Fuzzy SuperHyperGraph Neural Network (F-SHGNN)

Takaaki Fujita [1] *
[1] Independent Researcher, Shinjuku, Shinjuku-ku, Tokyo, Japan.

Abstract

Graph theory investigates relationships among entities through mathematical structures composed of vertices (nodes) and edges (connections) [1]. A *hypergraph* generalizes the classical graph by introducing *hyperedges*, which can join any number of vertices rather than just two, thereby enabling the modeling of complex multi-way relationships [2]. Building on this, the concept of a *SuperHyperGraph* has been proposed as a further extension of hypergraphs and has recently become a subject of active research [3, 4].

Graph Neural Networks (GNNs) are among the most extensively studied frameworks in artificial intelligence [5, 6]. The HyperGraph Neural Network (HGNN) extends GNNs by leveraging the expressive power of hypergraphs to capture higher-order dependencies [7, 8]. More recently, SuperHyperGraph Neural Networks have begun to emerge as an additional generalization [9]. Furthermore, these models have been augmented with fuzzy logic, giving rise to Fuzzy Graph Neural Networks and Fuzzy HyperGraph Neural Networks(cf. [10]).

In this paper, we introduce and investigate the Fuzzy SuperHyperGraph Neural Network (F-SHGNN), a novel framework that integrates and extends SuperHyperGraph Neural Networks, Fuzzy Graph Neural Networks, and Fuzzy HyperGraph Neural Networks.

Keywords: HyperGraph, SuperHyperGraph, Graph Theory, Graph Neural Networks (GNNs), HyperGraph Neural Network (HGNN), Fuzzy Graph Neural Networks, Fuzzy HyperGraph Neural Network

1 Preliminaries

This section provides an introduction to the foundational concepts and definitions required for the discussions in this paper. Throughout this paper, all sets and structures are assumed to be finite. Unless otherwise stated, the symbol n denotes a non-negative integer.

1.1 SuperHyperGraph

A *hypergraph* generalizes a classical graph by introducing *hyperedges*, which can connect any number of vertices—not just two—making it suitable for modeling complex, multi-way relationships [11–13]. A *SuperHyperGraph* takes this concept even further. Recently introduced and actively studied in a growing body of literature [3, 4, 14–19], the SuperHyperGraph incorporates recursive structures into hypergraphs through iterated applications of the power set operation. Conceptually, a SuperHyperGraph can be viewed as a hierarchical generalization of a hypergraph, in which both vertices and hyperedges are drawn from higher-order powersets of a base vertex set. The formal definition is presented below.

Definition 1.1 (Powerset)**.** Let S be any 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. [9, 20])

The n-th powerset of a set H, denoted $P_n(H)$, is constructed iteratively. Beginning with the standard powerset, the process is defined as:

$$P_1(H) = P(H), \quad P_{n+1}(H) = P(P_n(H)), \quad \text{for } n \geq 1.$$

In a similar manner, the n-th non-empty powerset, represented as $P_n^*(H)$, is recursively defined as:

$$P_1^*(H) = P^*(H), \quad P_{n+1}^*(H) = P^*(P_n^*(H)).$$

Here, $P^*(H)$ refers to the powerset of H excluding the empty set.

Example 1.3 (Household Chore Scheduling via n-th Powerset)**.** Let the set of daily household tasks be

$$H = \{\text{Dust, Vacuum, Mopping}\}.$$

First powerset ($n = 1$) gives all possible daily chore-sets:

$$P_1(H) = \big\{\emptyset, \{\text{Dust}\}, \{\text{Vacuum}\}, \{\text{Mopping}\}, \{\text{Dust,Vacuum}\},$$

$$\{\text{Dust,Mopping}\}, \{\text{Vacuum,Mopping}\}, \{\text{Dust,Vacuum,Mopping}\}\big\}.$$

For example, $\{\text{Dust,Vacuum}\} \in P_1(H)$ represents "Dust and Vacuum scheduled on a single day."

Second powerset ($n = 2$) yields all collections of daily schedules—i.e. a multi-day plan:

$$P_2(H) = P\big(P_1(H)\big).$$

One element of $P_2(H)$ is

$$\big\{\{\text{Dust,Vacuum}\}, \{\text{Mopping}\}, \{\text{Dust,Vacuum,Mopping}\}\big\},$$

which encodes a three-day cleaning plan:

- Day 1: Dust and Vacuum,
- Day 2: Mopping,
- Day 3: Dust, Vacuum, and Mopping.

Third powerset ($n = 3$) produces collections of multi-day plans—e.g. a seasonal schedule:

$$P_3(H) = P\big(P_2(H)\big).$$

For instance,

$$\big\{\{\{\text{Dust}\}\}, \{\{\text{Vacuum,Mopping}\}, \{\emptyset\}\}\big\} \in P_3(H)$$

could represent two weeks of plans, where in one week only Dust is done each day, and in another week there is a day with no chores and a day with Vacuum & Mopping.

Thus, the n-th powerset $P_n(H)$ models hierarchical scheduling:

- $P_1(H)$: daily task-sets,
- $P_2(H)$: multi-day (e.g. weekly) plans,
- $P_3(H)$: collections of weekly plans (e.g. monthly or seasonal schedules),
- and so on.

Definition 1.4 (Hypergraph [2, 21])**.** A *hypergraph* $H = (V(H), E(H))$ is a pair where:

- $V(H)$: A non-empty set of vertices.
- $E(H)$: A set of hyperedges, each of which is a subset of $V(H)$.

This paper focuses exclusively on finite hypergraphs.

Definition 1.5 (n-SuperHyperGraph)**.** [4,22] Let V_0 be a finite *base set* of vertices. For each $k \geq 0$, define the iterative powerset $\mathcal{P}^k(V_0)$ by

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

where $\mathcal{P}(\cdot)$ denotes the power set. An *n-SuperHyperGraph* is a pair

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

with

$$V \;\subseteq\; \mathcal{P}^n(V_0) \quad \text{and} \quad E \;\subseteq\; \mathcal{P}^n(V_0).$$

Each element of V is an *n-supervertex*, and each element of E is an *n-superedge*.

Example 1.6 (2-SuperHyperGraph Modeling Cross-Department Projects)**.** Let the base vertex set be

$$V_0 = \{\text{Alice, Bob, Carol, Dave}\}, \quad n = 2.$$

We form the nonempty 2nd-powerset

$$P_2^*(V_0) = \Big\{\{\{\text{Alice}\}, \{\text{Bob}\}\},\ \{\{\text{Carol}\}, \{\text{Dave}\}\},\ \{\{\text{Alice}\}, \{\text{Bob}\}, \{\{\text{Alice}\}, \{\text{Bob}\}\}\},\ \{\{\text{Carol}\}, \{\text{Dave}\}, \{\{\text{Carol}\}, \{\text{Dave}\}\}\}\Big\}$$

Choose the 2-SuperHyperGraph

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

with

$$V^{(2)} = \big\{u_1 = \{\{\text{Alice}\}, \{\text{Bob}\}\},\, u_2 = \{\{\text{Carol}\}, \{\text{Dave}\}\}\big\},$$
$$E^{(2)} = \big\{e_1 = \{\, u_1,\, u_2\},\quad e_2 = \{\,\{\{\text{Alice}\}\},\, \{\{\text{Bob}\}\},\, \{\{\text{Carol}\}\}\}\big\}.$$

Here:

- Each *2-supervertex* u_1 represents the "AB-Team" {Alice, Bob}, and u_2 the "CD-Team" {Carol, Dave}.
- Hyperedge e_1 models a *cross-department initiative* requiring both AB-Team and CD-Team.
- Hyperedge e_2 models an *ad-hoc working group* of individual contributors Alice, Bob, and Carol.

Thus $\mathrm{SHT}^{(2)} = (V^{(2)}, E^{(2)})$ is a concrete 2-SuperHyperGraph describing both team-level and cross-team collaborations in an organization.

1.2 Fuzzy n-Superhypergraph

A *fuzzy set* assigns to each element of a universe a membership degree in the interval $[0, 1]$ [23,24]. A *fuzzy graph* extends this concept by equipping both vertices and edges of a graph with membership degrees [25–27]. A *fuzzy hypergraph* further generalizes fuzzy graphs by allowing membership degrees on vertices and hyperedges [28–31]. A *fuzzy n-superhypergraph* extends the idea to n-superhypergraphs, assigning membership degrees to n-supervertices and n-superedges. The formal definition of a fuzzy n-superhypergraph is given as follows (cf. [4,19]).

Definition 1.7 (Fuzzy Hypergraph)**.** [31–33] A *fuzzy hypergraph* $G = (V, E, \psi, w)$ is a hypergraph where vertices have fuzzy membership degrees in hyperedges, and each hyperedge has an associated weight. The fuzzy hypergraph is defined as follows:

- V is the set of vertices.
- E is the set of hyperedges, where each hyperedge $e \in E$ is a subset of V.
- $\psi \in [0, 1]^{|E|\times|V|}$ is a matrix where ψ_{ei} represents the degree of membership of vertex $i \in V$ in hyperedge $e \in E$, satisfying $\sum_{i\in V} \psi_{ei} = 1$ for each $e \in E$ and $\sum_{e\in E} \psi_{ei} > 0$ for each $i \in V$.
- $w : E \longrightarrow \mathbb{R}_+$ assigns a positive weight $w(e)$ to each hyperedge $e \in E$.

Here, the matrix ψ serves as the incidence matrix of the fuzzy hypergraph, where each hyperedge quantifies the participation of each vertex. The weight function w provides a quantitative measure for the importance or relevance of each hyperedge.

Example 1.8 (Fuzzy Hypergraph Modeling E-commerce Purchase Baskets)**.** Consider an online store that sells four products:

$$V = \{\text{Laptop, Smartphone, Headphones, Smartwatch}\}.$$

We observe two purchase baskets:

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

where

- Basket e_1 contains a Laptop for \$800 and Headphones for \$200.
- Basket e_2 contains a Smartphone for \$600, a Smartwatch for \$200, and Headphones for \$200.

Define the fuzzy incidence matrix ψ by assigning to each product i in basket e the fraction of that basket's total spend:

$$\psi(e, i) = \frac{\text{price of } i \text{ in } e}{\sum_{j \in V} \text{price of } j \text{ in } e},$$

so that $\sum_{i \in V} \psi(e, i) = 1$ for each e. Concretely:

$$\psi(e_1, \text{Laptop}) = \tfrac{800}{1000} = 0.80, \quad \psi(e_1, \text{Headphones}) = \tfrac{200}{1000} = 0.20,$$
$$\psi(e_1, \text{Smartphone}) = 0, \quad \psi(e_1, \text{Smartwatch}) = 0;$$

$$\psi(e_2, \text{Smartphone}) = \tfrac{600}{1000} = 0.60, \quad \psi(e_2, \text{Smartwatch}) = \tfrac{200}{1000} = 0.20,$$
$$\psi(e_2, \text{Headphones}) = \tfrac{200}{1000} = 0.20, \quad \psi(e_2, \text{Laptop}) = 0.$$

Assign each basket a weight equal to its total value:

$$w(e_1) = 1000, \quad w(e_2) = 1000.$$

Thus

$$G = \big(V,\, E,\, \psi,\, w\big)$$

is a fuzzy hypergraph in which:

- Vertices represent products.
- Hyperedges represent individual purchase baskets.
- ψ_{ei} gives the degree to which product i contributes to basket e.
- $w(e)$ measures the overall importance (total spend) of basket e.

Definition 1.9 (Fuzzy n-Superhypergraph)**.** (cf. [4, 19]) Let $\mathrm{SHT}^{(n)} = (V, E)$ be an n-Superhypergraph. A *fuzzy n-Superhypergraph* is a quadruple

$$\big(V,\, E,\, \sigma,\, \mu\big),$$

where

- $\sigma : V \to [0, 1]$ assigns to each n-supervertex v a membership degree $\sigma(v)$.
- $\mu : E \to [0, 1]$ assigns to each n-superedge e a membership degree $\mu(e)$.

These functions satisfy the *appurtenance constraint*

$$\mu(e) \;\leq\; \min_{v \in e} \sigma(v), \quad \forall\, e \in E.$$

Example 1.10 (Fuzzy 2-Superhypergraph Modeling a Research Collaboration Network)**.** Consider a small research community with three base researchers:

$$V_0 = \{\text{Alice, Bob, Carol}\}, \quad n = 2.$$

We form the nonempty 2nd powerset

$$P_2^*(V_0) = \big\{\{\text{Alice}\}, \{\text{Bob}\}, \{\text{Carol}\}, \{\{\text{Alice}\}, \{\text{Bob}\}\}, \{\{\text{Bob}\}, \{\text{Carol}\}\}\big\}.$$

Define the fuzzy 2-superhypergraph

$$\mathrm{SHT}^{(2)} = \big(V^{(2)}, E^{(2)}, \sigma, \mu\big)$$

by

$$V^{(2)} = P_2^*(V_0), \qquad E^{(2)} = \big\{e_1 = \{\{\text{Alice}\}, \{\text{Bob}\}, \{\{\text{Alice}\}, \{\text{Bob}\}\}\},\ e_2 = \{\{\text{Bob}\}, \{\text{Carol}\}, \{\{\text{Bob}\}, \{\text{Carol}\}\}\}\big\}.$$

Here:

- Single-researcher vertices represent individual contributions: $\{\text{Alice}\}, \{\text{Bob}\}, \{\text{Carol}\}$.
- Two-researcher supervertices represent joint subteams: $\{\{\text{Alice}\}, \{\text{Bob}\}\}$ and $\{\{\text{Bob}\}, \{\text{Carol}\}\}$.
- Hyperedge e_1 models "Project AB" involving Alice, Bob, and their subteam; e_2 models "Project BC".

Assign fuzzy membership degrees (e.g. reflecting reliability of contribution):

$$\sigma(\{\text{Alice}\}) = 0.92, \quad \sigma(\{\text{Bob}\}) = 0.85, \quad \sigma(\{\text{Carol}\}) = 0.88,$$
$$\sigma\big(\{\{\text{Alice}\}, \{\text{Bob}\}\}\big) = 0.78, \quad \sigma\big(\{\{\text{Bob}\}, \{\text{Carol}\}\}\big) = 0.74.$$

By the appurtenance constraint, each superedge's degree is

$$\mu(e_1) = \min\{0.92,\ 0.85,\ 0.78\} = 0.78, \quad \mu(e_2) = \min\{0.85,\ 0.88,\ 0.74\} = 0.74.$$

Thus the quadruple

$$\big(V^{(2)}, E^{(2)}, \sigma, \mu\big)$$

is a *fuzzy 2-superhypergraph* capturing both individual and subteam reliabilities, with hyperedges modeling collaborative projects and their overall confidence levels.

1.3 Hypergraph Neural Networks

Graph Neural Networks (GNNs) are neural architectures designed for graph-structured data, where node representations are learned by aggregating information from adjacent vertices and edges [5, 34–36]. Hypergraph Neural Networks (HGNNs) extend this paradigm to hypergraphs, leveraging hyperedges—subsets of vertices that may connect more than two nodes—to capture higher-order relationships beyond simple pairwise interactions [11, 37]. Owing to their expressive power and versatility, HGNNs have attracted significant interest across a range of applications [38–40]. In what follows, we give a precise definition of the HGNN layer and then present an illustrative example.

Definition 1.11 (Hypergraph Neural Network (HGNN))**.** [11] Let $G = (V, E, W)$ be a finite, undirected hypergraph with

$$V = \{v_1, \dots, v_n\}, \quad E = \{e_1, \dots, e_m\}, \quad W = \mathrm{diag}(w_1, \dots, w_m),$$

where each $w_j > 0$. Let

$$X \in \mathbb{R}^{n \times d}$$

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

- The *incidence matrix* $H \in \{0,1\}^{n\times m}$ by

$$H_{ij} = \begin{cases} 1, & v_i \in e_j, \\ 0, & \text{otherwise.} \end{cases}$$

- The diagonal *vertex-degree matrix* $D_V \in \mathbb{R}^{n\times n}$ and *hyperedge-degree matrix* $D_E \in \mathbb{R}^{m\times m}$ by

$$(D_V)_{ii} = \sum_{j=1}^{m} H_{ij}\, w_j, \qquad (D_E)_{jj} = \sum_{i=1}^{n} H_{ij}.$$

- A learnable weight matrix $\Theta \in \mathbb{R}^{d\times c}$ and an activation function $\sigma(\cdot)$ (e.g. ReLU).

A single HGNN convolutional layer updates X to

$$Y = \sigma\big(D_V^{-\frac{1}{2}}\, H\, W\, D_E^{-1}\, H^{\top}\, D_V^{-\frac{1}{2}}\, X\, \Theta\big) \;\in\; \mathbb{R}^{n\times c}.$$

By stacking L such layers, one obtains

$$X^{(l+1)} = \sigma\big(D_V^{-\frac{1}{2}}\, H\, W\, D_E^{-1}\, H^{\top}\, D_V^{-\frac{1}{2}}\, X^{(l)}\, \Theta^{(l)}\big), \quad X^{(0)} = X.$$

For node classification, a final softmax readout produces

$$\hat{Y} = \mathrm{softmax}\big(X^{(L)}\big) \;\in\; \mathbb{R}^{n\times C}.$$

1.4 Fuzzy Graph Neural Network (F-GNN)

F-GNN combines fuzzy logic and graph neural networks, employing membership degrees to process and adaptively propagate uncertain node–edge features hierarchically [41–47].

Definition 1.12 (Fuzzy Graph Neural Network (F-GNN))**.** (cf. [41–44]) An *Fuzzy Graph Neural Network* is defined as a quintuple

$$\text{F-GNN} \;=\; \big(G,\ F_V,\ F_E,\ R,\ D\big),$$

where

- $G = (V, E)$ is a (crisp) graph with vertex set V and edge set E.
- $F_V\colon X_V \to [0,1]^M$ and $F_E\colon X_E \to [0,1]^M$ are the *fuzzification functions* for vertices and edges, respectively, mapping each attribute vector to an M-dimensional membership vector.
- R is the *rule layer*, encoding a finite set of fuzzy inference rules of the form

$$\textbf{IF}\ v_i \in A_m \wedge u_j \in A_n \quad \textbf{THEN} \quad y_k = f_k(x_{v_i}, x_{u_j}),$$

  where each A_m, A_n is a fuzzy subset and f_k is a trainable function.
- D is the *defuzzification function*, aggregating the normalized rule outputs into a crisp prediction.

2 Review: n-SuperHyperGraph Neural Network

In this section, we present the definition and illustrative examples of the n-SuperHyperGraph Neural Network [9,48–50]. We hope that these concrete examples will serve as a foundation for further research and development in the study of n-SuperHyperGraph Neural Networks.

Definition 2.1 (n-SuperHyperGraph Neural Network (n-SHGNN))**.** [9] 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 2.2 (One-Layer 2-SHGNN on a Simple 2-SuperHyperGraph)**.** Let the base vertex set be

$$V_0 = \{v_1, v_2\}, \quad n = 2.$$

We form the 2-superhypergraph

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

by choosing

$$V^{(2)} = \big\{\{v_1\},\ \{v_2\},\ \{\{v_1\},\{v_2\}\}\big\}, \quad E^{(2)} = \big\{\, \{\{v_1\},\{v_2\}\},\ \{\{v_1\},\{\{v_1\},\{v_2\}\}\}\big\}.$$

The expanded hypergraph $H' = (V_0, E')$ is obtained by

$$E' = \big\{\, e' \subseteq V_0 \mid e' = \bigcup_{v \in e} v,\ e \in E^{(2)} \big\} = \big\{\{v_1, v_2\},\ \{v_1, v_2\}\big\}.$$

Its binary incidence matrix is

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

Choose a one-layer 2-SHGNN with:

$$X = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \in \mathbb{R}^{2\times 2}, \quad W = I_2, \quad \Theta = I_2, \quad \sigma(x) = x.$$

Compute the diagonal degree matrices:

$$D_E = \operatorname{diag}\bigl(|e_1'|, |e_2'|\bigr) = 2\, I_2, \quad D_V = \operatorname{diag}\Bigl(\sum_j H'_{1j}, \sum_j H'_{2j}\Bigr) = 2\, I_2.$$

Then the single-layer update is

$$Y = \sigma\Bigl(D_V^{-\frac{1}{2}}\, H'\, W\, D_E^{-1}\, H'^{\top}\, D_V^{-\frac{1}{2}}\, X\,\Theta\Bigr) = \tfrac{1}{2}\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{pmatrix}.$$

Thus, each base vertex's updated feature is the average of its two hyperedge aggregates.

3 Review: Fuzzy HyperGraph Neural Network (F-HGNN)

F-HGNN extends GNNs to fuzzy hypergraphs, using vertex and hyperedge membership degrees to build fuzzy incidence matrices, apply inference rules, and defuzzify aggregated node features(cf. [10]).

Definition 3.1 (Fuzzy HyperGraph Neural Network (F-HGNN))**.** (cf. [10]) An *Fuzzy HyperGraph Neural Network* is defined as a sextuple

$$\text{F-HGNN} = \bigl(H,\ F_V,\ F_E,\ R,\ \Psi,\ D\bigr),$$

where

- $H = (V, E, \psi)$ is a finite fuzzy hypergraph with

$$\psi : E \times V \longrightarrow [0,1], \quad \sum_{v\in V} \psi(e,v) = 1 \quad (\forall e \in E).$$

- $F_V : X_V \to [0,1]^M$ and $F_E : X_E \to [0,1]^M$ are *fuzzification functions* mapping raw vertex/edge attributes to M-dimensional membership vectors.
- $R = \{r_k\}_{k=1}^K$ is a finite set of fuzzy inference rules.
- $\Psi \in [0,1]^{|V|\times|E|}$ is the *fuzzy incidence matrix*, $\Psi_{ij} = \psi(e_j, v_i)$.
- D is a defuzzification function aggregating normalized rule outputs into crisp predictions.

A single convolutional layer updates the node feature matrix $H^{(\ell-1)} \in \mathbb{R}^{|V|\times d}$ via

$$H^{(\ell)} = \sigma\bigl(D_V^{-\frac{1}{2}}\, \Psi\, W\, D_E^{-1}\, \Psi^{\top}\, D_V^{-\frac{1}{2}}\, H^{(\ell-1)}\, \Theta^{(\ell-1)}\bigr),$$

where $W \in \mathbb{R}^{|E|\times|E|}$ and $\Theta^{(\ell-1)} \in \mathbb{R}^{d\times c}$ are learnable weights, σ is a nonlinear activation, and D_V, D_E are diagonal degree matrices.

Example 3.2 (One-layer F-HGNN on a three-vertex fuzzy hypergraph)**.** Consider the finite fuzzy hypergraph

$$H = (V, E, \psi), \quad V = \{v_1, v_2, v_3\}, \quad E = \{e_1, e_2\},$$

with fuzzy incidence function ψ given by

$$\psi(e_1, v_1) = 0.6,\ \psi(e_1, v_2) = 0.4,\ \psi(e_1, v_3) = 0, \quad \psi(e_2, v_1) = 0,\ \psi(e_2, v_2) = 0.3,\ \psi(e_2, v_3) = 0.7.$$

Hence the fuzzy incidence matrix is

$$\Psi = \begin{bmatrix} 0.6 & 0 \\ 0.4 & 0.3 \\ 0 & 0.7 \end{bmatrix}.$$

Let the initial node feature matrix and model parameters be

$$X = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix}, \quad W = I_2, \quad \Theta = I_2, \quad \sigma(x) = x.$$

For simplicity we choose trivial fuzzification functions $F_V(x_v) = x_v$ and $F_E(x_e) = x_e$.

Compute the diagonal degree matrices:

$$D_E = \begin{pmatrix} \sum_{i=1}^{3} \Psi_{i1} & 0 \\ 0 & \sum_{i=1}^{3} \Psi_{i2} \end{pmatrix} = I_2, \qquad D_V = \operatorname{diag}(0.6,\ 0.7,\ 0.7).$$

A single convolutional update is

$$H^{(1)} \; = \; D_V^{-\frac{1}{2}} \, \Psi \, W \, D_E^{-1} \, \Psi^{\top} \, D_V^{-\frac{1}{2}} \, X \, \Theta \; = \; \begin{bmatrix} 0.600 & 0.371 \\ 0.671 & 0.657 \\ 0.700 & 1.000 \end{bmatrix}.$$

Thus the updated feature matrix $H^{(1)}$ reflects fuzzy aggregation over hyperedges.

Theorem 3.3 (Generalization of F-GNN)**.** *If each hyperedge in H is of size two and $\psi(e, v) \in \{0, 1\}$, then F-HGNN reduces exactly to the standard Fuzzy Graph Neural Network.*

Proof. Under these restrictions, $E \subseteq \{\{u, v\}\}$ and Ψ becomes the binary incidence matrix of a crisp graph. The convolutional update and rule layer coincide term-by-term with those in F-GNN (Definition of F-GNN), hence F-HGNN specializes to F-GNN. □

Theorem 3.4 (Fuzzy Hypergraph Structure)**.** *The convolutional operator of F-HGNN preserves and exploits the fuzzy hypergraph structure by aggregating node features along hyperedges weighted by membership degrees.*

Proof. By construction, $\Psi_{ij} = \psi(e_j, v_i)$ encodes fuzzy membership of node v_i in hyperedge e_j. The term $\Psi \, W \, D_E^{-1} \, \Psi^{\top}$ thus aggregates contributions from all nodes in each hyperedge according to their membership, and redistributes back to nodes, exactly reflecting the fuzzy hypergraph connectivity. □

4 Result: Fuzzy n-SuperHyperGraph Neural Network (F-nSHGNN)

F-nSHGNN integrates fuzzy logic with hierarchical n-level superhypergraphs, mapping base vertex features to fuzzy membership, applying inference rules over supervertices and superedges, aggregating through defuzzification.

Definition 4.1 (Fuzzy n-SuperHyperGraph Neural Network (F-nSHGNN))**.** Let V_0 be a finite set of *base vertices* and denote by $P_n^*(V_0)$ the collection of all nonempty subsets obtained by applying the power-set operator n times to V_0. A *Fuzzy n-SuperHyperGraph Neural Network* is specified by the data

$$\text{F-nSHGNN} \; = \; \big(\mathrm{SHT}^{(n)},\ F_V^{(n)},\ F_E^{(n)},\ R_n,\ D_n\big),$$

where:

(a) **Fuzzy n-superhypergraph.** $\mathrm{SHT}^{(n)} = (V^{(n)}, E^{(n)}, \sigma, \mu)$ with

$$V^{(n)} \subseteq P_n^*(V_0), \quad E^{(n)} \subseteq P_n^*(V_0), \quad \sigma : V^{(n)} \to [0,1], \quad \mu : E^{(n)} \to [0,1],$$

satisfying the *appurtenance constraint*

$$\mu(e) \leq \min_{v \in e} \sigma(v), \quad \forall e \in E^{(n)}.$$

(b) **Vertex fuzzification.** $F_V^{(n)} : X_V \to [0,1]^M$ maps each raw vertex feature $x_v \in X_V \subseteq \mathbb{R}^d$ to an M-dimensional membership vector $\big[\mu_1(x_v), \ldots, \mu_M(x_v)\big]$.

(c) **Edge fuzzification.** $F_E^{(n)} : X_E \to [0,1]^M$ maps each raw (super)edge feature $x_e \in X_E \subseteq \mathbb{R}^p$ to a membership vector $\big[\nu_1(x_e), \ldots, \nu_M(x_e)\big]$.

(d) **Fuzzy rule layer.** $R_n = \{r_n^k\}_{k=1}^K$ is a finite set of fuzzy inference rules of the form

$$\textbf{IF } x_v \in A_v^k \ \wedge \ x_e \in A_e^k \quad \textbf{THEN} \quad z^k = f^k(x_v, x_e),$$

where each $A_v^k \subseteq V^{(n)}$ and $A_e^k \subseteq E^{(n)}$ are fuzzy antecedents, and f^k is a learnable function producing a scalar output z^k.

(e) **Defuzzification.** $D_n : [0,1]^K \times \mathbb{R}^K \to \mathbb{R}$ aggregates rule strengths α^k and outputs z^k into a crisp value, for example

$$D_n\big(\alpha, z\big) = \frac{\sum_{k=1}^K \alpha^k z^k}{\sum_{k=1}^K \alpha^k}, \quad \alpha^k = \text{firing strength of } r_n^k.$$

Example 4.2 (One-Layer F-2SHGNN on a Simple Fuzzy 2-SuperHyperGraph)**.** Let the base vertex set be

$$V_0 = \{v_1, v_2\}, \quad n = 2.$$

The nonempty 2nd powerset is

$$P_2^*(V_0) = \big\{\{v_1\}, \{v_2\}, \{\{v_1\}, \{v_2\}\}\big\}.$$

We choose the fuzzy 2-superhypergraph

$$\mathrm{SHT}^{(2)} = \big(V^{(2)}, E^{(2)}, \sigma, \mu\big)$$

with

$$V^{(2)} = P_2^*(V_0), \quad E^{(2)} = \big\{e_1, e_2\big\},$$

where

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

Assign fuzzy membership degrees:

$$\sigma\big(\{v_1\}\big) = 0.9, \ \sigma\big(\{v_2\}\big) = 0.7, \ \sigma\big(\{\{v_1\}, \{v_2\}\}\big) = 0.8,$$

and, by the appurtenance constraint,

$$\mu(e_1) = \min(0.9, 0.7) = 0.7, \quad \mu(e_2) = \min(0.9, 0.8) = 0.8.$$

Choose trivial fuzzification:

$$F_V^{(2)}(x_v) = \sigma(v), \quad F_E^{(2)}(x_e) = \mu(e).$$

Define a single fuzzy inference rule:

$$r^1 : \quad \textbf{IF } v \in V^{(2)} \text{ with degree } \alpha_v \ \wedge \ e \in E^{(2)} \text{ with degree } \alpha_e \quad \textbf{THEN} \quad z = \alpha_v \cdot \alpha_e.$$

Thus the firing strength is $\alpha^1 = \min(\alpha_v, \alpha_e)$, and the rule output is $z^1 = \alpha_v \alpha_e$.

Defuzzification uses the weighted average:

$$D_2(\{\alpha^k\}, \{z^k\}) = \frac{\sum_k \alpha^k z^k}{\sum_k \alpha^k}.$$

Compute for each base vertex:

(i) Vertex $\{v_1\}$. It belongs to e_1, e_2. For e_1:

$$\alpha_1^1 = \min(0.9, 0.7) = 0.7, \quad z_1^1 = 0.9 \times 0.7 = 0.63.$$

For e_2:

$$\alpha_1^2 = \min(0.9, 0.8) = 0.8, \quad z_1^2 = 0.9 \times 0.8 = 0.72.$$

Defuzzified output:

$$y_1 = \frac{0.7 \cdot 0.63 + 0.8 \cdot 0.72}{0.7 + 0.8} = \frac{0.441 + 0.576}{1.5} = 0.678.$$

(ii) Vertex $\{v_2\}$. It belongs only to e_1:

$$\alpha_2^1 = \min(0.7, 0.7) = 0.7, \quad z_2^1 = 0.7 \times 0.7 = 0.49,$$

so

$$y_2 = \frac{0.7 \cdot 0.49}{0.7} = 0.49.$$

Hence the one-layer F-2SHGNN produces the crisp outputs $\big[y_1, y_2\big] = [0.678,\ 0.49]$.

Example 4.3 (One-Layer F-2SHGNN on a 2-SuperHyperGraph with Three Base Vertices)**.** Let the base vertex set be

$$V_0 = \{v_1,\ v_2,\ v_3\}, \quad n = 2.$$

The nonempty 2nd powerset is

$$P_2^*(V_0) = \big\{\{v_1\},\ \{v_2\},\ \{v_3\},\ \{\{v_1\}, \{v_2\}\},\ \{\{v_2\}, \{v_3\}\}\big\}.$$

We choose the fuzzy 2-superhypergraph

$$\mathrm{SHT}^{(2)} = \big(V^{(2)},\ E^{(2)},\ \sigma,\ \mu\big)$$

with

$$V^{(2)} = \big\{\, u_1 = \{v_1\},\ u_2 = \{v_2\},\ u_3 = \{v_3\},\ u_4 = \{\{v_1\}, \{v_2\}\},\ u_5 = \{\{v_2\}, \{v_3\}\}\big\},$$

$$E^{(2)} = \{\, e_1 = \{u_1, u_4, u_2\},\ e_2 = \{u_2, u_4, u_5, u_3\}\}.$$

Assign membership degrees:

$$\sigma(u_1) = 0.90, \quad \sigma(u_2) = 0.80, \quad \sigma(u_3) = 0.60, \quad \sigma(u_4) = 0.70, \quad \sigma(u_5) = 0.65,$$

and by the appurtenance constraint

$$\mu(e_1) = \min\{0.90, 0.70, 0.80\} = 0.70, \quad \mu(e_2) = \min\{0.80, 0.70, 0.65, 0.60\} = 0.60.$$

Fuzzification and Rule:

$$F_V^{(2)}(x_u) = \sigma(u), \quad F_E^{(2)}(x_e) = \mu(e),$$

One fuzzy inference rule applied per hyperedge:

$$r^k: \quad \textbf{IF } u \text{ has degree } \alpha_u \wedge e_k \text{ has degree } \alpha_{e_k} \text{ THEN } z^k = \alpha_u\, \alpha_{e_k},$$

with firing strength $\alpha^k = \min(\alpha_u, \alpha_{e_k})$.

Defuzzification:

$$D_2(\{\alpha^k\}, \{z^k\}) = \frac{\sum_k \alpha^k\, z^k}{\sum_k \alpha^k}.$$

Compute outputs for each u_i:

(i) $u_1 = \{v_1\}$. Belongs only to e_1:

$$\alpha_1^1 = \min(0.90, 0.70) = 0.70, \quad z_1^1 = 0.90 \times 0.70 = 0.63,$$

$$y_1 = \frac{0.70 \cdot 0.63}{0.70} = 0.63.$$

(ii) $u_2 = \{v_2\}$. Belongs to e_1, e_2:

$$\alpha_2^1 = \min(0.80, 0.70) = 0.70,\ z_2^1 = 0.80 \times 0.70 = 0.56, \quad \alpha_2^2 = \min(0.80, 0.60) = 0.60,\ z_2^2 = 0.80 \times 0.60 = 0.48,$$

$$y_2 = \frac{0.70 \cdot 0.56 + 0.60 \cdot 0.48}{0.70 + 0.60} = \frac{0.392 + 0.288}{1.30} \approx 0.528.$$

(iii) $u_3 = \{v_3\}$. Belongs only to e_2:

$$\alpha_3^2 = \min(0.60, 0.60) = 0.60, \quad z_3^2 = 0.60 \times 0.60 = 0.36,$$

$$y_3 = \frac{0.60 \cdot 0.36}{0.60} = 0.36.$$

(iv) $u_4 = \{\{v_1\}, \{v_2\}\}$. Belongs to both:

$$\alpha_4^1 = \min(0.70, 0.70) = 0.70,\ z_4^1 = 0.70 \times 0.70 = 0.49, \quad \alpha_4^2 = \min(0.70, 0.60) = 0.60,\ z_4^2 = 0.70 \times 0.60 = 0.42,$$

$$y_4 = \frac{0.70 \cdot 0.49 + 0.60 \cdot 0.42}{0.70 + 0.60} = \frac{0.343 + 0.252}{1.30} \approx 0.4577.$$

(v) $u_5 = \{\{v_2\}, \{v_3\}\}$. Belongs only to e_2:

$$\alpha_5^2 = \min(0.65, 0.60) = 0.60, \quad z_5^2 = 0.65 \times 0.60 = 0.39,$$

$$y_5 = \frac{0.60 \cdot 0.39}{0.60} = 0.39.$$

Thus the one-layer F-2SHGNN outputs the crisp vector $[y_1, y_2, y_3, y_4, y_5] \approx [0.63,\ 0.528,\ 0.36,\ 0.458,\ 0.39]$.

Theorem 4.4 (Simultaneous Generalization of F-GNN, F-HGNN, and n-SHGNN)**.** *The F-nSHGNN framework recovers the following as special cases:*

- Fuzzy Graph Neural Network (F-GNN): *set $n = 0$, so $V^{(0)} = V_0$, $E^{(0)} \subseteq V_0 \times V_0$, and restrict $\sigma, \mu \in \{0, 1\}$. Then $F_V^{(0)}, F_E^{(0)}, R_0, D_0$ coincide with those of F-GNN.*
- Fuzzy HyperGraph Neural Network (F-HGNN): *set $n = 1$, so* $\mathrm{SHT}^{(1)}$ *is a fuzzy hypergraph; the fuzzification and inference reduce exactly to the F-HGNN definition.*
- n-SuperHyperGraph Neural Network (n-SHGNN): *if $\sigma(v) = \mu(e) = 1$ for all v, e, then fuzzification is trivial and R_n, D_n become crisp, yielding the classical n-SHGNN update.*

Proof. (i) $n = 0$. Here $P_0^*(V_0) = V_0$ and $E^{(0)}$ consists of vertex–vertex pairs. Binary σ, μ force all membership degrees to 0 or 1, so the fuzzy components reduce to those in F-GNN by identification.

(ii) $n = 1$. Then $V^{(1)}, E^{(1)}$ recover a fuzzy hypergraph; the rule layer R_1 and defuzzifier D_1 thus match the F-HGNN specification.

(iii) *Trivial fuzzification.* If $\sigma(v) = \mu(e) = 1$ uniformly, then $F_V^{(n)}, F_E^{(n)}$ output all-ones vectors and R_n, D_n perform ordinary numerical aggregation, reproducing the crisp n-SHGNN equations.

Hence F-nSHGNN indeed generalizes F-GNN, F-HGNN, and n-SHGNN. □

5 Conclusion and Future Works

In this paper, we introduced and investigated the Fuzzy SuperHyperGraph Neural Network (F-SHGNN), a novel framework that integrates and extends SuperHyperGraph Neural Networks, Fuzzy Graph Neural Networks, and Fuzzy HyperGraph Neural Networks. In future work, we plan to explore further extensions based on Neutrosophic Sets [51], Plithogenic Sets [52], and other advanced uncertainty frameworks.

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.

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 Approval

As this research is entirely theoretical in nature and does not involve human participants or animal subjects, no ethical approval is required.

Conflicts of Interest

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

Disclaimer

This work presents theoretical concepts that have not yet undergone practical testing or validation. Future researchers are encouraged to apply and assess these ideas in empirical contexts. While every effort has been made to ensure accuracy and appropriate referencing, unintentional errors or omissions may still exist. Readers are advised to verify referenced materials on their own. The views and conclusions expressed here are the authors' own and do not necessarily reflect those of their affiliated organizations.

References

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

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

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

[4] 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.

[5] Guillaume Verdon, Trevor McCourt, Enxhell Luzhnica, Vikash Singh, Stefan Leichenauer, and Jack Hidary. Quantum graph neural networks. *arXiv preprint arXiv:1909.12264*, 2019.

[6] Lilas Alrahis and Ozgur Sinanoglu. Graph neural networks for hardware vulnerability analysis-can you trust your gnn? In *2023 IEEE 41st VLSI Test Symposium (VTS)*, pages 1–4. IEEE, 2023.

[7] Xiaojun Kang, Xinchuan Li, Hong Yao, Dan Li, Bo Jiang, Xiaoyue Peng, Tiejun Wu, Shihua Qi, and Lijun Dong. Dynamic hypergraph neural networks based on key hyperedges. *Information Sciences*, 616:37–51, 2022.

[8] Yuxin Wang, Quan Gan, Xipeng Qiu, Xuanjing Huang, and David Wipf. From hypergraph energy functions to hypergraph neural networks. In *International Conference on Machine Learning*, pages 35605–35623. PMLR, 2023.

[9] Takaaki Fujita and Florentin Smarandache. Superhypergraph neural networks and plithogenic graph neural networks: Theoretical foundations. 2025.

[10] Tao Yin, Weiping Ding, Hengrong Ju, Jiashuang Huang, and Yuepeng Chen. The fuzzy hypergraph neural network model based on sparse k-nearest neighborhood granule. *Applied Soft Computing*, 170:112721, 2025.

[11] 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.

[12] Yue Gao, Zizhao Zhang, Haojie Lin, Xibin Zhao, Shaoyi Du, and Changqing Zou. Hypergraph learning: Methods and practices. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(5):2548–2566, 2020.

[13] Yifan Feng, Jiashu Han, Shihui Ying, and Yue Gao. Hypergraph isomorphism computation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[14] Y. V. M. Cepeda, M. A. R. Guevara, E. J. J. Mogro, and R. P. Tizano. Impact of irrigation water technification on seven directories of the san juan-patoa river using plithogenic n-superhypergraphs based on environmental indicators in the canton of pujili, 2021. *Neutrosophic Sets and Systems*, 74:46–56, 2024.

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

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

[17] Mohammad Hamidi, Florentin Smarandache, and Elham Davneshvar. Spectrum of superhypergraphs via flows. *Journal of Mathematics*, 2022(1):9158912, 2022.

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

[19] Mohammad Hamidi, Florentin Smarandache, and Mohadeseh Taghinezhad. *Decision Making Based on Valued Fuzzy Superhypergraphs*. Infinite Study, 2023.

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

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

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

[23] Lotfi A Zadeh. Fuzzy sets. *Information and control*, 8(3):338–353, 1965.

[24] Lotfi A Zadeh. Fuzzy logic, neural networks, and soft computing. In *Fuzzy sets, fuzzy logic, and fuzzy systems: selected papers by Lotfi A Zadeh*, pages 775–782. World Scientific, 1996.

[25] TM Nishad, Talal Ali Al-Hawary, and B Mohamed Harif. General fuzzy graphs. *Ratio Mathematica*, 47, 2023.

[26] Azriel Rosenfeld. Fuzzy graphs. In *Fuzzy sets and their applications to cognitive and decision processes*, pages 77–95. Elsevier, 1975.

[27] Takaaki Fujita and Florentin Smarandache. A review of the hierarchy of plithogenic, neutrosophic, and fuzzy graphs: Survey and applications. In *Advancing Uncertain Combinatorics through Graphization, Hyperization, and Uncertainization: Fuzzy, Neutrosophic, Soft, Rough, and Beyond (Second Volume)*. Biblio Publishing, 2024.

[28] Mahdi Farshi and Bijan Davvaz. Generalized fuzzy hypergraphs and hypergroupoids. *Filomat*, 30(9):2375–2387, 2016.

[29] Sovan Samanta and Madhumangal Pal. Bipolar fuzzy hypergraphs. *International Journal of Fuzzy Logic Systems*, 2(1):17–28, 2012.

[30] Muhammad Akram and Anam Luqman. Fuzzy hypergraphs and related extensions. In *Studies in Fuzziness and Soft Computing*, 2020.

[31] John N Mordeson and Premchand S Nair. *Fuzzy graphs and fuzzy hypergraphs*, volume 46. Physica, 2012.

[32] Leonid S Bershtein and Alexander V Bozhenyuk. Fuzzy graphs and fuzzy hypergraphs. In *Encyclopedia of Artificial Intelligence*, pages 704–709. IGI Global, 2009.

[33] Muhammad Akram and Anam Luqman. *Fuzzy hypergraphs and related extensions*. Springer, 2020.

[34] Chuxu Zhang, Dongjin Song, Chao Huang, Ananthram Swami, and Nitesh V Chawla. Heterogeneous graph neural network. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 793–803, 2019.

[35] Hayoung Jo and Seong-Whan Lee. Edge conditional node update graph neural network for multi-variate time series anomaly detection. *Information Sciences*, page 121062, 2024.

[36] Chunhang Zheng and Kechao Cai. Genet: A graph neural network-based anti-noise task-oriented semantic communication paradigm. *arXiv preprint arXiv:2403.18296*, 2024.

[37] Bo-Lan Wang, Guibao Shen, Dong Li, Jianye Hao, Wulong Liu, Yu Huang, Hongzhong Wu, Yibo Lin, Guangyong Chen, and Pheng-Ann Heng. Lhnn: lattice hypergraph neural network for vlsi congestion prediction. *Proceedings of the 59th ACM/IEEE Design Automation Conference*, 2022.

[38] Wei Li, Bin Xiang, Fan Yang, Yuchen Rong, Yanbin Yin, Jianhua Yao, and Han Zhang. scmhnn: a novel hypergraph neural network for integrative analysis of single-cell epigenomic, transcriptomic and proteomic data. *Briefings in bioinformatics*, 24 6, 2023.

[39] Luotao Liu, Feng Huang, Xuan Liu, Zhankun Xiong, Menglu Li, Congzhi Song, and Wen Zhang. Multi-view contrastive learning hypergraph neural network for drug-microbe-disease association prediction. In *International Joint Conference on Artificial Intelligence*, 2023.

[40] Yousra Moh Ousellam, Bikram Pratim BHUYAN, Rachida Fissoune, and Amar Ramdane Cherif. Directed hypergraph neural network: Building a predictive framework. *Bikram Pratim and Fissoune, Rachida and Cherif, Amar Ramdane, Directed Hypergraph Neural Network: Building a Predictive Framework*, 2025.

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

[42] 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.

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

[44] Tong Wei, Junlin Hou, and Rui Feng. Fuzzy graph neural network for few-shot learning. In *2020 International joint conference on neural networks (IJCNN)*, pages 1–8. IEEE, 2020.

[45] Giovanna Castellano, Raffaele Scaringi, Gennaro Vessio, Gianluca Zaza, et al. Integrating graph neural networks and fuzzy logic to enhance deep learning interpretability. In *CEUR WORKSHOP PROCEEDINGS*, volume 3831, 2024.

[46] Xianchun Zhou. Homology detection of malicious codes based on a fuzzy graph neural network. In *2021 IEEE International Conference on Industrial Application of Artificial Intelligence (IAAI)*, pages 202–207. IEEE, 2021.

[47] Amrut Ranjan Jena, Raja Das, and DP Acharjya. An integrated approach for prediction of radial overcut in electro discharge machining using fuzzy graph recurrent neural network. *International Journal of Embedded Systems*, 14(4):345–354, 2021.

[48] Takaaki Fujita. Superhypergraph neural network and dynamic superhypergraph neural network. 2025.

[49] Takaaki Fujita. Directed superhypergraph neural networks. 2025.

[50] Takaaki Fujita. Multi-superhypergraph neural networks: A generalization of multi-hypergraph neural networks. 2025.

[51] Florentin Smarandache. A unifying field in logics: Neutrosophic logic. In *Philosophy*, pages 1–141. American Research Press, 1999.

[52] Florentin Smarandache. Plithogeny, plithogenic set, logic, probability, and statistics. *arXiv preprint arXiv:1808.03948*, 2018.