad E_1^{(2)} = \{\,\{v_1, v_2\}, \ \{v_2\}\,\}, \quad \mu_1 = (1,2).$$

$$\mathrm{MSHG}_2^{(2)}: \ V_2^{(2)} = \{u_1 = \{\{1\}\}, \ u_2 = \{\{2\}\}\}, \quad E_2^{(2)} = \{\,\{u_1\}, \ \{u_2\}\,\}, \quad \mu_2 = (1,1).$$

Here $\exp_2(\{v_1, v_2\}) = \{1,2\}$ and $\exp_2(\{v_2\}) = \{1,2\}$; thus

$$H_1' = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}, \qquad H_2' = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix},$$

with $W_1 = \mathrm{diag}(1,2)$, $W_2 = \mathrm{diag}(1,1)$. Degrees:

$$D_{E,1} = \mathrm{diag}(2,2), \quad D_{V,1} = \mathrm{diag}(3,3), \qquad D_{E,2} = \mathrm{diag}(1,1), \quad D_{V,2} = \mathrm{diag}(1,1).$$

With $X = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$, $\Theta_1^{(0)} = \Theta_2^{(0)} = (1)$, $\sigma = \mathrm{id}$,

$$Z_1^{(1)} = \begin{pmatrix} 1.5 \\ 1.5 \end{pmatrix}, \qquad Z_2^{(1)} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}, \qquad Z = \tfrac{1}{2}\big(Z_1^{(1)} + Z_2^{(1)}\big) = \begin{pmatrix} 1.25 \\ 1.75 \end{pmatrix}.$$

Example 2.5 (Concrete Multi 3-SuperHyperGraph Neural Network)**.** Let $V_0 = \{1, 2\}$, $n = 3$, $M = 2$. Define

$$\mathrm{MSHG}_1^{(3)}: \; V_1^{(3)} = \{p_1 = \{\{1\}\},\, p_2 = \{\{2\}\},\, p_3 = \{\emptyset\}\}, \quad E_1^{(3)} = \{\{p_1, p_2\},\, \{p_1, p_3\}\}, \quad \mu_1 = (1, 2).$$

$$\mathrm{MSHG}_2^{(3)}: \; V_2^{(3)} = \{q_1 = \{\{1, 2\}\},\, q_2 = \{\{2\}\},\, q_3 = \{\emptyset\}\}, \quad E_2^{(3)} = \{\{q_1, q_3\},\, \{q_2, q_3\}\}, \quad \mu_2 = (3, 1).$$

Then

$$E_1' = \{\{1, 2\},\ \{1\}\}, \qquad E_2' = \{\{1, 2\},\ \{2\}\},$$

so

$$H_1' = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}, \qquad H_2' = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}, \quad W_1 = \mathrm{diag}(1, 2), \quad W_2 = \mathrm{diag}(3, 1).$$

Degrees:

$$D_{E,1} = \mathrm{diag}(2, 1), \quad D_{V,1} = \mathrm{diag}(3, 1), \qquad D_{E,2} = \mathrm{diag}(2, 1), \quad D_{V,2} = \mathrm{diag}(3, 4).$$

With $X = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$, $\Theta_m^{(0)} = I_2$, $\sigma = \mathrm{id}$,

$$Z_1^{(1)} \approx \begin{pmatrix} 0.8333 & 0.2887 \\ 0.2887 & 0.5000 \end{pmatrix}, \qquad Z_2^{(1)} \approx \begin{pmatrix} 0.5000 & 0.4330 \\ 0.4330 & 0.6250 \end{pmatrix},$$

and the average

$$Z = \tfrac{1}{2}\big(Z_1^{(1)} + Z_2^{(1)}\big) \approx \begin{pmatrix} 0.6667 & 0.3608 \\ 0.3608 & 0.5625 \end{pmatrix}.$$

Theorem 2.6. *The Multi n-SuperHyperGraph Neural Network generalizes both*

(1) *the MultiHypergraph Neural Network ($n = 0$), and*

(2) *the n-SuperHyperGraph Neural Network ($M = 1$).*

Proof. If $n = 0$, then flat_{-1} is vacuous and $\exp_0(e) = e$, so the construction reduces to the MultiHypergraph case. If $M = 1$, the aggregation is the identity and the update rule coincides with that of the n-SuperHyperGraph Neural Network. □

3. Conclusion

In this paper, we introduced and analyzed the *Multi n-SuperHyperGraph Neural Network*, a theoretical extension of SuperHyperGraph Neural Networks based on Multi-SuperHyperGraph structures. We anticipate that this framework will stimulate further developments in the study and application of Graph Neural Networks. For future work, we aim to extend the framework to encompass Directed Graph Neural Networks [39, 52–55], Dynamic Graph Neural Networks [56–59], Fuzzy Graph Neural Networks [7, 8, 38, 60], and Neutrosophic Graph Neural Networks [10, 61], including the design of their corresponding algorithms and the implementation of quantitative analyses on benchmark datasets. Moreover, we plan to explore extensions of the concepts discussed in this paper using HyperFuzzy Sets [62–64], Hesitant Fuzzy Sets [65, 66], Quadripartitioned Neutrosophic Sets [67–69], MetaStructure [70,71], Picture Fuzzy Set [72–74], and Plithogenic Sets [75–78], together with the design of suitable algorithms and quantitative evaluations on real data.

Funding

This study did not receive any financial or external support from organizations or individuals.

Acknowledgments

We extend our sincere gratitude to everyone who provided insights, inspiration, and assistance throughout this research. We particularly thank our readers for their interest and acknowledge the authors of the cited works for laying the foundation that made our study possible. We also appreciate the support from individuals and institutions that provided the resources and infrastructure needed to produce and share this paper. Finally, we are grateful to all those who supported us in various ways during this project.

Author Contributions

The paper has been solely authored by the corresponding author at this stage.

Data Availability

This research is purely theoretical, involving no data collection or analysis. We encourage future researchers to pursue empirical investigations to further develop and validate the concepts introduced here.

Ethical Considerations

This work does not involve any experiments or studies involving human participants or animals, and therefore no ethical approvals were required.

Conflicts of Interest

The authors confirm that there are no conflicts of interest related to the research or its publication.

Research Integrity

The authors hereby confirm that, to the best of their knowledge, this manuscript is their original work, has not been published in any other journal, and is not currently under consideration for publication elsewhere at this stage.

Disclaimer (Note on Computational Tools)

No computer-assisted proof, symbolic computation, or automated theorem proving tools (e.g., Mathematica, SageMath, Coq, etc.) were used in the development or verification of the results presented in this paper. All proofs and derivations were carried out manually and analytically by the authors.

Disclaimer (Limitations and Claims)

The theoretical concepts presented in this paper have not yet been subject to practical implementation or empirical validation. Future researchers are invited to explore these ideas in applied or experimental settings. Although every effort has been made to ensure the accuracy of the content and the proper citation of sources, unintentional errors or omissions may persist. Readers should independently verify any referenced materials.

To the best of the authors' knowledge, all mathematical statements and proofs contained herein are correct and have been thoroughly vetted. Should you identify any potential errors or ambiguities, please feel free to contact the authors for clarification.

The results presented are valid only under the specific assumptions and conditions detailed in the manuscript. Extending these findings to broader mathematical structures may require additional research. The opinions and conclusions expressed in this work are those of the authors alone and do not necessarily reflect the official positions of their affiliated institutions.

References

[1] Reinhard Diestel. Graduate texts in mathematics: Graph theory.

[2] Jonathan L Gross, Jay Yellen, and Mark Anderson. *Graph theory and its applications*. Chapman and Hall/CRC, 2018.

[3] Claude Berge. *Hypergraphs: combinatorics of finite sets*, volume 45. Elsevier, 1984.

[4] Takaaki Fujita and Florentin Smarandache. Fundamental computational problems and algorithms for superhypergraphs. *HyperSoft Set Methods in Engineering*, 3:32–61, 2025.

[5] Florentin Smarandache. *Extension of HyperGraph to n-SuperHyperGraph and to Plithogenic n-SuperHyperGraph, and Extension of HyperAlgebra to n-ary (Classical-/Neutro-/Anti-) HyperAlgebra*. Infinite Study, 2020.

[6] Boyu Du, Jingya Zhou, Ling Liu, and Xiaolong She. Fl-gnn: Efficient fusion of fuzzy neural network and graph neural network. In *ECAI 2024*, pages 1768–1775. IOS Press, 2024.

[7] Dalibor Krleža and Krešimir Fertalj. Graph matching using hierarchical fuzzy graph neural networks. *Ieee transactions on fuzzy systems*, 25(4):892–904, 2016.

[8] Haotian Chen and Jialiang Xie. Eeg-based tsk fuzzy graph neural network for driver drowsiness estimation. *Information Sciences*, 679:121101, 2024.

[9] Yifan Feng, Haoxuan You, Zizhao Zhang, Rongrong Ji, and Yue Gao. Hypergraph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 3558–3565, 2019.

[10] Takaaki Fujita and Florentin Smarandache. *Superhypergraph neural networks and plithogenic graph neural networks: Theoretical foundations*. Infinite Study, 2025.

[11] Song Feng, Emily Heath, Brett Jefferson, Cliff Joslyn, Henry Kvinge, Hugh D Mitchell, Brenda Praggastis, Amie J Eisfeld, Amy C Sims, Larissa B Thackray, et al. Hypergraph models of biological networks to identify genes critical to pathogenic viral response. *BMC bioinformatics*, 22(1):287, 2021.

[12] Xiaowei Liao, Yong Xu, and Haibin Ling. Hypergraph neural networks for hypergraph matching. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1266–1275, 2021.

[13] Muhammad Akram and Gulfam Shahzadi. Hypergraphs in m-polar fuzzy environment. *Mathematics*, 6(2):28, 2018.

[14] Georg Gottlob, Nicola Leone, and Francesco Scarcello. Hypertree decompositions and tractable queries. In *Proceedings of the eighteenth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 21–32, 1999.

[15] Georg Gottlob, Nicola Leone, and Francesco Scarcello. Hypertree decompositions: A survey. In *Mathematical Foundations of Computer Science 2001: 26th International Symposium, MFCS 2001 Mariánské Lázne, Czech Republic, August 27–31, 2001 Proceedings 26*, pages 37–57. Springer, 2001.

[16] N. B. Nalawade, M. S. Bapat, S. G. Jakkewad, G. A. Dhanorkar, and D. J. Bhosale. Structural properties of zero-divisor hypergraph and superhypergraph over $\mathbb{Z}_n$: Girth and helly property. *Panamerican Mathematical Journal*, 35(4S):485–495, 2025.

[17] Takaaki Fujita and Florentin Smarandache. A concise study of some superhypergraph classes. *Neutrosophic Sets and Systems*, 77:548–593, 2024.

[18] Florentin Smarandache. n-superhypergraph and plithogenic n-superhypergraph. *Nidus Idearum*, 7:107–113, 2019.

[19] Masoud Ghods, Zahra Rostami, and Florentin Smarandache. Introduction to neutrosophic restricted superhypergraphs and neutrosophic restricted superhypertrees and several of their properties. *Neutrosophic Sets and Systems*, 50:480–487, 2022.

[20] Mohammad Hamidi and Mohadeseh Taghinezhad. *Application of Superhypergraphs-Based Domination Number in Real World*. Infinite Study, 2023.

[21] Florentin Smarandache. *Introduction to the n-SuperHyperGraph-the most general form of graph today.* Infinite Study, 2022.

[22] Thomas Jech. *Set theory: The third millennium edition, revised and expanded.* Springer, 2003.

[23] Florentin Smarandache. Foundation of superhyperstructure & neutrosophic superhyperstructure. *Neutrosophic Sets and Systems*, 63(1):21, 2024.

[24] Takaaki Fujita, Maisam Jdid, and Florentin Smarandache. Hyperfunctions and superhyperfunctions in linear programming: Foundations and applications. *International Journal of Neutrosophic Science*, 26(4):65–76, 2025.

[25] Ajoy Kanti Das, Rajat Das, Suman Das, Bijoy Krishna Debnath, Carlos Granados, Bimal Shil, and Rakhal Das. A comprehensive study of neutrosophic superhyper bci-semigroups and their algebraic significance. *Transactions on Fuzzy Sets and Systems*, 8(2):80, 2025.

[26] Florentin Smarandache. Extension of hyperalgebra to superhyperalgebra and neutrosophic superhyperalgebra (revisited). In *International Conference on Computers Communications and Control*, pages 427–432. Springer, 2022.

[27] Razieh Sheikhpour, Kamal Berahmand, Mehrnoush Mohammadi, and Hassan Khosravi. Sparse feature selection using hypergraph laplacian-based semi-supervised discriminant analysis. *Pattern Recognition*, 157:110882, 2025.

[28] Xiaoling Yang, Hongmei Chen, Tianrui Li, Jihong Wan, and Binbin Sang. Neighborhood rough sets with distance metric learning for feature selection. *Knowledge-Based Systems*, 224:107076, 2021.

[29] Isabelle M Guyon and André Elisseeff. An introduction to variable and feature selection. *J. Mach. Learn. Res.*, 3:1157–1182, 2003.

[30] Pablo A Estévez, Michel Tesmer, Claudio A Perez, and Jacek M Zurada. Normalized mutual information feature selection. *IEEE Transactions on neural networks*, 20(2):189–201, 2009.

[31] Alain Bretto. Hypergraph theory. *An introduction. Mathematical Engineering. Cham: Springer*, 1, 2013.

[32] WB Vasantha Kandasamy, K Ilanthenral, and Florentin Smarandache. *Subset Vertex Multigraphs and Neutrosophic Multigraphs for Social Multi Networks*. Infinite Study, 2019.

[33] Hiroshi Nagamochi, Takashi Shiraki, and Toshihide Ibaraki. Augmenting a submodular and posi-modular set function by a multigraph. *Journal of Combinatorial Optimization*, 5:175–212, 2001.

[34] Kelly J Pearson and Tan Zhang. The laplacian tensor of a multi-hypergraph. *Discrete Mathematics*, 338(6):972–982, 2015.

[35] Zhe Yang, Liangkui Xu, and Lei Zhao. Efbh: Collaborative filtering model based on multi-hypergraph encoder. *IEEE Transactions on Consumer Electronics*, 70(1):2939–2948, 2023.

[36] Le An, Xiaojing Chen, Songfan Yang, and Xuelong Li. Person re-identification by multi-hypergraph fusion. *IEEE transactions on neural networks and learning systems*, 28(11):2763–2774, 2016.

[37] Yue Gao, Yifan Feng, Shuyi Ji, and Rongrong Ji. Hgnn+: General hypergraph neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3):3181–3199, 2022.

[38] Xinyu Guo, Bingjie Tian, and Xuedong Tian. Hfgnn-proto: Hesitant fuzzy graph neural network-based prototypical network for few-shot text classification. *Electronics*, 11(15):2423, 2022.

[39] Yixuan He, Quan Gan, David Wipf, Gesine D Reinert, Junchi Yan, and Mihai Cucuringu. Gnnrank: Learning global rankings from pairwise comparisons via directed graph neural networks. In *international conference on machine learning*, pages 8581–8612. PMLR, 2022.

[40] Federico Monti, Michael Bronstein, and Xavier Bresson. Geometric matrix completion with recurrent multi-graph neural networks. *Advances in neural information processing systems*, 30, 2017.

[41] Ding Yao, Zhang Zhi-li, Zhao Xiao-feng, Cai Wei, He Fang, Cai Yao-ming, and Wei-Wei Cai. Deep hybrid: multi-graph neural network collaboration for hyperspectral image classification. *Defence Technology*, 23:164–176, 2023.

[42] Du Yin, Renhe Jiang, Jiewen Deng, Yongkang Li, Yi Xie, Zhongyi Wang, Yifan Zhou, Xuan Song, and Jedi S Shang. Mtmgnn: Multi-time multi-graph neural network for metro passenger flow prediction. *GeoInformatica*, 27(1):77–105, 2023.

[43] Wenjie Du, Shuai Zhang, Zhaohui Cai, Xuqiang Li, Zhiyuan Liu, Junfeng Fang, Jianmin Wang, Xiang Wang, and Yang Wang. Molecular merged hypergraph neural network for explainable solvation gibbs free energy prediction. *Research*, 8:0740, 2025.

[44] Qiang He, Yunting Bao, Hui Fang, Yuting Lin, and Hao Sun. Hhan: Comprehensive infectious disease source tracing via heterogeneous hypergraph neural network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 291–299, 2025.

[45] Jie Wang, Luohuang Wu, Hulin Kuang, and Jianxin Wang. Hybrid region and population hypergraph neural network for mild cognitive impairment detection. *Pattern Recognition*, 169:111864, 2026.

[46] Md Tanvir Alam, Chowdhury Farhan Ahmed, and Carson K Leung. Hyperedge anomaly detection with hypergraph neural network. *arXiv preprint arXiv:2412.05641*, 2024.

[47] Rongping Ye, Xiaobing Pei, Haoran Yang, and Ruiqi Wang. Hyperedge interaction-aware hypergraph neural network. *arXiv preprint arXiv:2401.15587*, 2024.

[48] Junjie Zhu, Xibin Zhao, Han Hu, and Yue Gao. Emotion recognition from physiological signals using multi-hypergraph neural networks. In *2019 IEEE International Conference on Multimedia and Expo (ICME)*, pages 610–615. IEEE, 2019.

[49] Liping Nong, Jie Peng, Wenhui Zhang, Jiming Lin, Hongbing Qiu, and Junyi Wang. Adaptive multi-hypergraph convolutional networks for 3d object classification. *IEEE Transactions on Multimedia*, 25:4842–4855, 2022.

[50] Ziang Li, Jie Wu, Guojing Han, Chi Ma, and Yuenai Chen. Multi-hypergraph neural network with fusion of location information for session-based recommendation. *IAENG International Journal of Applied Mathematics*, 53(4), 2023.

[51] Jing Huang, Xiaolin Huang, and Jie Yang. Residual enhanced multi-hypergraph neural network. In *2021 IEEE international conference on image processing (ICIP)*, pages 3657–3661. IEEE, 2021.

[52] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Skeleton-based action recognition with directed graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7912–7921, 2019.

[53] Yixuan He, Gesine Reinert, David Wipf, and Mihai Cucuringu. Robust angular synchronization via directed graph neural networks. *arXiv preprint arXiv:2310.05842*, 2023.

[54] Takaaki Fujita. Directed superhypergraph neural networks. *Authorea Preprints*, 2025.

[55] Guo Zhenyu and Zhang Wanhong. An efficient inference schema for gene regulatory networks using directed graph neural networks. In *2023 42nd Chinese Control Conference (CCC)*, pages 6829–6834. IEEE, 2023.

[56] Chenguang Song, Yiyang Teng, Yangfu Zhu, Siqi Wei, and Bin Wu. Dynamic graph neural network for fake news detection. *Neurocomputing*, 505:362–374, 2022.

[57] Takaaki Fujita. Superhypergraph neural network and dynamic superhypergraph neural network. *Authorea Preprints*, 2025.

[58] Dongqi Fu and Jingrui He. Sdg: A simplified and dynamic graph neural network. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2273–2277, 2021.

[59] Mingyu Guan, Anand Padmanabha Iyer, and Taesoo Kim. Dynagraph: dynamic graph neural networks at scale. In *Proceedings of the 5th ACM SIGMOD Joint International Workshop on Graph Data Management Experiences & Systems (GRADES) and Network Data Analytics (NDA)*, pages 1–10, 2022.

[60] Takaaki Fujita. Hyperfuzzy graph neural networks and hyperplithogenic graph neural networks: Theoretical foundations. *Preprint*, 2025.

[61] A Meenakshi, J Shivangi Mishra, Jeong Gon Lee, Antonios Kalampakas, and Sovan Samanta. Advanced risk prediction in healthcare: Neutrosophic graph neural networks for disease transmission. *Complex & Intelligent Systems*, 11(9):413, 2025.

[62] Jayanta Ghosh and Tapas Kumar Samanta. Hyperfuzzy sets and hyperfuzzy group. *Int. J. Adv. Sci. Technol*, 41:27–37, 2012.

[63] Seok-Zun Song, Seon Jeong Kim, and Young Bae Jun. Hyperfuzzy ideals in bck/bci-algebras. *Mathematics*, 5(4):81, 2017.

[64] Takaaki Fujita. *Advancing Uncertain Combinatorics through Graphization, Hyperization, and Uncertainization: Fuzzy, Neutrosophic, Soft, Rough, and Beyond*. Biblio Publishing, 2025.

[65] Vicenç Torra and Yasuo Narukawa. On hesitant fuzzy sets and decision. In *2009 IEEE international conference on fuzzy systems*, pages 1378–1382. IEEE, 2009.

[66] Zeshui Xu. *Hesitant fuzzy sets theory*, volume 314. Springer, 2014.

[67] M Balamurugan, Khalil H Hakami, Moin A Ansari, Anas Al-Masarwah, and K Loganathan. Quadri-polar fuzzy fantastic ideals in bci-algebras: A topsis framework and application. *European Journal of Pure and Applied Mathematics*, 17(4):3129–3155, 2024.

[68] R Radha, A Stanis Arul Mary, and Florentin Smarandache. Quadripartitioned neutrosophic pythagorean soft set. *International Journal of Neutrosophic Science (IJNS) Volume 14, 2021*, page 11, 2021.

[69] Satham Hussain, Jahir Hussain, Isnaini Rosyida, and Said Broumi. Quadripartitioned neutrosophic soft graphs. In *Handbook of Research on Advances and Applications of Fuzzy Sets and Logic*, pages 771–795. IGI Global, 2022.

[70] Takaaki Fujita. Metastructure, meta-hyperstructure, and meta-superhyperstructure, 2025. Preprint.

[71] Takaaki Fujita. Metahypergraphs, metasuperhypergraphs, and iterated metagraphs: Modeling graphs of graphs, hypergraphs of hypergraphs, superhypergraphs of superhypergraphs, and beyond, 2025.

[72] Bui Cong Cuong and Vladik Kreinovich. Picture fuzzy sets-a new concept for computational intelligence problems. In *2013 third world congress on information and communication technologies (WICT 2013)*, pages 1–6. IEEE, 2013.

[73] Sankar Das, Soumitra Poulik, and Ganesh Ghorai. Picture fuzzy ϕ-tolerance competition graphs with its application. *Journal of Ambient Intelligence and Humanized Computing*, 15(1):547–559, 2024.

[74] Waheed Ahmad Khan, Waqar Arif, Quoc Hung NGUYEN, Thanh Trung Le, and Hai Van Pham. Picture fuzzy directed hypergraphs with applications towards decision-making and managing hazardous chemicals. *IEEE Access*, 2024.

[75] Fazeelat Sultana, Muhammad Gulistan, Mumtaz Ali, Naveed Yaqoob, Muhammad Khan, Tabasam Rashid, and Tauseef Ahmed. A study of plithogenic graphs: applications in spreading coronavirus disease (covid-19) globally. *Journal of ambient intelligence and humanized computing*, 14(10):13139–13159, 2023.

[76] Florentin Smarandache. *Plithogenic set, an extension of crisp, fuzzy, intuitionistic fuzzy, and neutrosophic sets-revisited*. Infinite study, 2018.

[77] Prem Kumar Singh. Plithogenic set for multi-variable data analysis. *International Journal of Neutrosophic Science*, 2020.

[78] Prem Kumar Singh. *Intuitionistic Plithogenic Graph*. Infinite Study, 2022.