Article

Not peer-reviewed version

SuperHyperGraph Foundations for Artificial Intelligence, Machine Learning, and Neural Networks

Takaaki Fujita [*]

Posted Date: 24 December 2025

doi: 10.20944/preprints202512.2233.v1

Keywords: SuperHyperGraph; HyperGraph; GCN; GraphRAG; causal graphs; graph embedding

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 24 December 2025 doi:10.20944/preprints202512.2233.v1

Article

SuperHyperGraph Foundations for Artificial Intelligence, Machine Learning, and Neural Networks

Takaaki Fujita

Independent Researcher, Tokyo, Japan; Takaaki.fujita060@gmail.com

Abstract

A *hypergraph* replaces ordinary edges by *hyperedges*, where each hyperedge may connect an arbitrary nonempty subset of the vertex set, thus encoding higher-order relations directly. Beyond this, a *superhypergraph* introduces explicit nesting by iterating the powerset operation; this produces multi-level collections of vertex-sets and edge-sets and thereby allows one to represent hierarchical groupings and multi-layer connectivity within a single combinatorial object. This paper extends several fundamental frameworks-including GCN, GraphRAG, causal graphs, graph embedding, graph-based natural language processing, and graph generation-by incorporating the SuperHyperGraph viewpoint.

Keywords: SuperHyperGraph; HyperGraph; GCN; GraphRAG; causal graphs; graph embedding

---

1. Introduction

1.1. Graph, Hypergraph, and Superhypergraph Theory

Graph theory offers a rigorous framework for analyzing discrete relational systems: a set of vertices is connected by edges, and the resulting structure naturally captures pairwise interactions [1,2]. Owing to the clarity and flexibility of this abstraction, graphs have become standard models in many disciplines, including data mining, algorithm design, artificial intelligence, neural computation, quantum information, and chemistry (see, e.g., [3,4]). However, requiring every edge to link exactly two vertices may be inadequate when the underlying phenomenon involves genuine multiway relations or layered (hierarchical) organization. This motivates the parallel development of *hypergraph* and *superhypergraph* theories, which enlarge the classical setting in complementary directions.

A *hypergraph* replaces ordinary edges by *hyperedges*, where each hyperedge may connect an arbitrary nonempty subset of the vertex set, thus encoding higher-order relations directly [5–7]. As a strict generalization of graphs, hypergraph theory has grown into a mature area with substantial structural and algorithmic results and with a wide range of applications (see, e.g., [8–10]). Beyond this, a *superhypergraph* introduces explicit nesting by iterating the powerset operation; this produces multi-level collections of vertex-sets and edge-sets and thereby allows one to represent hierarchical groupings and multi-layer connectivity within a single combinatorial object [11–15]. Accordingly, superhypergraphs extend hypergraphs and are particularly suitable for modeling complex systems with several interacting layers [16–18].

Table 1 highlights the principal differences among graphs, hypergraphs, and superhypergraphs. Throughout, n denotes a finite positive integer. We also emphasize that extensive algorithmic literature exists for these models, ranging from classical graph algorithms to modern extensions and adaptations (see, e.g., [19,20]).

Table 1. Comparison of Graph, Hypergraph, and Superhypergraph.

Concept	Notation	Edge Type	How it extends the model
Graph	$G = (V, E)$	$E \subseteq \{\{u, v\} \mid u, v \in V,\ u \neq v\}$	Uses pairwise edges; each edge joins exactly two distinct vertices.
Hypergraph	$H = (V, E)$	$E \subseteq \mathrm{PS}(V) \setminus \{\varnothing\}$	Uses hyperedges; each hyperedge may join any nonempty subset of vertices.
Superhypergraph	$\mathrm{SHG}^{(n)} = (V_0, V, E)$	$V \subseteq \mathrm{PS}(V_0),\ E \subseteq \mathrm{PS}(V)$	Introduces nested objects via an n-fold iterated powerset, enabling hierarchical (multi-level) connectivity.

(Here $\mathrm{PS}(V)$ denotes the powerset of V, i.e., the set of all subsets of V. The iterated powerset $\mathrm{PS}^n(V)$ is defined recursively by $\mathrm{PS}^1(V) = \mathrm{PS}(V)$ and $\mathrm{PS}^{k+1}(V) = \mathrm{PS}(\mathrm{PS}^k(V))$ for $k \geq 1$.)

Note that graphs, hypergraphs, and superhypergraphs admit natural directed counterparts—Directed Graphs [21,22], Directed HyperGraphs [23,24], and Directed SuperHyperGraphs [25]—and these directed models can be further extended to bidirected versions, namely Bidirected Graphs [26–28], Bidirected HyperGraphs [22], and Bidirected SuperHyperGraphs [22]. Such directed and bidirected variants enrich the theory by explicitly incorporating orientation information into the underlying combinatorial structures.

1.2. HyperGraph and SuperHyperGraph for Artificial Intelligence

It is needless to say that research in machine learning and artificial intelligence has become extremely active in recent years and now plays a very significant role in modern society. Graph, HyperGraph, and SuperHyperGraph are also used in the field of Artificial Intelligence. Examples of related concepts are presented in Table 2.

Table 2. Parallel terminology list: Graph vs. HyperGraph vs. SuperHyperGraph.

Category	Graph	HyperGraph	SuperHyperGraph
Structure	Graph [1]	HyperGraph [29]	SuperHyperGraph [15,25]
Neural-network family	GNN (Graph Neural Network)	HGNN (HyperGraph Neural Network) [5,30–32]	SHGNN (SuperHyperGraph Neural Network) [33]
Directed / multi-edge variants (neural)	Directed GNN (DGNN); MultiGraph Neural Network (MGNN) [34–36]	Directed HyperGraph Neural Network[37–39]; MultiHyperGraph Neural Network[40]	Directed SuperHyperGraph Neural Network; Multi-SuperHyperGraph Neural Network[40]
Uncertainty modeling	Probabilistic Graph [41]	Probabilistic HyperGraph [42,43]	Probabilistic SuperHyperGraph [43]
DAG modeling	Directed Acyclic Graph [44,45]	Directed Acyclic HyperGraph [46]	Directed Acyclic SuperHyperGraph [46]
Molecular modeling	Molecular Graph [47,48]	Molecular HyperGraph [49,50]	Molecular SuperHyperGraph [51,52]
Attention models	Graph Attention Network (GAT) [53,54]	HyperGraph Attention Network (HGAT) [55,56]	SuperHyperGraph Attention Network (SHGAT) [57]
RAG (retrieval-augmented generation)	GraphRAG [58,59]	HyperGraphRAG [60]	SuperHyperGraphRAG
Representation learning	Graph Embedding [61]	HyperGraph Embedding [61,62]	SuperHyperGraph Embedding
Generative modeling	Graph Generation [63,64]	HyperGraph Generation [65]	SuperHyperGraph Generation

Table 2. *Cont.*

Category	Graph	HyperGraph	SuperHyperGraph
Convolutional models	Graph Convolutional Network (GCN) [66–68]	HyperGraph Convolutional Network (HGCN) [69–71]	SuperHyperGraph Convolutional Network (SHGCN)
Dynamics (temporal)	Dynamic Graph [72,73]	Dynamic HyperGraph [74–76]	Dynamic SuperHyperGraph [77]
Uncertain (Fuzzy/ Neutrosophic/Plithogenic) [78]	Uncertain Graph [79,80]	Uncertain HyperGraph [81]	Uncertain SuperHyperGraph [15]
Weights / attributes	Weighted Graph [82,83]	Weighted HyperGraph [70,84]	Weighted SuperHyperGraph [85]

1.3. Our Contributions

From the above discussion, research based on SuperHyperGraphs is important and is expected to play a major role in fields such as AI. Accordingly, this paper extends several fundamental frameworks-including GCN, GraphRAG, causal graphs, graph embedding, graph-based natural kanguage processing and graph generation-by incorporating the SuperHyperGraph viewpoint. Some of these parallel notions are summarized in Table 2.

2. Preliminaries

This section introduces the notation and basic terminology used in the sequel. Unless explicitly stated otherwise, all graphs and hypergraphs in this paper are assumed to be *finite*.

2.1. SuperHyperGraphs

A *hypergraph* generalizes an ordinary graph by allowing an edge to connect any nonempty subset of the vertex set. Hence, hypergraphs provide a direct language for modeling multiway interactions [5,6,86,87]. By iterating the powerset operation one step further, one obtains *nested* (higher-order) vertex objects and, consequently, a finite *SuperHyperGraph* whose vertices and edges may themselves be set-valued at multiple levels [16,18,40,88]. Such hierarchical representations are useful, for instance, in molecular design, complex-network analysis, and related applications [57,89,90]. Throughout, the index n in $\mathrm{PS}_n(\cdot)$ and in an n-SuperHyperGraph is always taken to be a nonnegative integer.

Definition 1 (Base set). *A* base set *S is the underlying universe of discourse:*

$$S = \{\, x \mid x \text{ is an admissible object in the context under consideration} \,\}.$$

All sets appearing in $\mathrm{PS}(S)$ and in the iterated powersets $\mathrm{PS}_n(S)$ are ultimately built from elements of S.

Definition 2 (Powerset). *(see [91–93]) For a set S, the* powerset *of S is*

$$\mathrm{PS}(S) = \{\, A \mid A \subseteq S \,\}.$$

In particular, $\emptyset \in \mathrm{PS}(S)$ and $S \in \mathrm{PS}(S)$.

Definition 3 (Hypergraph). *[29,94] A* hypergraph *is a pair $H = (V, E)$ such that:*

- *V is a finite set (the* vertices*), and*
- *E is a finite family of nonempty subsets of V (the* hyperedges*).*

Thus, a hyperedge may involve more than two vertices, capturing genuinely multiway relations.

Definition 4 (n-th powerset). *[95,96] Let X be a set. Define $\mathrm{PS}_1(X) := \mathrm{PS}(X)$ and, for every $n \geq 1$,*

$$\mathrm{PS}_{n+1}(X) = \mathrm{PS}\big(\mathrm{PS}_n(X)\big).$$

When it is convenient to exclude the empty set, we write

$$\mathrm{PS}_n^*(X) \;=\; \mathrm{PS}_n(X) \setminus \{\emptyset\}.$$

Definition 5 (n-SuperHyperGraph)**.** *(see [15,97]) Let V_0 be a finite, nonempty base set. Define*

$$\mathrm{PS}^0(V_0) := V_0, \qquad \mathrm{PS}^{k+1}(V_0) := \mathrm{PS}\big(\mathrm{PS}^k(V_0)\big) \quad (k \in \mathbb{N}).$$

*For $n \geq 0$, an n-*SuperHyperGraph on *V_0 is a pair*

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

satisfying

$$V \subseteq \mathrm{PS}^n(V_0) \quad \textit{and} \quad E \subseteq \mathrm{PS}(V) \setminus \{\emptyset\}.$$

*Elements of V are called n-*supervertices, *and elements of E are called n-*superedges *(i.e., each n-superedge is a nonempty subset of V).*

3. Review: SuperHyperGraph Neural Network

A HyperGraph Neural Network learns node representations by aggregating features through hyperedges, capturing higher-order interactions among multiple vertices for prediction tasks [5]. A SuperHyperGraph Neural Network learns base-vertex representations by aggregating through nested hyperedges, capturing hierarchical hyperedge-of-hyperedge interactions for complex reasoning tasks [33].

Definition 6 (HyperGraph Neural Network (HGNN))**.** *[5] Let $G = (V, E, W)$ be a finite undirected hypergraph, where*

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

with $w_j > 0$ the (hyperedge) weight of e_j. 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 of vertex v_i.

Define the incidence matrix $H \in \{0,1\}^{n\times m}$ by

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

Define the diagonal degree matrices $D_V \in \mathbb{R}^{n\times n}$ and $D_E \in \mathbb{R}^{m\times m}$ by

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

Let $\Theta \in \mathbb{R}^{d\times c}$ be a learnable weight matrix and let $\sigma(\cdot)$ be a nonlinearity (e.g., ReLU). A single HGNN convolutional layer is the map

$$\mathrm{HGNN}_\Theta : \mathbb{R}^{n\times d} \to \mathbb{R}^{n\times c}, \qquad Y = \mathrm{HGNN}_\Theta(X) := \sigma\Big(D_V^{-\frac{1}{2}}\, H\, W\, D_E^{-1}\, H^\top\, D_V^{-\frac{1}{2}}\, X\, \Theta\Big).$$

By stacking L layers, one obtains embeddings $X^{(0)} = X$ and

$$X^{(\ell+1)} = \sigma\Big(D_V^{-\frac{1}{2}}\, H\, W\, D_E^{-1}\, H^\top\, D_V^{-\frac{1}{2}}\, X^{(\ell)}\, \Theta^{(\ell)}\Big), \qquad \ell = 0, 1, \ldots, L-1.$$

Definition 7 (n-SuperHyperGraph Neural Network (n-SHGNN))**.** *Let $H^{(n)} = \big(V^{(n)}, E^{(n)}\big)$ be an n-SuperHyperGraph over a finite base vertex set V_0 (so the ultimate objects of interest are the base vertices in V_0). Define the* expanded hypergraph *$H' = (V_0, E')$ by declaring that a subset $e' \subseteq V_0$ belongs to E' if and only if there exists a superhyperedge $e \in E^{(n)}$ such that*

$$e' = \bigcup_{v \in e} v.$$

(Here each $v \in e$ is treated as a subset of V_0 via the natural "flattening/union" operation.)

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 $v_i \in V_0$. Let $E' = \{e'_1, \dots, e'_{m'}\}$ and define the incidence matrix $H' \in \{0,1\}^{|V_0| \times m'}$ by

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

Let $w(e'_j) > 0$ be a learnable weight for e'_j, and set

$$W = \operatorname{diag}\big(w(e'_1), \dots, w(e'_{m'})\big) \in \mathbb{R}^{m' \times m'}.$$

Define diagonal degree matrices $D_V \in \mathbb{R}^{|V_0| \times |V_0|}$ and $D_E \in \mathbb{R}^{m' \times m'}$ by

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

Let $\Theta \in \mathbb{R}^{d \times c}$ be learnable and let $\sigma(\cdot)$ be a nonlinearity. A single n-SHGNN convolutional layer is the map

$$\text{n-SHGNN}_\Theta : \mathbb{R}^{|V_0| \times d} \to \mathbb{R}^{|V_0| \times c}, \qquad Y = \text{n-SHGNN}_\Theta(X) := \sigma\Big(D_V^{-\frac{1}{2}}\, H'\, W\, D_E^{-1}\, H'^{\top}\, D_V^{-\frac{1}{2}}\, X\, \Theta\Big),$$

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

4. Review and Results: HyperGraphRAG and SuperHyperGraphRAG

We examine HyperGraphRAG and SuperHyperGraphRAG. Retrieval-augmented generation combines external knowledge retrieval with neural text generation, grounding model outputs in relevant, up-to-date context for accurate responses[98–101]. GraphRAG is retrieval-augmented generation using a graph linking context chunks or entities; retrieval seeds similar nodes, expands along edges to gather connected evidence, serializes the subgraph, then an LLM generates answers [58,59,102–104].

Definition 8 (GraphRAG: graph-based retrieval-augmented generation)**.** *Let $\mathcal{D}$ be a finite corpus and let $\mathcal{C} = \{c_1, \dots, c_N\}$ be a finite set of* atomic contexts *(e.g., chunks, passages, tables, figure captions) extracted from $\mathcal{D}$. Let $\mathcal{Q}$ be a set of queries.*

A GraphRAG system *is a tuple*

$$\text{GRAG} = (\mathcal{C},\ \Phi,\ G,\ \text{Score}_V,\ \text{Expand},\ \text{Select},\ \text{Serialize},\ \mathcal{M}),$$

where:

(i) $\Phi : \mathcal{C} \cup \mathcal{Q} \to \mathbb{R}^d$ *is an embedding map.*

(ii) *$G = (V, E)$ is a finite graph whose vertex set V indexes retrievable units (e.g., $V = \mathcal{C}$, or $V = \mathcal{C} \cup \mathcal{E}$ with entities $\mathcal{E}$). Each edge $\{u, v\} \in E$ encodes a* binary relation *between u and v (e.g., co-mention, hyperlink, citation, temporal adjacency, or a knowledge-graph triple).*

(iii) $\mathrm{Score}_V : V \times \mathcal{Q} \to \mathbb{R}$ *is a vertex scoring function; a typical choice is*

$$\mathrm{Score}_V(v;q) = \frac{\langle \Phi(v), \Phi(q) \rangle}{\|\Phi(v)\| \cdot \|\Phi(q)\|} \quad (v \in V,\ q \in \mathcal{Q}),$$

with the convention that $\Phi(v)$ *means the embedding of the context (or entity) represented by* v.

(iv) $\mathrm{Expand} : \mathcal{P}(V) \times \mathcal{Q} \to \mathcal{P}(V)$ *is an expansion operator that enlarges a seed set of vertices using the graph structure (e.g., k-hop neighborhood, weighted random walk, or personalized PageRank).*

(v) $\mathrm{Select} : \mathcal{P}(V) \times \mathcal{Q} \to \mathcal{P}(V)$ *is a selection operator producing the final vertex set to be used as evidence.*

(vi) $\mathrm{Serialize} : (G, S) \mapsto$ *a finite text string converts a vertex-induced subgraph* $G[S]$ *(and optionally edge labels) into an LLM-readable context (e.g., ordered passages, entity summaries, relation triples).*

(vii) $\mathcal{M}$ *is a conditional generator (LLM) that maps* $(q, \mathrm{context})$ *to an answer.*

Given $q \in \mathcal{Q}$*, GraphRAG computes:*

(a) Seeding: *choose* $S_0(q) \subseteq V$ *(e.g., top-k vertices by* $\mathrm{Score}_V(\cdot;q)$*).*

(b) Graph expansion: *set* $S_1(q) = \mathrm{Expand}(S_0(q);q)$.

(c) Final selection: *set* $S(q) = \mathrm{Select}(S_1(q);q)$.

(d) Generation: *output*

$$\mathrm{GRAG}(q) \;=\; \mathcal{M}\big(q, \mathrm{Serialize}(G, S(q))\big).$$

Remark 1 (How GraphRAG differs from classical RAG)**.** *Classical RAG typically retrieves a* set *of contexts by similarity alone. GraphRAG additionally uses* E *to expand and organize evidence, so that retrieved contexts are not only similar to* q*, but also* structurally connected *via relations encoded in* G.

HyperGraphRAG is retrieval-augmented generation using a hypergraph, where vertices are context chunks and hyperedges encode multiway relations; retrieval expands through relevant hyperedges, selects evidence, then an LLM generates grounded answers [60].

Definition 9 (HyperGraphRAG: hypergraph-based retrieval-augmented generation)**.** *[60] Let* $\mathcal{D}, \mathcal{C}, \mathcal{Q}, \Phi$ *be as in Definition 8. A* HyperGraphRAG system *is a tuple*

$$\mathrm{HGRAG} = \big(\mathcal{C},\ \Phi,\ H,\ \mathrm{Score}_V,\ \mathrm{Score}_E,\ \mathrm{Expand}_H,\ \mathrm{Select}_H,\ \mathrm{Serialize}_H,\ \mathcal{M}\big),$$

where:

(i) $H = (V, \mathcal{E})$ *is a hypergraph (Definition 3) whose vertices* V *index retrievable units, and whose hyperedges* $\mathcal{E} \subseteq \mathcal{P}(V) \setminus \{\emptyset\}$ *encode* multiway relations*. A hyperedge* $e = \{v_1, \ldots, v_m\} \in \mathcal{E}$ *represents that* $v_1, \ldots, v_m$ *participate in a shared interaction (e.g., same event, same topic, same table row, same co-author group, or the same multi-entity mention).*

(ii) $\mathrm{Score}_V : V \times \mathcal{Q} \to \mathbb{R}$ *is a vertex scoring function (e.g., cosine similarity as in Definition 8).*

(iii) $\mathrm{Score}_E : \mathcal{E} \times \mathcal{Q} \to \mathbb{R}$ *is a hyperedge scoring function; one concrete choice is*

$$\mathrm{Score}_E(e;q) = \mathrm{Agg}\big(\{\mathrm{Score}_V(v;q) \mid v \in e\}\big),$$

where Agg *is a fixed aggregator such as* max*, mean, or a softmax-weighted mean.*

(iv) $\mathrm{Expand}_H : \mathcal{P}(V) \times \mathcal{Q} \to \mathcal{P}(V)$ *is a hypergraph expansion operator. A canonical expansion is:*

$$\mathrm{Expand}_H(S;q) \;=\; S \,\cup\, \bigcup \Big\{ e \in \mathcal{E} \;\Big|\; e \cap S \neq \emptyset \ \text{and} \ \ \mathrm{Score}_E(e;q) \geq \tau(q) \Big\},$$

for some threshold $\tau(q) \in \mathbb{R}$ *(possibly query-dependent).*

(v) $\mathrm{Select}_H : \mathcal{P}(V) \times \mathcal{Q} \to \mathcal{P}(V)$ *selects the final evidence vertices.*

(vi) $\mathrm{Serialize}_H$ *serializes the induced subhypergraph* $H[S]$ *(optionally including hyperedge descriptions) into a finite text context.*

(vii) $\mathcal{M}$ *is an LLM used for generation, as before.*

Given $q \in \mathcal{Q}$*, HyperGraphRAG computes:*

(a) Seeding: *choose* $S_0(q) \subseteq V$ *(e.g., top-k by* $\text{Score}_V(\cdot;q)$*).*

(b) Hyperedge-aware expansion: *set* $S_1(q) = \text{Expand}_H(S_0(q);q)$.

(c) Final selection: *set* $S(q) = \text{Select}_H(S_1(q);q)$.

(d) Generation: *output*

$$\text{HGRAG}(q) \;=\; \mathcal{M}(q, \text{Serialize}_H(H, S(q))).$$

Proposition 1 (GraphRAG is a special case of HyperGraphRAG)**.** *Assume that every hyperedge of* $H = (V, \mathcal{E})$ *has size* 2*, i.e.,*

$$\forall e \in \mathcal{E}, \quad |e| = 2.$$

Define the graph $G = (V, E)$ *by*

$$E = \{\{u,v\} \subseteq V \mid \{u,v\} \in \mathcal{E}\}.$$

Then any HyperGraphRAG instance on H induces a GraphRAG instance on G that yields the same expanded vertex sets, provided Expand *uses the same threshold rule via* $\text{Score}_E(\{u,v\};q)$ *and neighborhood expansion.*

Proof. If $|e| = 2$ for all $e \in \mathcal{E}$, then each hyperedge is of the form $e = \{u,v\}$ with $u, v \in V$, so $\mathcal{E}$ is in bijection with the edge set E of a simple graph $G = (V, E)$ defined above. For any seed set $S \subseteq V$,

$$\bigcup\{\, e \in \mathcal{E} \mid e \cap S \neq \emptyset \,\} = \bigcup\{\, \{u,v\} \in E \mid \{u,v\} \cap S \neq \emptyset \,\},$$

which is exactly the one-step graph-neighborhood expansion (up to the same score/threshold filter applied to edges). Therefore the expanded vertex sets coincide, and hence the selected substructures and serialized contexts can be chosen identical. ☐

SuperHyperGraphRAG is retrieval-augmented generation over an n-superhypergraph whose supervertices are nested context-sets, expanded via superedges; it generalizes GraphRAG and HyperGraphRAG by enabling hierarchical multiway retrieval and structured evidence serialization. As a reference, the comparison of GraphRAG, HyperGraphRAG, and SuperHyperGraphRAG is presented in Table 3.

Table 3. Concise comparison of GraphRAG, HyperGraphRAG, and SuperHyperGraphRAG.

Concept	Index structure	Unit of retrieval	What the structure helps capture
GraphRAG	Graph $G = (V, E)$	Relevant subgraph (nodes/edges) around the query entities	Pairwise relations (who-relates-to-whom), paths, neighborhoods, and evidence chains
HyperGraphRAG	Hypergraph $H = (V, \mathcal{E})$	Relevant sub-hypergraph (vertices and hyperedges matched to the query)	Higher-order relations among multiple entities at once (group facts, co-occurrence, multi-actor events)
SuperHyperGraphRAG	Superhypergraph with nested hyperrelations	Relevant nested substructure (hyperedges and hyperedges-of-hyperedges)	Multi-level higher-order structure (hierarchical groupings, topic → claim → evidence layers, nested interactions)

Definition 10 (Iterated powerset and flattening)**.** *Let V_0 be a finite nonempty base set of atomic contexts (chunks, passages, entity cards, etc.). Define $\mathrm{PS}^0(V_0) := V_0$ and $\mathrm{PS}^{n+1}(V_0) := \mathrm{PS}(\mathrm{PS}^n(V_0))$ for $n \geq 0$.*

For each $n \geq 0$, define a map

$$\mathrm{Flat}_n : \mathrm{PS}^n(V_0) \setminus \{\varnothing\} \longrightarrow \mathrm{PS}(V_0) \setminus \{\varnothing\}$$

recursively by

$$\mathrm{Flat}_0(x) := \{x\} \quad (x \in V_0), \qquad \mathrm{Flat}_{n+1}(X) := \bigcup_{Y \in X} \mathrm{Flat}_n(Y) \quad (X \in \mathrm{PS}^{n+1}(V_0) \setminus \{\varnothing\}).$$

Thus, each n-level object $v \in \mathrm{PS}^n(V_0)$ is a nested set of depth n, and $\mathrm{Flat}_n(v) \subseteq V_0$ is the set of atomic contexts contained in v after unnesting.

Remark 2 (Where the SuperHyperGraph idea enters)**.** *In SuperHyperGraphRAG, retrievable units are allowed to be* supervertices *$v \in \mathrm{PS}^n(V_0)$, so a single retrieved unit can represent a multi-level group of atomic contexts. Edges then connect these supervertices, enabling retrieval over nested (iterated-powerset) structures rather than only flat units.*

Definition 11 (SuperHyperGraphRAG)**.** *Fix a finite set V_0 of atomic contexts and a set $\mathcal{Q}$ of queries. Fix an integer $n \geq 0$. Let $\mathrm{SHG}^{(n)} = (V, E)$ be an n–SuperHyperGraph on V_0 in the sense of Definition 5, and assume $V \subseteq \mathrm{PS}^n(V_0) \setminus \{\varnothing\}$.*

A SuperHyperGraphRAG system *of depth n is a tuple*

$$\mathrm{SHGRAG}^{(n)} = \big(V_0, \mathcal{Q}, \Phi_0, \mathrm{SHG}^{(n)}, \mathrm{Score}_V, \mathrm{Score}_E, \mathrm{Expand}_n, \mathrm{Select}_n, \mathrm{Serialize}_n, \mathcal{M}\big),$$

where:

(i) $\Phi_0 : V_0 \cup \mathcal{Q} \to \mathbb{R}^d$ *is an embedding map on atomic contexts and queries.*

(ii) $\mathcal{M}$ *is a conditional generator (LLM) that maps a pair $(q, \text{context})$ to an output string.*

(iii) *The induced embedding of a supervertex $v \in V \subseteq \mathrm{PS}^n(V_0)$ is defined by the explicit average*

$$\Phi(v) := \frac{1}{|\,\mathrm{Flat}_n(v)|} \sum_{x \in \mathrm{Flat}_n(v)} \Phi_0(x) \in \mathbb{R}^d,$$

where Flat_n is from Definition 10.

(iv) *The vertex score is the cosine similarity*

$$\mathrm{Score}_V(v; q) := \frac{\langle \Phi(v), \Phi_0(q) \rangle}{\|\Phi(v)\| \cdot \|\Phi_0(q)\|} \quad (v \in V,\ q \in \mathcal{Q}),$$

assuming $\Phi(v) \neq 0$ and $\Phi_0(q) \neq 0$.

(v) *The superedge score uses a fixed aggregator* Agg *(already declared in the preamble):*

$$\mathrm{Score}_E(e; q) := \mathrm{Agg}\big(\{\mathrm{Score}_V(v; q) \mid v \in e\}\big) \quad (e \in E,\ q \in \mathcal{Q}).$$

(vi) *The expansion operator is*

$$\mathrm{Expand}_n(S; q) := S \cup \bigcup \Big\{ e \in E \ \Big|\ e \cap S \neq \varnothing \ \text{and} \ \mathrm{Score}_E(e; q) \geq \tau(q) \Big\},$$

for a threshold function $\tau : \mathcal{Q} \to \mathbb{R}$.

(vii) *The selection operator $\mathrm{Select}_n : \mathrm{PS}(V) \times \mathcal{Q} \to \mathrm{PS}(V)$ returns the final evidence set.*

(viii) *The serialization operator $\mathrm{Serialize}_n$ converts the induced sub-superhypergraph $\mathrm{SHG}^{(n)}[S]$ (with optional labels and summaries) into a finite text string.*

Given $q \in \mathcal{Q}$, the SuperHyperGraphRAG answer is computed by:

(a) *Seed $S_0(q) \subseteq V$ (e.g., top-k supervertices by $\mathrm{Score}_V(\cdot; q)$).*

(b) *Expand for $T \in \mathbb{N}$ steps:*

$$S_{t+1}(q) := \mathrm{Expand}_n(S_t(q); q) \qquad (t = 0, 1, \dots, T-1).$$

(c) *Select $S(q) := \mathrm{Select}_n(S_T(q); q)$.*

(d) *Generate*

$$\mathrm{SHGRAG}^{(n)}(q) := \mathcal{M}\big(q, \mathrm{Serialize}_n(\mathrm{SHG}^{(n)}, S(q))\big).$$

Theorem 1 (HyperGraphRAG is a special case of SuperHyperGraphRAG)**.** *Let $H = (V, \mathcal{E})$ be a hypergraph with $V = V_0$ (atomic contexts as vertices). Fix any HyperGraphRAG instance on H whose expansion rule is*

$$\mathrm{Expand}_H(S; q) = S \cup \bigcup \Big\{ e \in \mathcal{E} \;\Big|\; e \cap S \neq \emptyset \text{ and } \mathrm{Score}_E(e; q) \geq \tau(q) \Big\}.$$

Define an 0–SuperHyperGraph $\mathrm{SHG}^{(0)} = (V', E')$ by

$$V' := V_0, \qquad E' := \mathcal{E}.$$

Then the depth-0 SuperHyperGraphRAG system $\mathrm{SHGRAG}^{(0)}$ built on $\mathrm{SHG}^{(0)}$, with the same

$$\Phi_0, \mathrm{Agg}, \tau, \mathrm{Select}, \mathrm{Serialize}, \mathcal{M}$$

, produces exactly the same expanded sets and the same final generated output as the given HyperGraphRAG system, for every query $q \in \mathcal{Q}$ and every step $t \in \{0, \dots, T\}$.

Proof. Since $n = 0$, we have $\mathrm{PS}^0(V_0) = V_0$, hence $V' = V_0 \subseteq \mathrm{PS}^0(V_0)$. For any $v \in V'$, Definition 10 gives $\mathrm{Flat}_0(v) = \{v\}$. Therefore the induced supervertex embedding in Definition 11 satisfies

$$\Phi(v) = \frac{1}{|\mathrm{Flat}_0(v)|} \sum_{x \in \mathrm{Flat}_0(v)} \Phi_0(x) = \frac{1}{|\{v\}|} \sum_{x \in \{v\}} \Phi_0(x) = 1 \cdot \Phi_0(v) = \Phi_0(v).$$

Hence for all $v \in V'$ and $q \in \mathcal{Q}$,

$$\mathrm{Score}_V(v; q) = \frac{\langle \Phi(v), \Phi_0(q) \rangle}{\|\Phi(v)\| \cdot \|\Phi_0(q)\|} = \frac{\langle \Phi_0(v), \Phi_0(q) \rangle}{\|\Phi_0(v)\| \cdot \|\Phi_0(q)\|},$$

which is exactly the vertex scoring used by the HyperGraphRAG instance (under the shared choice of cosine scoring).

Now $E' = \mathcal{E}$, and for every hyperedge $e \in \mathcal{E}$ we also have $e \in E'$ and the same aggregator Agg, so the edge scores coincide:

$$\mathrm{Score}_E^{\mathrm{SHG}}(e; q) = \mathrm{Agg}(\{\mathrm{Score}_V(v; q) \mid v \in e\}) = \mathrm{Score}_E^{H}(e; q).$$

Consequently, for any $S \subseteq V' = V$ and any $q \in \mathcal{Q}$,

$$\begin{aligned}
\mathrm{Expand}_0(S; q) &= S \cup \bigcup \Big\{ e \in E' \;\Big|\; e \cap S \neq \emptyset \text{ and } \mathrm{Score}_E^{\mathrm{SHG}}(e; q) \geq \tau(q) \Big\} \\
&= S \cup \bigcup \Big\{ e \in \mathcal{E} \;\Big|\; e \cap S \neq \emptyset \text{ and } \mathrm{Score}_E^{H}(e; q) \geq \tau(q) \Big\} \\
&= \mathrm{Expand}_H(S; q).
\end{aligned}$$

By induction on t, starting from the same seed $S_0(q)$, we obtain $S_t^{\mathrm{SHG}}(q) = S_t^H(q)$ for all $t \leq T$. Since Select, Serialize, and $\mathcal{M}$ are assumed identical, the final output strings are identical as well. □

Theorem 2 (GraphRAG is a special case of SuperHyperGraphRAG)**.** *Let $G = (V, E_G)$ be a (finite, undirected) graph on atomic contexts $V = V_0$. Assume a GraphRAG instance whose expansion is realized by iterating one-step neighbor expansion T times, i.e.,*

$$N_G(S) := S \cup \{\, u \in V \mid \exists v \in S : \ \{u, v\} \in E_G \,\}, \qquad S_{t+1}(q) := N_G(S_t(q)).$$

Define a depth-0 superhypergraph $\mathrm{SHG}^{(0)} = (V', E')$ *by*

$$V' := V_0, \qquad E' := \{\{u, v\} \subseteq V_0 \mid \{u, v\} \in E_G\}.$$

Assume further that the SuperHyperGraphRAG expansion Expand_0 *uses $\tau(q) = -\infty$ (no threshold filtering), so that every incident edge expands the set. Then, for every query $q \in \mathcal{Q}$ and every $t \in \{0, \ldots, T\}$, the SuperHyperGraphRAG iterates satisfy*

$$S_t^{\mathrm{SHG}}(q) = S_t^G(q),$$

and hence the induced evidence substructures (and outputs, if Select, Serialize, $\mathcal{M}$ *coincide) match the GraphRAG instance.*

Proof. Because E' consists only of 2-element subsets, $\mathrm{SHG}^{(0)}$ is a 2-uniform hypergraph on V_0, and by construction it encodes exactly the same adjacencies as G.

Fix $S \subseteq V_0$. Since $\tau(q) = -\infty$, the expansion rule becomes

$$\mathrm{Expand}_0(S; q) = S \cup \bigcup \{\, e \in E' \mid e \cap S \neq \varnothing \,\}.$$

We prove the set equality

$$\mathrm{Expand}_0(S; q) = N_G(S).$$

First, let $x \in \mathrm{Expand}_0(S; q)$. Then either $x \in S$, which implies $x \in N_G(S)$, or else

$$x \in \bigcup \{\, e \in E' \mid e \cap S \neq \varnothing \,\},$$

so there exists $e \in E'$ such that $x \in e$ and $e \cap S \neq \varnothing$. Because $|e| = 2$, we may write $e = \{u, v\}$ with $\{u, v\} \in E_G$ and $x \in \{u, v\}$. Also $e \cap S \neq \varnothing$ means either $u \in S$ or $v \in S$. Hence x is either in S or adjacent in G to a vertex in S, so $x \in N_G(S)$. Thus $\mathrm{Expand}_0(S; q) \subseteq N_G(S)$.

Conversely, let $x \in N_G(S)$. If $x \in S$, then $x \in \mathrm{Expand}_0(S; q)$ holds immediately. Otherwise, by definition of $N_G(S)$, there exists $v \in S$ with $\{x, v\} \in E_G$. Then $e := \{x, v\} \in E'$, and $e \cap S \neq \varnothing$ since $v \in e \cap S$. Therefore $x \in e \subseteq \bigcup \{\, e \in E' \mid e \cap S \neq \varnothing \,\} \subseteq \mathrm{Expand}_0(S; q)$. So $N_G(S) \subseteq \mathrm{Expand}_0(S; q)$.

Combining both inclusions gives $\mathrm{Expand}_0(S; q) = N_G(S)$. Starting from the same seed $S_0(q)$, induction on t yields

$$S_{t+1}^{\mathrm{SHG}}(q) = \mathrm{Expand}_0(S_t^{\mathrm{SHG}}(q); q) = N_G(S_t^G(q)) = S_{t+1}^G(q),$$

hence $S_t^{\mathrm{SHG}}(q) = S_t^G(q)$ for all $t \leq T$. If Select, Serialize, $\mathcal{M}$ are shared between the two systems, the final outputs coincide as well. □

5. Review and Results: SuperHyperGraph Generation

Graph generation produces new graphs satisfying constraints by learning node–edge patterns, enabling molecule design, networks synthesis, and simulation [63–65,105,106]. Hypergraph generation produces new hypergraphs by learning multi-vertex interaction patterns, enabling group-relation synthesis in chemistry, text, and systems [107–110]. Superhypergraph generation produces nested higher-order hypergraphs by learning hyperedge-of-hyperedge patterns, enabling hierarchical interaction synthesis and complex structure design. The comparison of Graph Generation, HyperGraph Generation, and SuperHyperGraph Generation is presented in Table 4.

Definition 12 (Graph Generation (probabilistic / model-based))**.** *Fix a finite vertex set V and a measurable* condition space $\mathcal{C}$ *(e.g., labels, text, constraints, statistics). A* graph generator *is a triple*

$$\mathsf{Gen}_G = (\mathcal{Z}, \pi_\theta, g_\theta),$$

where

1. $\mathcal{Z}$ *is a latent (random-seed) space,*
2. π_θ *is a probability distribution on* $\mathcal{Z}$*,*
3. $g_\theta : \mathcal{Z} \times \mathcal{C} \to \mathcal{G}(V)$ *is a measurable map.*

Given $c \in \mathcal{C}$, graph generation conditioned on c *is the random output*

$$G \;=\; g_\theta(Z, c) \quad \text{where } Z \sim \pi_\theta.$$

Equivalently, Gen_G *induces a conditional distribution* $P_\theta(\,\cdot\,|c)$ *on* $\mathcal{G}(V)$ *by*

$$P_\theta(G \in A \mid c) \;=\; \pi_\theta\big(\{z \in \mathcal{Z} : \; g_\theta(z, c) \in A\}\big) \quad \text{for all measurable } A \subseteq \mathcal{G}(V).$$

Table 4. Concise comparison of Graph Generation, HyperGraph Generation, and SuperHyperGraph Generation.

Concept	Generated object	Core relations captured	Typical outputs / uses
Graph Generation	A graph $G = (V, E)$	Pairwise relations (edges) between two vertices	Synthetic networks; molecule graphs; link/structure completion
HyperGraph Generation	A hypergraph $H = (V, \mathcal{E})$ with $\mathcal{E} \subseteq \mathrm{PS}(V) \setminus \{\emptyset\}$	Higher-order relations (hyperedges) among multiple vertices	Group interactions; co-author/topic groups; multi-entity chemical interactions
SuperHyperGraph Generation	A superhypergraph with nested hyperrelations (hyperedge-of-hyperedge structures)	Multi-level higher-order relations (nested, hierarchical interactions)	Hierarchical group systems; layered knowledge structures; nested interaction synthesis

Definition 13 (Hypergraph Generation (probabilistic / model-based))**.** *Fix a finite vertex set V and a measurable condition space $\mathcal{C}$. A* hypergraph generator *is a triple*

$$\mathsf{Gen}_H = (\mathcal{Z}, \pi_\theta, h_\theta),$$

where $\mathcal{Z}$ is a latent space, π_θ is a probability distribution on $\mathcal{Z}$, and $h_\theta : \mathcal{Z} \times \mathcal{C} \to \mathcal{H}(V)$ is a measurable map. Given $c \in \mathcal{C}$, hypergraph generation conditioned on c *is the random output*

$$H \;=\; h_\theta(Z, c) \quad \text{where } Z \sim \pi_\theta,$$

which induces a conditional distribution on $\mathcal{H}(V)$ in the same way as above.

Definition 14 (Constraint-based (deterministic) Generation)**.** *Let V be a finite vertex set and let Φ be a predicate (constraint) on $\mathcal{G}(V)$. A* constraint-based graph generation procedure *is an algorithm* Alg *that, given Φ, outputs a graph $G \in \mathcal{G}(V)$ such that*

$$\Phi(G) = true.$$

The enumerative *variant outputs the entire solution set*

$$\mathrm{Sol}(\Phi) \;=\; \{G \in \mathcal{G}(V) : \; \Phi(G) = true\}.$$

Replacing $\mathcal{G}(V)$ by $\mathcal{H}(V)$ gives the corresponding notions for hypergraphs.

Definition 15 (n-SuperHypergraph generation (probabilistic / model-based))**.** *Fix $n \geq 0$, a finite, nonempty base set V_0, and a measurable condition space $\mathcal{C}$. Let*

$$\mathcal{SH}^{(n)}(V_0) \;:=\; \Big\{ \mathrm{SHG}^{(n)} = (V, E) \;\Big|\; V \subseteq \mathcal{P}^n_*(V_0),\; E \subseteq \mathcal{P}(V) \setminus \{\emptyset\} \Big\}$$

denote the set of all n-SuperHypergraphs on V_0.

*An n-*SuperHypergraph generator *is a triple*

$$\mathsf{Gen}_n = (\mathcal{Z}, \pi_\theta, g_\theta^{(n)}),$$

where

(i) $\mathcal{Z}$ *is a latent (random-seed) space,*

(ii) π_θ *is a probability distribution on* $\mathcal{Z}$*,*

(iii) $g_\theta^{(n)} : \mathcal{Z} \times \mathcal{C} \to \mathcal{SH}^{(n)}(V_0)$ *is a measurable map.*

Given $c \in \mathcal{C}$, generation conditioned on c *is the random output*

$$\mathrm{SHG}^{(n)} \;=\; g_\theta^{(n)}(Z, c) \qquad \text{where } Z \sim \pi_\theta.$$

Equivalently, Gen_n *induces a conditional distribution $P_\theta^{(n)}(\,\cdot\,|c)$ on $\mathcal{SH}^{(n)}(V_0)$ by*

$$P_\theta^{(n)}(A \mid c) \;=\; \pi_\theta\big(\{z \in \mathcal{Z} : \; g_\theta^{(n)}(z, c) \in A\}\big) \quad \text{for all measurable } A \subseteq \mathcal{SH}^{(n)}(V_0).$$

Definition 16 (Constraint-based (deterministic) n-SuperHypergraph generation)**.** *Fix $n \geq 0$ and V_0. Let Ψ be a predicate (constraint) on $\mathcal{SH}^{(n)}(V_0)$. A* constraint-based n-SuperHypergraph generation procedure *is an algorithm* Alg *that outputs* $\mathrm{SHG}^{(n)} \in \mathcal{SH}^{(n)}(V_0)$ *such that*

$$\Psi(\mathrm{SHG}^{(n)}) = true.$$

The enumerative *variant outputs the entire solution set*

$$\mathrm{Sol}(\Psi) \;=\; \{\mathrm{SHG}^{(n)} \in \mathcal{SH}^{(n)}(V_0) : \Psi(\mathrm{SHG}^{(n)}) = true\}.$$

Theorem 3 (Hypergraph generation is a special case of n-SuperHypergraph generation)**.** *Let V_0 be a finite vertex set, and let $\mathcal{H}(V_0)$ be the set of hypergraphs on V_0:*

$$\mathcal{H}(V_0) = \{(V_0, \mathcal{E}) \mid \mathcal{E} \subseteq \mathcal{P}(V_0) \setminus \{\emptyset\}\}.$$

Then $\mathcal{H}(V_0)$ is naturally identified with the subset of $\mathcal{SH}^{(0)}(V_0)$ consisting of those 0-SuperHyperGraphs $\mathrm{SHG}^{(0)} = (V, E)$ with $V = V_0$. Consequently, every hypergraph generator $\mathsf{Gen}_H = (\mathcal{Z}, \pi_\theta, h_\theta)$ on V_0 is an instance of a 0-SuperHypergraph generator $\mathsf{Gen}_0 = (\mathcal{Z}, \pi_\theta, g_\theta^{(0)})$.

Proof. By the Definition, $\mathcal{P}^0(V_0) = V_0$ and $\mathcal{P}^0_*(V_0) = V_0$ (since $V_0 \neq \emptyset$). Thus, by the Definition, a 0-SuperHypergraph on V_0 is a pair (V, E) with

$$V \subseteq V_0, \qquad E \subseteq \mathcal{P}(V) \setminus \{\emptyset\}.$$

If we impose the additional condition $V = V_0$, then the allowable edge families are exactly $\mathcal{E} \subseteq \mathcal{P}(V_0) \setminus \{\emptyset\}$. Hence the correspondence

$$(V_0, \mathcal{E}) \in \mathcal{H}(V_0) \quad \longleftrightarrow \quad \mathrm{SHG}^{(0)} = (V_0, \mathcal{E}) \in \mathcal{SH}^{(0)}(V_0)$$

is a bijection onto the subset $\{\mathrm{SHG}^{(0)} \in \mathcal{SH}^{(0)}(V_0) \mid V = V_0\}$.

Now let $\mathsf{Gen}_H = (\mathcal{Z}, \pi_\theta, h_\theta)$ be a hypergraph generator, so $h_\theta : \mathcal{Z} \times \mathcal{C} \to \mathcal{H}(V_0)$. Define $g_\theta^{(0)} : \mathcal{Z} \times \mathcal{C} \to \mathcal{SH}^{(0)}(V_0)$ by

$$g_\theta^{(0)}(z, c) := h_\theta(z, c),$$

viewing $h_\theta(z, c) = (V_0, \mathcal{E})$ as the 0-SuperHypergraph $(V_0, \mathcal{E})$. Then $g_\theta^{(0)}$ is measurable whenever h_θ is measurable, and the output distributions coincide. Therefore Gen_H is a special case of Gen_0. □

Theorem 4 (Graph generation is a special case of n-SuperHypergraph generation)**.** *Let V_0 be a finite vertex set and let $\mathcal{G}(V_0)$ be the set of simple graphs on V_0:*

$$\mathcal{G}(V_0) = \{(V_0, E_G) \mid E_G \subseteq \binom{V_0}{2}\}.$$

Then $\mathcal{G}(V_0)$ is naturally identified with the subset of $\mathcal{SH}^{(0)}(V_0)$ consisting of those 0-SuperHyperGraphs (V, E) with $V = V_0$ and $E \subseteq \binom{V_0}{2}$. Consequently, every graph generator is an instance of a 0-SuperHypergraph generator.

Proof. As in the proof of Theorem 3, a 0-SuperHypergraph with $V = V_0$ is of the form (V_0, E) where $E \subseteq \mathcal{P}(V_0) \setminus \{\emptyset\}$. If we additionally require $E \subseteq \binom{V_0}{2}$, then each edge is a 2-element subset of V_0, so (V_0, E) is precisely a simple undirected graph on V_0. Thus the map

$$(V_0, E_G) \in \mathcal{G}(V_0) \quad \longmapsto \quad (V_0, E_G) \in \mathcal{SH}^{(0)}(V_0)$$

is a bijection onto $\{\mathrm{SHG}^{(0)} \in \mathcal{SH}^{(0)}(V_0) \mid V = V_0,\ E \subseteq \binom{V_0}{2}\}$.

Given any graph generator $(\mathcal{Z}, \pi_\theta, \gamma_\theta)$ with $\gamma_\theta : \mathcal{Z} \times \mathcal{C} \to \mathcal{G}(V_0)$, define $g_\theta^{(0)}$ by the same output map viewed in $\mathcal{SH}^{(0)}(V_0)$:

$$g_\theta^{(0)}(z, c) := \gamma_\theta(z, c).$$

Then the induced output distributions coincide, so graph generation is a special case of 0-SuperHypergraph generation. □

6. Review and Results: Causal SuperHyperGraph

A causal graph is a directed graph that represents direct causal dependencies among variables, usually derived from structural equations and used to define do-interventions and identify causal effects[111–115]. A causal hypergraph extends this idea by using directed hyperedges so that a set of variables can jointly act as a single causal mechanism on a target variable[116–119]. A causal superhypergraph further extends causal hypergraphs by allowing vertices to be nested set-objects obtained via iterated powersets, enabling hierarchical multiway causal mechanisms across multiple abstraction levels.

Table 5. Concise comparison: causal graph vs. causal hypergraph vs. causal superhypergraph.

Aspect	Causal Graph	Causal HyperGraph	Causal SuperHyperGraph
Underlying object	Directed graph $G = (V, E)$ with $E \subseteq V \times V$	Hypergraph $H = (V, \mathcal{E})$ with $\mathcal{E} \subseteq \mathcal{P}(V) \setminus \{\emptyset\}$	n–SuperHyperGraph $\mathrm{SHG}^{(n)} = (V, E)$ with $V \subseteq \mathcal{P}^n(V_0)$ and $E \subseteq \mathcal{P}(V) \setminus \{\emptyset\}$
Causal primitive	Edge $j \to i$ (direct cause)	Hyperedge $e \in \mathcal{E}$ (multiway relation), optionally oriented $e \Rightarrow v$	Superedge $\varepsilon \in E$ among supervertices (nested groups), optionally oriented across levels
Structural equation form	$X_i = f_i\big((X_j)_{j \in \mathrm{Pa}(i)}, U_i\big)$	$X_v = f_v\big((X_u)_{u \in N_H(v)}, U_v\big)$ or $X_{e \Rightarrow v} = f_{e \Rightarrow v}((X_u)_{u \in e}, U_{e \Rightarrow v})$	$Y_s = F_s\big((Y_t)_{t \in N_{\mathrm{SHG}}(s)}, U_s\big)$ for supervertices $s \in V$
Independence reading	d-separation on G	Separation via hyperedge-based paths / factorization over $\mathcal{E}$ (model-dependent)	Separation on superpaths and level-wise neighborhoods; supports hierarchical blocking (model-dependent)
What it represents	Pairwise directed influence	Multiway (group) influence in one step	Multiway influence with explicit nesting / hierarchy of groups and relations
Reduction to simpler model	—	If all hyperedges have size 2 and are oriented, reduces to a causal graph	If $n = 0$ and supervertices are singletons, reduces to a causal hypergraph; if also edges are size 2, reduces to a causal graph

Definition 17 (Structural Causal Model (SCM))**.** *Fix a finite index set of endogenous variables* $V = \{1, \ldots, n\}$*. For each* $i \in V$*, let* $\mathcal{X}_i$ *be the state space of* X_i*. Let* $U = \{1, \ldots, n\}$ *index exogenous noises, with spaces* $\mathcal{U}_i$*.*

A structural causal model *is a quadruple*

$$M = (\mathcal{U}, \mathcal{X}, F, P_U),$$

where

1. $\mathcal{U} = \prod_{i \in V} \mathcal{U}_i$ *and* $\mathcal{X} = \prod_{i \in V} \mathcal{X}_i$*;*
2. P_U *is a probability distribution on* $\mathcal{U}$*;*
3. $F = \{f_i\}_{i \in V}$ *is a family of measurable maps*

$$f_i : \Big(\prod_{j \in \mathrm{Pa}(i)} \mathcal{X}_j\Big) \times \mathcal{U}_i \longrightarrow \mathcal{X}_i,$$

with some parent set $\mathrm{Pa}(i) \subseteq V \setminus \{i\}$*.*

The endogenous variables $X = (X_i)_{i \in V}$ *are determined by the structural equations*

$$X_i = f_i\big((X_j)_{j \in \mathrm{Pa}(i)}, U_i\big) \qquad (i \in V),$$

where $U = (U_i)_{i \in V} \sim P_U$*.*

Definition 18 (Causal graph induced by an SCM)**.** *Let* $M = (\mathcal{U}, \mathcal{X}, F, P_U)$ *be an SCM with endogenous index set* V*. The* causal graph *(also called the* functional graph*) of* M *is the directed graph*

$$G_M = (V, E)$$

whose edge set is

$$E = \{\, (j, i) \in V \times V \mid j \in \mathrm{Pa}(i) \,\}.$$

We write $j \to i$ *for* $(j, i) \in E$ *and interpret* $j \to i$ *as a* direct causal dependence *of* X_i *on* X_j *in the model* M*. If* G_M *is acyclic, then* M *is called* recursive *(or* acyclic*).*

Definition 19 (Intervention (do-operator))**.** *Let $I \subseteq V$ and fix values $x_I \in \prod_{i\in I} \mathcal{X}_i$. The intervention* $\mathrm{do}(X_I = x_I)$ *produces the* intervened SCM

$$M^{\mathrm{do}(I=x_I)} \;=\; (\mathcal{U}, \mathcal{X}, F', P_U),$$

where

$$f'_i \;=\; \begin{cases} (\,\cdot\,) \mapsto x_i, & i \in I, \\ f_i, & i \notin I. \end{cases}$$

The corresponding interventional distribution $P(X \mid \mathrm{do}(X_I = x_I))$ is the distribution of the unique solution X (when it exists) under the modified equations and $U \sim P_U$.

Definition 20 (Directed hypergraph)**.** *A* directed hypergraph *is a pair*

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

where V is a finite node set and $\mathcal{E}$ is a finite family of directed hyperedges*. Each $e \in \mathcal{E}$ is an ordered pair*

$$e \;=\; \big(T(e) \to h(e)\big),$$

where the tail *$T(e) \subseteq V \setminus \{h(e)\}$ is nonempty and the* head *$h(e) \in V$.*

Definition 21 (Causal hypergraph model)**.** *Fix state spaces $\{\mathcal{X}_i\}_{i\in V}$ and noise spaces $\{\mathcal{U}_i\}_{i\in V}$. A* causal hypergraph model *is a tuple*

$$\mathfrak{M} \;=\; (\,\mathcal{U},\, \mathcal{X},\, H,\, \Phi,\, G,\, P_U\,),$$

where

1. $\mathcal{U} = \prod_{i\in V} \mathcal{U}_i$, $\mathcal{X} = \prod_{i\in V} \mathcal{X}_i$, *and P_U is a distribution on $\mathcal{U}$;*
2. $H = (V, \mathcal{E})$ *is a directed hypergraph;*
3. $\Phi = \{\varphi_e\}_{e\in\mathcal{E}}$ *is a family of* hyperedge mechanisms

$$\varphi_e : \prod_{j\in T(e)} \mathcal{X}_j \longrightarrow \mathcal{Z}_e,$$

for some intermediate spaces $\mathcal{Z}_e$;

4. $G = \{g_i\}_{i\in V}$ *is a family of* aggregation functions

$$g_i : \Big(\prod_{e\in\mathcal{E}_i} \mathcal{Z}_e\Big) \times \mathcal{U}_i \longrightarrow \mathcal{X}_i, \qquad \mathcal{E}_i := \{e \in \mathcal{E} \mid h(e) = i\}.$$

The structural equations of $\mathfrak{M}$ are

$$X_i \;=\; g_i\Big(\big(\varphi_e(X_{T(e)})\big)_{e\in\mathcal{E}_i},\, U_i\Big) \qquad (i \in V),$$

where $X_{T(e)} := (X_j)_{j\in T(e)}$ and $U \sim P_U$. Intuitively, each hyperedge $T(e) \to i$ represents a joint causal mechanism *acting on the tuple $X_{T(e)}$.*

Definition 22 (Graph induced by a causal hypergraph)**.** *Let $H = (V, \mathcal{E})$ be a directed hypergraph. Its* 2-section causal graph *(edge-expansion) is the directed graph*

$$\Gamma(H) \;=\; (V, E_\Gamma), \qquad E_\Gamma \;=\; \{\,(j,i) \mid \exists e \in \mathcal{E} \text{ s.t. } i = h(e) \text{ and } j \in T(e)\,\}.$$

Thus every hyperedge $T \to i$ is expanded into ordinary directed edges $j \to i$ for all $j \in T$.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 24 December 2025 doi:10.20944/preprints202512.2233.v1

Theorem 5 (Causal graphs are a special case of causal hypergraphs)**.** *Let $G = (V, E)$ be a directed graph. Define a directed hypergraph $H_G = (V, \mathcal{E}_G)$ by*

$$\mathcal{E}_G \;=\; \{\, (\{j\} \to i) \mid (j,i) \in E \,\}.$$

Then:

1. $\Gamma(H_G) = G$.
2. *For any SCM M with causal graph $G_M = G$, there exists a causal hypergraph model $\mathfrak{M}$ on H_G whose induced edge-expansion graph equals G, and whose structural equations coincide with those of M.*

Proof. (1) By construction, a directed edge $(j,i) \in E$ corresponds to the hyperedge $\{j\} \to i \in \mathcal{E}_G$. Hence $(j,i) \in E_\Gamma$ if and only if $(j,i) \in E$, so $\Gamma(H_G) = G$.

(2) Let $M = (\mathcal{U}, \mathcal{X}, F, P_U)$ be an SCM with

$$X_i \;=\; f_i\big((X_j)_{j\in \mathrm{Pa}(i)}, U_i\big), \qquad \mathrm{Pa}(i) = \{j \mid (j,i) \in E\}.$$

Define $\mathfrak{M}$ on H_G as follows. For each hyperedge $e = (\{j\} \to i)$, set $\mathcal{Z}_e := \mathcal{X}_j$ and

$$\varphi_{(\{j\}\to i)}(x_j) := x_j.$$

For each $i \in V$, list the incoming hyperedges as $\mathcal{E}_i = \{(\{j\} \to i) \mid j \in \mathrm{Pa}(i)\}$, identify $\prod_{e\in\mathcal{E}_i} \mathcal{Z}_e \cong \prod_{j\in\mathrm{Pa}(i)} \mathcal{X}_j$, and set

$$g_i\Big((z_e)_{e\in\mathcal{E}_i}, u_i\Big) := f_i\big((z_{(\{j\}\to i)})_{j\in\mathrm{Pa}(i)}, u_i\big).$$

Then the structural equation of $\mathfrak{M}$ becomes

$$X_i = g_i\Big((\varphi_{(\{j\}\to i)}(X_j))_{j\in\mathrm{Pa}(i)}, U_i\Big) = g_i\Big((X_j)_{j\in\mathrm{Pa}(i)}, U_i\Big) = f_i\big((X_j)_{j\in\mathrm{Pa}(i)}, U_i\big),$$

which is exactly the SCM equation. The induced graph statement follows from (1). □

Definition 23 (Granger-causal hypergraph (time-series causal hypergraph))**.** *Let $Y_t = (Y_t^{(1)}, \ldots, Y_t^{(d)})$ be a multivariate time series and fix a maximum lag $K \in \mathbb{N}$. Form a node set of lagged variables*

$$V \;=\; \{\, Y_{t-k}^{(j)} \mid j \in \{1, \ldots, d\},\; k \in \{1, \ldots, K\} \,\}.$$

For a target component $Y_t^{(i)}$, let $C_i \subseteq V$ be a nonempty set of lagged predictors. We say C_i Granger-causes *$Y_t^{(i)}$ (relative to the history V) if, in a chosen model class, using C_i improves the prediction of $Y_t^{(i)}$ beyond what is achievable without C_i (e.g., via likelihood-ratio tests, information criteria, or out-of-sample error reduction).*

A Granger-causal hypergraph *is a directed hypergraph $H = (V, \mathcal{E})$ whose hyperedges are*

$$e_i \;=\; (C_i \to Y_t^{(i)}),$$

interpreted as a multi-source temporal influence relation. (Granger causality is predictive/temporal and does not, by itself, assert interventional causality.)

Definition 24 (Directed n-SuperHyperGraph)**.** *Fix $n \geq 0$ and a finite, nonempty base set V_0. A* directed n-SuperHyperGraph *on V_0 is a pair*

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

such that

$$V \subseteq \mathrm{PS}_*^n(V_0), \qquad \mathcal{E} \subseteq \Big(\mathrm{PS}(V) \setminus \{\emptyset\}\Big) \times V.$$

Each directed superhyperedge $e \in \mathcal{E}$ is written as

$$e = \bigl(\mathrm{Tail}(e) \to \mathrm{Head}(e)\bigr),$$

where $\mathrm{Tail}(e) \subseteq V \setminus \{\mathrm{Head}(e)\}$ *is nonempty and* $\mathrm{Head}(e) \in V$. *Elements of V are n-*supervertices. *The pair* $(\mathrm{Tail}(e), \mathrm{Head}(e))$ *encodes a multi-source directed relation.*

Remark 3 (Use of the SuperHyperGraph idea)**.** *Definition 24 uses the SuperHyperGraph viewpoint by allowing vertices to be nested objects $v \in \mathrm{PS}^n(V_0)$ (iterated powerset level n). For $n = 0$, one recovers ordinary vertices $V \subseteq V_0$.*

Definition 25 (Causal n-SuperHyperGraph model)**.** *Fix $n \geq 0$ and a directed n-SuperHyperGraph $\overrightarrow{\mathrm{SHG}}^{(n)} = (V, \mathcal{E})$ on V_0. For each $v \in V$, let $\mathcal{X}_v$ be a measurable state space and $\mathcal{U}_v$ a measurable noise space. Set $\mathcal{X} := \prod_{v \in V} \mathcal{X}_v$ and $\mathcal{U} := \prod_{v \in V} \mathcal{U}_v$.*

A causal n-SuperHyperGraph model *is a tuple*

$$\mathfrak{M}^{(n)} = \bigl(\mathcal{U}, \mathcal{X}, \overrightarrow{\mathrm{SHG}}^{(n)}, \Phi, G, P_{\mathcal{U}}\bigr),$$

where:

(i) $P_{\mathcal{U}}$ *is a probability distribution on $\mathcal{U}$.*

(ii) $\Phi = \{\varphi_e\}_{e \in \mathcal{E}}$ *is a family of* superhyperedge mechanisms *such that for each* $e = (\mathrm{Tail}(e) \to \mathrm{Head}(e)) \in \mathcal{E}$ *there exists a measurable space $\mathcal{Z}_e$ and a measurable map*

$$\varphi_e : \prod_{u \in \mathrm{Tail}(e)} \mathcal{X}_u \longrightarrow \mathcal{Z}_e.$$

(iii) $G = \{g_v\}_{v \in V}$ *is a family of* aggregation functions *where, for each $v \in V$, letting*

$$\mathcal{E}_v := \{\, e \in \mathcal{E} \mid \mathrm{Head}(e) = v \,\},$$

we require a measurable map

$$g_v : \Bigl(\prod_{e \in \mathcal{E}_v} \mathcal{Z}_e\Bigr) \times \mathcal{U}_v \longrightarrow \mathcal{X}_v.$$

The endogenous variables $X = (X_v)_{v \in V}$ are determined by the structural equations

$$X_v = g_v\Bigl(\bigl(\varphi_e(X_{\mathrm{Tail}(e)})\bigr)_{e \in \mathcal{E}_v}, U_v\Bigr) \qquad (v \in V),$$

where $X_{\mathrm{Tail}(e)} := (X_u)_{u \in \mathrm{Tail}(e)}$ *and* $U = (U_v)_{v \in V} \sim P_{\mathcal{U}}$.

Definition 26 (Intervention (do-operator) on a causal n-SuperHyperGraph)**.** *Let $\mathfrak{M}^{(n)}$ be as in Definition 25. Fix a nonempty index set $I \subseteq V$ and values $x_I \in \prod_{v \in I} \mathcal{X}_v$. The intervention* $\mathrm{do}(X_I = x_I)$ *produces the intervened model*

$$\bigl(\mathfrak{M}^{(n)}\bigr)^{\mathrm{do}(I = x_I)} = \bigl(\mathcal{U}, \mathcal{X}, \overrightarrow{\mathrm{SHG}}^{(n)}, \Phi, G', P_{\mathcal{U}}\bigr),$$

where g'_v is defined by

$$g'_v = \begin{cases} (\,\cdot\,) \mapsto x_v, & v \in I, \\ g_v, & v \notin I. \end{cases}$$

The interventional distribution $P(X \mid \mathrm{do}(X_I = x_I))$ *is the distribution of the solution under G' and $U \sim P_{\mathcal{U}}$, when a (measurable) solution exists.*

Theorem 6 (Causal hypergraphs are a special case ($n = 0$))**.** *Let $H = (V_0, \mathcal{E}_H)$ be a directed hypergraph whose hyperedges are pairs $(T \to i)$ with $T \subseteq V_0 \setminus \{i\}$, $T \neq \emptyset$, and $i \in V_0$. Consider any causal hypergraph model on H given by structural equations*

$$X_i = g_i\Big(\big(\varphi_e(X_{\mathrm{Tail}(e)})\big)_{e \in \mathcal{E}_{H,i}}, U_i\Big), \qquad \mathcal{E}_{H,i} := \{e \in \mathcal{E}_H \mid \mathrm{Head}(e) = i\}.$$

Then it is a special case of a causal 0-SuperHyperGraph model (Definition 25) by taking $V := V_0 \subseteq \mathrm{PS}^0_*(V_0)$ and $\mathcal{E} := \mathcal{E}_H$.

Proof. Since $\mathrm{PS}^0(V_0) = V_0$ and $V_0 \neq \emptyset$, we have $\mathrm{PS}^0_*(V_0) = V_0$. Hence $V := V_0$ satisfies $V \subseteq \mathrm{PS}^0_*(V_0)$, so $(V, \mathcal{E}_H)$ is a directed 0-SuperHyperGraph in the sense of Definition 24. Keeping the same state spaces $\mathcal{X}_i$, noise spaces $\mathcal{U}_i$, distribution P_U, mechanisms φ_e, and aggregators g_i, the structural equations in Definition 25 become

$$X_i = g_i\Big(\big(\varphi_e(X_{\mathrm{Tail}(e)})\big)_{e \in \mathcal{E}_i}, U_i\Big) \quad \text{with} \quad \mathcal{E}_i = \{e \in \mathcal{E}_H \mid \mathrm{Head}(e) = i\},$$

which is exactly the given causal hypergraph model. Therefore the causal hypergraph model is a special case. □

Theorem 7 (Causal graphs (SCMs) are a special case ($n = 0$, singleton tails))**.** *Let $G = (V_0, E_G)$ be a directed graph and let an SCM be given by*

$$X_i = f_i\big((X_j)_{j \in \mathrm{Pa}(i)}, U_i\big), \qquad \mathrm{Pa}(i) := \{j \in V_0 \mid (j, i) \in E_G\}.$$

Define a directed hypergraph $H_G = (V_0, \mathcal{E}_G)$ by

$$\mathcal{E}_G := \{\, (\{j\} \to i) \mid (j, i) \in E_G \,\}.$$

Then the SCM is a special case of a causal 0-SuperHyperGraph model on $\overrightarrow{\mathrm{SHG}}^{(0)} = (V_0, \mathcal{E}_G)$.

Proof. By $\mathrm{PS}^0_*(V_0) = V_0$, the pair $(V_0, \mathcal{E}_G)$ is a directed 0-SuperHyperGraph. We now construct $\Phi = \{\varphi_e\}$ and $G = \{g_i\}$ so that the 0-SuperHyperGraph equations coincide with the SCM equations.

For each edge $(j, i) \in E_G$, let $e = (\{j\} \to i) \in \mathcal{E}_G$ and set $\mathcal{Z}_e := \mathcal{X}_j$ and

$$\varphi_e(x_j) := x_j \qquad (x_j \in \mathcal{X}_j).$$

For a fixed $i \in V_0$, the incoming hyperedges are

$$\mathcal{E}_i = \{(\{j\} \to i) \mid j \in \mathrm{Pa}(i)\}.$$

Hence

$$\prod_{e \in \mathcal{E}_i} \mathcal{Z}_e = \prod_{j \in \mathrm{Pa}(i)} \mathcal{X}_j.$$

Define g_i by

$$g_i\Big((z_e)_{e \in \mathcal{E}_i}, u_i\Big) := f_i\big((z_{(\{j\} \to i)})_{j \in \mathrm{Pa}(i)}, u_i\big).$$

Then the structural equation in Definition 25 becomes

$$\begin{aligned} X_i &= g_i\Big(\big(\varphi_{(\{j\} \to i)}(X_j)\big)_{j \in \mathrm{Pa}(i)}, U_i\Big) \\ &= g_i\Big((X_j)_{j \in \mathrm{Pa}(i)}, U_i\Big) \\ &= f_i\big((X_j)_{j \in \mathrm{Pa}(i)}, U_i\big), \end{aligned}$$

which is exactly the SCM equation. Therefore the causal graph (SCM) is a special case of a causal 0-SuperHyperGraph model. □

7. Review and Results: SuperHyperGraph Convolutional Network

GCN aggregates each node's features from its neighbors via normalized adjacency, then applies a learnable linear transform and nonlinearity per layer [120–124]. HGCN aggregates vertex features through hyperedges via an incidence-based diffusion operator, then applies learnable transformations to model multiway interactions [69,71,76,125]. SHGCN propagates features on supervertices using super-incidence across nested superedges, enabling hierarchical multiway diffusion and generalizing both GCN and HGCN. Table 6 presents the comparison among GCN, HGCN, and SHGCN.

Table 6. Concise comparison: GCN vs. HGCN vs. SHGCN.

Aspect	GCN	HGCN	SHGCN
Underlying structure	Graph $G = (V, E)$	Hypergraph $H = (V, \mathcal{E})$	n–SuperHyperGraph $\mathrm{SHG}^{(n)} = (V, E)$ with $V \subseteq \mathcal{P}^n(V_0)$
Adjacency / incidence	Adjacency matrix $A \in \{0,1\}^{\lvert V\rvert \times \lvert V\rvert}$	Incidence matrix $H \in \{0,1\}^{\lvert V\rvert \times \lvert\mathcal{E}\rvert}$	Super-incidence $\mathbf{B} \in \{0,1\}^{\lvert V\rvert \times \lvert E\rvert}$ between supervertices and superedges
Propagation operator	$\widehat{A} = \widetilde{D}^{-1/2}\widetilde{A}\widetilde{D}^{-1/2}$, $\widetilde{A} = A + I$	$\widehat{M} = D_v^{-1/2}\, H\, W\, D_e^{-1}\, H^\top\, D_v^{-1/2}$	$\widehat{M}^{(n)} = \Delta_v^{-1/2}\, \mathbf{B}\, W\, \Delta_e^{-1}\, \mathbf{B}^\top\, \Delta_v^{-1/2}$
One layer (typical)	$X^{(\ell+1)} = \sigma\big(\widehat{A}\, X^{(\ell)}\, W^{(\ell)}\big)$	$X^{(\ell+1)} = \sigma\big(\widehat{M}\, X^{(\ell)}\, W^{(\ell)}\big)$	$X^{(\ell+1)} = \sigma\big(\widehat{M}^{(n)}\, X^{(\ell)}\, W^{(\ell)}\big)$
Interaction modeled	Pairwise message passing along edges	Multiway aggregation through hyperedges	Multiway aggregation through superedges among nested (multi-level) supervertices
Specialization / reduction	—	If every hyperedge has size 2, becomes a graph model and aligns with GCN-style propagation	If $n = 0$, supervertices are singletons and SHGCN reduces to an HGCN on a hypergraph; if hyperedges are size 2, further reduces to GCN

Definition 27 (Graph and node features)**.** *Let $G = (V, E)$ be a finite (undirected) graph with $|V| = n$. Let $A \in \{0,1\}^{n\times n}$ be its adjacency matrix, and let $X \in \mathbb{R}^{n\times d_0}$ be a node-feature matrix whose ith row $x_i^\top$ is the feature vector of vertex $v_i \in V$. Define the self-loop augmented adjacency*

$$\widetilde{A} \;=\; A + I_n,$$

and its degree matrix $\widetilde{D} \in \mathbb{R}^{n\times n}$ by

$$\widetilde{D}_{ii} \;=\; \sum_{j=1}^{n} \widetilde{A}_{ij}.$$

Definition 28 (One GCN layer)**.** *Fix an activation function $\sigma : \mathbb{R} \to \mathbb{R}$ applied entrywise. A* graph convolutional layer *maps $X^{(\ell)} \in \mathbb{R}^{n\times d_\ell}$ to $X^{(\ell+1)} \in \mathbb{R}^{n\times d_{\ell+1}}$ by*

$$X^{(\ell+1)} \;=\; \sigma\Big(\widetilde{D}^{-1/2}\,\widetilde{A}\,\widetilde{D}^{-1/2}\,X^{(\ell)}\,W^{(\ell)}\Big),$$

where $W^{(\ell)} \in \mathbb{R}^{d_\ell\times d_{\ell+1}}$ is a learnable weight matrix. A GCN *is a composition of such layers (optionally followed by task-specific readout).*

Remark 4 (Interpretation)**.** *The operator $\widetilde{D}^{-1/2}\widetilde{A}\widetilde{D}^{-1/2}$ performs degree-normalized neighborhood averaging (including self-loops), so each layer mixes information along edges and then applies a learnable linear map.*

Definition 29 (Weighted hypergraph and incidence matrix)**.** *A (finite) weighted hypergraph is a triple $H = (V, \mathcal{E}, w)$ where $V = \{v_1, \ldots, v_n\}$ is a finite vertex set, $\mathcal{E} = \{e_1, \ldots, e_m\}$ is a finite family of nonempty subsets* *of V (hyperedges), and $w : \mathcal{E} \to \mathbb{R}_{>0}$ assigns positive weights. The* incidence matrix *is $H_{\mathrm{inc}} \in \{0,1\}^{n\times m}$ defined by*

$$(H_{\mathrm{inc}})_{i\alpha} \;=\; \begin{cases} 1, & v_i \in e_\alpha, \\ 0, & v_i \notin e_\alpha. \end{cases}$$

Let $W \in \mathbb{R}^{m\times m}$ be diagonal with $W_{\alpha\alpha} = w(e_\alpha)$. Define the vertex-degree matrix $D_v \in \mathbb{R}^{n\times n}$ and the hyperedge-degree matrix $D_e \in \mathbb{R}^{m\times m}$ by

$$(D_v)_{ii} \;=\; \sum_{\alpha=1}^{m} W_{\alpha\alpha}(H_{\mathrm{inc}})_{i\alpha}, \qquad (D_e)_{\alpha\alpha} \;=\; \sum_{i=1}^{n}(H_{\mathrm{inc}})_{i\alpha} \;=\; |e_\alpha|.$$

Definition 30 (One HGCN layer (spectral/message-passing form))**.** *Let $X^{(\ell)} \in \mathbb{R}^{n\times d_\ell}$ be vertex features at layer ℓ. A* hypergraph convolutional layer *maps $X^{(\ell)}$ to $X^{(\ell+1)}$ by*

$$X^{(\ell+1)} \;=\; \sigma\Big(D_v^{-1/2}\, H_{\mathrm{inc}}\, W\, D_e^{-1}\, H_{\mathrm{inc}}^{\top}\, D_v^{-1/2}\, X^{(\ell)}\, \Theta^{(\ell)}\Big),$$

where $\Theta^{(\ell)} \in \mathbb{R}^{d_\ell \times d_{\ell+1}}$ is learnable and σ is applied entrywise. An HGCN *is a composition of such hypergraph convolutional layers (optionally with readout).*

Remark 5 (Two-stage aggregation: vertex→hyperedge→vertex)**.** *The matrix product $H_{\mathrm{inc}}^{\top}X^{(\ell)}$ aggregates vertex features to hyperedges, and then $H_{\mathrm{inc}}(\cdot)$ propagates hyperedge information back to vertices. Thus one layer performs a normalized diffusion through hyperedges, capturing multiway interactions.*

Remark 6 (Row-normalized variant)**.** *A commonly used simplification replaces the symmetric normalization by*

$$X^{(\ell+1)} \;=\; \sigma\Big(D_v^{-1}\, H_{\mathrm{inc}}\, W\, D_e^{-1}\, H_{\mathrm{inc}}^{\top}\, X^{(\ell)}\, \Theta^{(\ell)}\Big),$$

which still implements the vertex→hyperedge→vertex propagation with degree control.

Definition 31 (Iterated powersets and flattening)**.** *Let V_0 be a finite nonempty set. Define*

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

Define the flattening maps $\mathrm{Flat}_n : \mathcal{P}^n(V_0) \to \mathcal{P}(V_0)$ recursively by

$$\mathrm{Flat}_0(x) := \{x\} \quad (x \in V_0), \qquad \mathrm{Flat}_{n+1}(X) := \bigcup_{Y\in X} \mathrm{Flat}_n(Y) \quad (X \in \mathcal{P}^{n+1}(V_0)).$$

Definition 32 (n-SuperHyperGraph and super-incidence)**.** *Fix $n \ge 0$ and a base set V_0. An* n-SuperHyperGraph *is a pair*

$$\mathrm{SHG}^{(n)} = (V, E), \qquad V \subseteq \mathcal{P}^n(V_0), \quad E \subseteq \mathcal{P}(V) \setminus \{\emptyset\}.$$

Write $V = \{v_1, \dots, v_N\}$ and $E = \{e_1, \dots, e_M\}$.

The super-incidence matrix *$H^{(n)} \in \{0,1\}^{N\times M}$ is defined by*

$$H^{(n)}_{i\alpha} := \begin{cases} 1, & v_i \in e_\alpha, \\ 0, & v_i \notin e_\alpha. \end{cases}$$

Let $W^{(n)} = \mathrm{diag}(w_1, \dots, w_M)$ be a diagonal matrix of positive superedge weights.

Define degree matrices $D_v^{(n)} \in \mathbb{R}^{N\times N}$ *and* $D_e^{(n)} \in \mathbb{R}^{M\times M}$ *by*

$$(D_v^{(n)})_{ii} := \sum_{\alpha=1}^{M} w_\alpha H_{i\alpha}^{(n)}, \qquad (D_e^{(n)})_{\alpha\alpha} := \sum_{i=1}^{N} H_{i\alpha}^{(n)} = |e_\alpha|.$$

Two canonical propagation operators on supervertices are:

(1) Superhypergraph diffusion operator *(incidence-based):*

$$S_{\text{diff}}^{(n)} := (D_v^{(n)})^{-1/2}\, H^{(n)}\, W^{(n)}\, (D_e^{(n)})^{-1}\, (H^{(n)})^{\top}\, (D_v^{(n)})^{-1/2}.$$

(2) Supervertex 2-section (clique-expansion) operator*: Let* $A_2^{(n)} \in \{0,1\}^{N\times N}$ *be the adjacency matrix of the 2-section graph on* V*, i.e., for* $i \neq j$*,*

$$(A_2^{(n)})_{ij} := \begin{cases} 1, & \exists\, \alpha \in \{1,\dots,M\} \text{ such that } v_i \in e_\alpha \text{ and } v_j \in e_\alpha, \\ 0, & \text{otherwise}, \end{cases} \qquad (A_2^{(n)})_{ii} := 0.$$

Define $\widetilde{A}_2^{(n)} := A_2^{(n)} + I_N$ *and* $\widetilde{D}_2^{(n)}$ *by* $\widetilde{D}_2^{(n)}(i,i) := \sum_{j=1}^{N} \widetilde{A}_2^{(n)}(i,j)$*. Then set*

$$S_{2\text{-sec}}^{(n)} := (\widetilde{D}_2^{(n)})^{-1/2}\, \widetilde{A}_2^{(n)}\, (\widetilde{D}_2^{(n)})^{-1/2}.$$

Definition 33 (Lifting base features to n-supervertices)**.** *Let* $\mathrm{SHG}^{(n)} = (V,E)$ *be on base set* $V_0 = \{x_1,\dots,x_{n_0}\}$*. Let* $X^{(0)} \in \mathbb{R}^{n_0\times d_0}$ *be base vertex features, where row* $X_{p,:}^{(0)}$ *corresponds to* x_p*.*

Define the support *of a supervertex* $v \in V$ *by*

$$B(v) := \mathrm{Flat}_n(v) \subseteq V_0.$$

Define lifted supervertex features $X^{(n,0)} \in \mathbb{R}^{|V|\times d_0}$ *by mean aggregation: for* $v_i \in V$*,*

$$X_{i,:}^{(n,0)} := \frac{1}{|B(v_i)|} \sum_{x_p \in B(v_i)} X_{p,:}^{(0)}.$$

Definition 34 (One n-SuperHyperGraph convolutional layer)**.** *Let* $\mathrm{SHG}^{(n)} = (V,E)$ *and let* $S^{(n)}$ *be a chosen propagation operator on* V *(e.g.,* $S_{\text{diff}}^{(n)}$ *or* $S_{2\text{-sec}}^{(n)}$ *from Definition 32). Let* $X^{(\ell)} \in \mathbb{R}^{|V|\times d_\ell}$ *be supervertex features at layer* ℓ*.*

An n-SuperHyperGraph convolutional layer *is the map*

$$X^{(\ell+1)} = \sigma\Big(S^{(n)}\, X^{(\ell)}\, \Theta^{(\ell)}\Big),$$

where $\Theta^{(\ell)} \in \mathbb{R}^{d_\ell \times d_{\ell+1}}$ *is learnable and* σ *is applied entrywise.*

Definition 35 (n-SuperHyperGraph Convolutional Network (n-SHGCN))**.** *Let* $\mathrm{SHG}^{(n)} = (V,E)$ *be an n-SuperHyperGraph on base set* V_0*. Given base features* $X^{(0)} \in \mathbb{R}^{|V_0|\times d_0}$*, form lifted features* $X^{(n,0)}$ *by Definition 33. Fix a propagation operator* $S^{(n)}$ *on* V*.*

An L-layer n-SHGCN *is a sequence*

$$X^{(n,0)} \mapsto X^{(n,1)} \mapsto \cdots \mapsto X^{(n,L)},$$

where each step is an n-SuperHyperGraph convolutional layer (Definition 34). Optionally, a readout map is applied for downstream tasks.

Theorem 8 (n-SHGCN generalizes HGCN and GCN)**.** *Fix* $d_0, d_1 \geq 1$*.*

(i) *(HGCN as a special case) Let* $H = (V_0, \mathcal{E}, w)$ *be a weighted hypergraph. Consider the* 0*-SuperHyperGraph* $\mathrm{SHG}^{(0)} = (V, E)$ *defined by*

$$V := V_0, \qquad E := \mathcal{E}.$$

Choose $S^{(0)} := S^{(0)}_{\mathrm{diff}}$*. Then a one-layer n-SHGCN update on* $\mathrm{SHG}^{(0)}$ *coincides exactly with a one-layer HGCN update on* H*.*

(ii) *(GCN as a special case) Let* $G = (V_0, E_G)$ *be a (simple) graph. Form the* 2*-uniform hypergraph* $H_G = (V_0, \mathcal{E}_G)$ *where* $\mathcal{E}_G := \{\{u, v\} \mid \{u, v\} \in E_G\}$*, and view it as the* 0*-SuperHyperGraph* $\mathrm{SHG}^{(0)} = (V_0, \mathcal{E}_G)$*. Choose* $S^{(0)} := S^{(0)}_{2\text{-sec}}$*. Then a one-layer n-SHGCN update on* $\mathrm{SHG}^{(0)}$ *coincides exactly with a one-layer GCN update on* G*.*

Proof. (i) Let $H = (V_0, \mathcal{E}, w)$ be given. Define $\mathrm{SHG}^{(0)} = (V, E)$ by $V = V_0$ and $E = \mathcal{E}$. Since $n = 0$, the flattening satisfies $\mathrm{Flat}_0(v) = \{v\}$ for $v \in V_0$, hence Definition 33 gives $X^{(0,0)} = X^{(0)}$ (no change of features).

The super-incidence matrix $H^{(0)}$ (Definition 32) is precisely the usual hypergraph incidence matrix, because its entries satisfy $H^{(0)}_{i\alpha} = 1$ iff $v_i \in e_\alpha$. The degree matrices $D^{(0)}_v$ and $D^{(0)}_e$ also match the standard hypergraph degrees:

$$(D^{(0)}_v)_{ii} = \sum_\alpha w_\alpha H^{(0)}_{i\alpha}, \qquad (D^{(0)}_e)_{\alpha\alpha} = \sum_i H^{(0)}_{i\alpha} = |e_\alpha|.$$

Therefore the n-SHGCN diffusion operator

$$S^{(0)}_{\mathrm{diff}} = (D^{(0)}_v)^{-1/2} H^{(0)} W^{(0)} (D^{(0)}_e)^{-1} (H^{(0)})^{\mathsf{T}} (D^{(0)}_v)^{-1/2}$$

is exactly the HGCN propagation matrix. Substituting into the layer update

$$X^{(1)} = \sigma\big(S^{(0)}_{\mathrm{diff}}\, X^{(0)}\, \Theta^{(0)}\big)$$

shows that the one-layer n-SHGCN update coincides with the one-layer HGCN update.

(ii) Let $G = (V_0, E_G)$ be a simple graph and let $\mathcal{E}_G = \{\{u, v\} \mid \{u, v\} \in E_G\}$. Consider $\mathrm{SHG}^{(0)} = (V_0, \mathcal{E}_G)$ and choose $S^{(0)} = S^{(0)}_{2\text{-sec}}$.

We show that the 2-section adjacency matrix $A^{(0)}_2$ equals the usual adjacency matrix A of G. For distinct $u, v \in V_0$,

$$(A^{(0)}_2)_{uv} = 1 \iff \exists\, e \in \mathcal{E}_G \text{ with } u \in e \text{ and } v \in e \iff \{u, v\} \in E_G \iff A_{uv} = 1.$$

Thus $A^{(0)}_2 = A$. Consequently,

$$\widetilde{A}^{(0)}_2 = A^{(0)}_2 + I = A + I, \qquad \widetilde{D}^{(0)}_2 = \widetilde{D},$$

where $\widetilde{D}$ is the degree matrix of $\widetilde{A} = A + I$. Hence

$$S^{(0)}_{2\text{-sec}} = (\widetilde{D}^{(0)}_2)^{-1/2}\, \widetilde{A}^{(0)}_2\, (\widetilde{D}^{(0)}_2)^{-1/2} = \widetilde{D}^{-1/2}\, (A + I)\, \widetilde{D}^{-1/2},$$

which is exactly the standard GCN propagation matrix. Substituting into the layer update yields

$$X^{(1)} = \sigma\big(\widetilde{D}^{-1/2} (A + I) \widetilde{D}^{-1/2}\, X^{(0)}\, \Theta^{(0)}\big),$$

so the one-layer n-SHGCN update coincides with the one-layer GCN update.

This proves both claims. □

8. SuperHyperGraph Embedding

Graph embedding maps nodes or graphs into low-dimensional vectors, preserving adjacency and structural similarity for efficient prediction and retrieval tasks [126–129]. Hypergraph embedding maps vertices and hyperedges into vectors, preserving higher-order relations among multiple vertices for robust classification and link prediction [61,130–132]. Superhypergraph embedding maps nested hyperedges into vectors, capturing hyperedge-of-hyperedge structure to support hierarchical reasoning, representation learning, and prediction tasks efficiently. For reference, Table 7 presents a comparison of Graph Embedding, HyperGraph Embedding, and SuperHyperGraph Embedding.

Table 7. Concise comparison: Graph Embedding vs. HyperGraph Embedding vs. SuperHyperGraph Embedding.

Aspect	Graph Embedding	HyperGraph Embedding	SuperHyperGraph Embedding
Underlying structure	Graph $G = (V, E)$	Hypergraph $H = (V, \mathcal{E})$	n–SuperHyperGraph $\mathrm{SHG}^{(n)} = (V, E)$ with $V \subseteq \mathcal{P}^n(V_0)$
Object to embed	Vertex $v \in V$ (and/or edge, whole-graph)	Vertex $v \in V$ and/or hyperedge $e \in \mathcal{E}$	Supervertex $s \in V$ and/or superedge $\varepsilon \in E$ (possibly multi-level)
Embedding map (typical)	$\varphi_V : V \to \mathbb{R}^d$	$\varphi_V : V \to \mathbb{R}^d$, $\varphi_E : \mathcal{E} \to \mathbb{R}^{d'}$	$\varphi_V : V \to \mathbb{R}^d$, $\varphi_E : E \to \mathbb{R}^{d'}$ (with level-aware features)
Main signal used	Pairwise adjacency / random-walk proximity / neighborhoods	Incidence (vertex–hyperedge) and multiway co-occurrence	Incidence between nested supervertices and superedges; cross-level co-membership
Typical objective (example)	Preserve neighborhood similarity: $\min \sum_{(u,v) \in E} \lVert \varphi_V(u) - \varphi_V(v) \rVert^2$	Preserve incidence: $\min \sum_{v \in V, e \in \mathcal{E}} w_{ve} \, \ell(\langle \varphi_V(v), \varphi_E(e) \rangle)$	Preserve super-incidence: $\min \sum_{s \in V, \varepsilon \in E} w_{s\varepsilon} \, \ell(\langle \varphi_V(s), \varphi_E(\varepsilon) \rangle)$
Reduction / specialization	—	If every hyperedge has size 2, reduces to a graph embedding setting	If $n = 0$, supervertices are singletons and reduces to hypergraph embedding; if moreover every hyperedge has size 2, reduces to graph embedding

Definition 36 (Graph embedding in a vector space)**.** *[128,133,134] Let $G = (V, E, w)$ be a (possibly weighted) simple graph with vertex set $V = \{v_1, \dots, v_n\}$, edge set $E \subseteq \binom{V}{2}$, and weight function $w : E \to \mathbb{R}_{>0}$. Fix an integer $d \geq 1$.*

A d-dimensional graph embedding *is a map*

$$\varphi : V \longrightarrow \mathbb{R}^d, \qquad \textit{equivalently} \qquad Z \in \mathbb{R}^{n \times d} \textit{ with row } Z_{i,:} = \varphi(v_i),$$

together with a prescribed graph proximity *(or relation) function*

$$s : V \times V \to \mathbb{R},$$

and a prescribed embedding similarity *function*

$$\mathrm{sim} : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R},$$

such that $\mathrm{sim}(\varphi(u), \varphi(v))$ *approximates* $s(u, v)$ *(or preserves its ordering) for relevant pairs* (u, v)*.*

Formally, one common way to specify this requirement is by defining φ as a minimizer of

$$\min_{\varphi : V \to \mathbb{R}^d} \sum_{(u,v) \in \Omega} L\Big(s(u,v),\ \mathrm{sim}(\varphi(u), \varphi(v))\Big) \ + \ \lambda \, \mathcal{R}(\varphi),$$

where $\Omega \subseteq V \times V$ is the set of pairs used for learning/evaluation, L is a loss function, $\mathcal{R}$ is a regularizer, and $\lambda \geq 0$.

Definition 37 (Hypergraph and incidence representation). *A (weighted) hypergraph is a triple* $\mathcal{H} = (V, \mathcal{E}, W)$ *where* $V = \{v_1, \ldots, v_n\}$ *is a vertex set,* $\mathcal{E} = \{e_1, \ldots, e_m\}$ *is a set of hyperedges with* $e \subseteq V$ *and* $|e| \geq 2$*, and* $W = \operatorname{diag}(w_{e_1}, \ldots, w_{e_m})$ *is a positive diagonal matrix of hyperedge weights.*

The vertex-hyperedge membership is represented by the incidence matrix $H \in \{0,1\}^{n \times m}$ *defined by*

$$H(v,e) = \begin{cases} 1, & v \in e, \\ 0, & v \notin e. \end{cases}$$

Define diagonal degree matrices $D_v \in \mathbb{R}^{n \times n}$ *and* $D_e \in \mathbb{R}^{m \times m}$ *by*

$$D_v(i,i) = \sum_{e \in \mathcal{E}} w_e \, H(v_i, e), \qquad D_e(j,j) = \sum_{v \in V} H(v, e_j).$$

A normalized hypergraph operator and Laplacian can be defined by

$$\Theta := D_v^{-1/2} H W D_e^{-1} H^\top D_v^{-1/2}, \qquad \Delta := I - \Theta.$$

Definition 38 (Hypergraph embedding in a vector space). *Let* $\mathcal{H} = (V, \mathcal{E}, W)$ *be a (weighted) hypergraph with* $|V| = n$ *and fix* $d \geq 1$.

A d-dimensional (vertex) hypergraph embedding *is a map*

$$\psi : V \longrightarrow \mathbb{R}^d, \qquad \textit{equivalently} \qquad Z \in \mathbb{R}^{n \times d} \textit{ with row } Z_{i,:} = \psi(v_i),$$

together with a prescribed hyperedge proximity functional

$$F_e : \ (\mathbb{R}^d)^{|e|} \to \mathbb{R} \quad (e \in \mathcal{E}),$$

such that the embedding preserves higher-order relations encoded by hyperedges.

A standard formal specification is to define ψ *as a minimizer of*

$$\min_{\psi : V \to \mathbb{R}^d} \sum_{e \in \mathcal{E}} w_e \, \mathcal{L}_e\Big(F_e\big((\psi(v))_{v \in e}\big)\Big) \; + \; \lambda \, \mathcal{R}(\psi),$$

where $\mathcal{L}_e$ *is a loss that enforces the desired within-hyperedge geometry (e.g., small dispersion, large n-wise similarity, or consistency with observed hyperedge events), and* $\mathcal{R}$ *is a regularizer.*

One concrete (spectral) instance uses the normalized hypergraph Laplacian Δ*: choose* $U \in \mathbb{R}^{n \times d}$ *whose columns are eigenvectors corresponding to the d smallest nontrivial eigenvalues of* Δ*, and set* $\psi(v_i) = U_{i,:}$.

Definition 39 (Hypergraph embedding for downstream tasks as a representation matrix). *In many learning settings, a hypergraph embedding is presented as a representation matrix*

$$Z_{\mathcal{H}} \in \mathbb{R}^{|V| \times C},$$

where each row represents the embedding of a vertex, and the representation is used for downstream tasks (e.g., node classification, graph classification) via a predictor built on $Z_{\mathcal{H}}$.

Definition 40 (Nesting map and flattening map). *Let* V_0 *be a finite, nonempty set, and let* $\mathrm{PS}^0(V_0) := V_0$ *and* $\mathrm{PS}^{k+1}(V_0) := \mathrm{PS}(\mathrm{PS}^k(V_0))$ *for* $k \geq 0$.

Define the n-fold nesting map $\mathrm{Nest}_n : V_0 \to \mathrm{PS}^n(V_0)$ *by*

$$\mathrm{Nest}_0(x) := x, \qquad \mathrm{Nest}_{n+1}(x) := \{\mathrm{Nest}_n(x)\} \quad (x \in V_0, \, n \geq 0).$$

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 24 December 2025 doi:10.20944/preprints202512.2233.v1

Define the n-level flattening map $\mathrm{Flat}_n : \mathrm{PS}^n(V_0) \to \mathrm{PS}(V_0)$ *recursively as follows:*

$$\mathrm{Flat}_0(x) := \{x\} \quad (x \in V_0), \qquad \mathrm{Flat}_{n+1}(X) := \bigcup_{Y \in X} \mathrm{Flat}_n(Y) \quad (X \in \mathrm{PS}^{n+1}(V_0),\, n \geq 0).$$

Lemma 1 (Flattening a nested singleton)**.** *For every* $n \geq 0$ *and every* $x \in V_0$*,*

$$\mathrm{Flat}_n\big(\mathrm{Nest}_n(x)\big) = \{x\}.$$

Proof. We prove by induction on n.

If $n = 0$, then $\mathrm{Nest}_0(x) = x$ and $\mathrm{Flat}_0(x) = \{x\}$ by definition, hence $\mathrm{Flat}_0(\mathrm{Nest}_0(x)) = \{x\}$.

Assume $\mathrm{Flat}_n(\mathrm{Nest}_n(x)) = \{x\}$ holds for some $n \geq 0$. Then

$$\mathrm{Nest}_{n+1}(x) = \{\mathrm{Nest}_n(x)\}.$$

Applying the recursive definition of Flat_{n+1} gives

$$\mathrm{Flat}_{n+1}\big(\mathrm{Nest}_{n+1}(x)\big) = \mathrm{Flat}_{n+1}\big(\{\mathrm{Nest}_n(x)\}\big) = \bigcup_{Y \in \{\mathrm{Nest}_n(x)\}} \mathrm{Flat}_n(Y) = \mathrm{Flat}_n(\mathrm{Nest}_n(x)) = \{x\}.$$

Thus the statement holds for $n + 1$. This completes the induction. □

Definition 41 (n-lift of a hypergraph)**.** *Let* $H = (V_0, \mathcal{E})$ *be a hypergraph, i.e.,* $\mathcal{E} \subseteq \mathrm{PS}(V_0) \setminus \{\emptyset\}$*. Fix* $n \geq 0$.

Define the n-lifted vertex set and hyperedge family by

$$V_n := \{\mathrm{Nest}_n(v) \mid v \in V_0\} \subseteq \mathrm{PS}^n(V_0), \qquad \mathcal{E}_n := \{\mathrm{Nest}_n(e) \mid e \in \mathcal{E}\},$$

where for each hyperedge $e \subseteq V_0$ *we set*

$$\mathrm{Nest}_n(e) := \{\mathrm{Nest}_n(v) \mid v \in e\} \subseteq V_n.$$

Then

$$\mathrm{Lift}_n(H) := (V_n, \mathcal{E}_n)$$

is an n-SuperHyperGraph in the sense that $V_n \subseteq \mathrm{PS}^n(V_0)$ *and* $\mathcal{E}_n \subseteq \mathrm{PS}(V_n) \setminus \{\emptyset\}$.

Definition 42 (n-SuperHypergraph embedding)**.** *Let* $\mathrm{SHG}^{(n)} = (V, E)$ *be an n-SuperHyperGraph on a base set* V_0*, so* $V \subseteq \mathrm{PS}^n(V_0)$ *and* $E \subseteq \mathrm{PS}(V) \setminus \{\emptyset\}$*. Fix an integer* $d \geq 1$.

An n-SuperHypergraph embedding *in* $\mathbb{R}^d$ *is a pair of maps*

$$\varphi : V_0 \to \mathbb{R}^d, \qquad \psi : V \to \mathbb{R}^d,$$

together with (i) an aggregation rule Agg *for finite multisets of vectors and (ii) a family of hyperedge-coherence losses* $(\mathcal{L}_e)_{e \in E}$*, where each* $\mathcal{L}_e$ *maps the tuple* $(\psi(v))_{v \in e}$ *to a nonnegative real number.*

A concrete (canonical) choice that is widely used is:

$$\mathrm{Agg}(\{y_1, \ldots, y_k\}) := \frac{1}{k} \sum_{i=1}^{k} y_i,$$

and for each $e \in E$*,*

$$\mathcal{L}_e\big((\psi(v))_{v \in e}\big) := \frac{1}{|e|} \sum_{v \in e} \|\psi(v) - \mu_e\|^2, \qquad \mu_e := \frac{1}{|e|} \sum_{u \in e} \psi(u).$$

In addition, to reflect the SuperHyperGraph *nesting, we enforce a hierarchical consistency term via* Flat_n*: for each* $v \in V$*, define its base support* $B(v) := \mathrm{Flat}_n(v) \subseteq V_0$ *and set*

$$\mathrm{Anchor}_\varphi(v) := \mathrm{Agg}\big(\{\varphi(x) \mid x \in B(v)\}\big) \in \mathbb{R}^d.$$

One standard way to specify that (φ, ψ) *is an embedding is to require it to minimize an objective of the form*

$$J(\varphi,\psi) := \sum_{e\in E} w_e\, \mathcal{L}_e\big((\psi(v))_{v\in e}\big) \;+\; \beta \sum_{v\in V} \big\|\psi(v) - \mathrm{Anchor}_\varphi(v)\big\|^2 \;+\; \lambda\, \mathcal{R}(\varphi,\psi),$$

where $w_e > 0$ *are hyperedge weights,* $\beta, \lambda \geq 0$*, and* $\mathcal{R}$ *is a regularizer.*

Theorem 9 (SuperHypergraph embeddings generalize hypergraph and graph embeddings)**.** *Fix* $d \geq 1$ *and* $n \geq 0$*.*

(i) *(Hypergraph case) Let* $H = (V_0, \mathcal{E})$ *be a hypergraph and let* $\Phi : V_0 \to \mathbb{R}^d$ *be any vertex embedding of* H *(e.g., specified by minimizing a hyperedge-coherence objective over* $\mathcal{E}$*). Then* $\mathrm{Lift}_n(H)$ *admits an n-SuperHypergraph embedding* (φ, ψ) *such that*

$$\psi(\mathrm{Nest}_n(v)) = \Phi(v) \quad \textit{for all } v \in V_0,$$

and for the canonical loss $\mathcal{L}_e$ *in Definition 42, the hyperedge terms on* H *and on* $\mathrm{Lift}_n(H)$ *coincide under this identification.*

(ii) *(Graph case) Let* $G = (V_0, E_G)$ *be a (simple) graph and regard it as the* 2*-uniform hypergraph* $H_G = (V_0, \mathcal{E}_G)$ *with* $\mathcal{E}_G := \{\{u,v\} \mid \{u,v\} \in E_G\}$*. Then every graph embedding* $\Phi : V_0 \to \mathbb{R}^d$ *induces an n-SuperHypergraph embedding of* $\mathrm{Lift}_n(H_G)$ *via the same rule* $\psi(\mathrm{Nest}_n(v)) = \Phi(v)$*. Thus graph embeddings are contained as a special case of n-SuperHypergraph embeddings.*

Proof. (i) Consider the lifted n-SuperHyperGraph $\mathrm{Lift}_n(H) = (V_n, \mathcal{E}_n)$ from Definition 41. Define

$$\varphi := \Phi \text{ on } V_0, \qquad \psi : V_n \to \mathbb{R}^d \text{ by } \psi(\mathrm{Nest}_n(v)) := \Phi(v).$$

This is well-defined because each element of V_n is uniquely of the form $\mathrm{Nest}_n(v)$.

Let $e \in \mathcal{E}$ be a hyperedge, and let $e_n := \mathrm{Nest}_n(e) \in \mathcal{E}_n$ be its lifted hyperedge. Using the canonical coherence loss from Definition 42, we compute explicitly:

$$\mu_{e_n} = \frac{1}{|e_n|} \sum_{u' \in e_n} \psi(u') = \frac{1}{|e|} \sum_{u\in e} \psi(\mathrm{Nest}_n(u)) = \frac{1}{|e|} \sum_{u\in e} \Phi(u) = \mu_e,$$

where $\mu_e := \frac{1}{|e|} \sum_{u\in e} \Phi(u)$ is the corresponding mean for the hyperedge e under Φ. Therefore,

$$\mathcal{L}_{e_n}\big((\psi(v'))_{v'\in e_n}\big) = \frac{1}{|e|} \sum_{v\in e} \|\psi(\mathrm{Nest}_n(v)) - \mu_{e_n}\|^2 = \frac{1}{|e|} \sum_{v\in e} \|\Phi(v) - \mu_e\|^2 = \mathcal{L}_e\big((\Phi(v))_{v\in e}\big).$$

Hence the sum of hyperedge-coherence terms is preserved by the lift.

Moreover, the hierarchical consistency term is compatible with the lift. For any $v \in V_0$, Lemma 1 yields

$$\mathrm{Flat}_n(\mathrm{Nest}_n(v)) = \{v\}.$$

With mean aggregation,

$$\mathrm{Anchor}_\varphi(\mathrm{Nest}_n(v)) = \mathrm{Agg}\big(\{\varphi(x) \mid x \in \mathrm{Flat}_n(\mathrm{Nest}_n(v))\}\big) = \mathrm{Agg}\big(\{\varphi(v)\}\big) = \varphi(v) = \Phi(v).$$

Therefore,

$$\big\|\psi(\mathrm{Nest}_n(v)) - \mathrm{Anchor}_\varphi(\mathrm{Nest}_n(v))\big\|^2 = \|\Phi(v) - \Phi(v)\|^2 = 0,$$

so the hierarchical term vanishes on lifted vertices. This shows that (φ, ψ) is an n-SuperHypergraph embedding of $\mathrm{Lift}_n(H)$ and that the lifted model recovers the original hypergraph embedding behavior.

(ii) A graph $G = (V_0, E_G)$ is a special case of a hypergraph by taking only 2-element hyperedges. Applying part (i) to $H_G = (V_0, \mathcal{E}_G)$ produces an n-SuperHypergraph embedding of $\mathrm{Lift}_n(H_G)$ with $\psi(\mathrm{Nest}_n(v)) = \Phi(v)$. Thus graph embeddings are included as a special case. □

9. Result and Reviews: SuperHyperGraph-Based Natural Language Processing

Natural language processing (NLP) studies learning algorithms that map text to structured representations, meanings, or decisions using corpora, representations, and task-specific objective functions [135–139]. Related notions such as Fuzzy Natural Language [140–142] and Natural HyperLanguage [143,144] are also known in the literature. Graph-based NLP encodes text as a graph of tokens, entities, or relations and learns predictors on vertices and edges via message passing and attention mechanisms [145–149]. Hypergraph-based NLP encodes text as a hypergraph whose hyperedges connect multiple items simultaneously, learning representations that capture higher-order, multiway linguistic interactions. Superhypergraph-based NLP encodes text as an n-level superhypergraph through iterated powerset constructions, enabling nested linguistic groupings and learning predictors that generalize both graph and hypergraph pipelines. The comparison of Graph-Based NLP, HyperGraph-Based NLP, and SuperHyperGraph-Based NLP is presented in Table 8.

Table 8. Concise comparison: Graph-Based NLP vs. HyperGraph-Based NLP vs. SuperHyperGraph-Based NLP.

Category	Graph-Based NLP	HyperGraph-Based NLP	SuperHyperGraph-Based NLP
Encoder	$\Gamma_G : \mathcal{X} \to \mathcal{G}$	$\Gamma_H : \mathcal{X} \to \mathcal{H}$	$\Gamma_{SH}^{(n)} : \mathcal{X} \to \mathcal{S}^{(n)}$
Structure	$G = (V, E), E \subseteq \binom{V}{2}$	$H = (V, \mathcal{E}), \mathcal{E} \subseteq \mathcal{P}(V) \setminus \{\emptyset\}$	$\mathrm{SHG}^{(n)} = (V_0, V, E), V \subseteq \mathcal{P}^n(V_0),$ $E \subseteq \mathcal{P}(V) \setminus \{\emptyset\}$
Relations	Pairwise	Multiway (hyperedges)	Nested multiway (multi-level)
Typical units	Tokens, entities, dependency edges	Phrases, co-reference sets, events	Nested discourse units (token $\subset$ phrase $\subset$ sentence $\subset$ doc)
Learner	$f_\theta : \mathcal{G} \to \mathcal{Y}$	$f_\theta : \mathcal{H} \to \mathcal{Y}$	$g_\theta : \mathcal{S}^{(n)} \to \mathcal{Y}$
Benefit	Local syntax/semantics	Higher-order semantics	Hierarchical, multi-resolution semantics

Definition 43 (Graph-Based Natural Language Processing (Graph-NLP))**.** *Fix a finite alphabet Σ and let $\mathcal{X} \subseteq \Sigma^*$ be a set of text objects (e.g., tokens, sentences, documents, dialogue threads), and let $\mathcal{Y}$ be an output space (e.g., labels, rankings, structured outputs).*

A graph construction (encoder) *is a mapping*

$$\Gamma : \mathcal{X} \longrightarrow \mathcal{G},$$

where $\mathcal{G}$ is a class of attributed graphs, and for each $x \in \mathcal{X}$,

$$\Gamma(x) = G_x = (V_x, E_x, \ell_V, \ell_E, w)$$

with

- *V_x a finite vertex set (representing linguistic units such as tokens, entities, sentences, posts),*
- *$E_x \subseteq V_x \times V_x$ an edge set (representing relations such as adjacency, dependency, coreference, reply-to, semantic links),*
- *$\ell_V : V_x \to \mathcal{A}_V$ and $\ell_E : E_x \to \mathcal{A}_E$ vertex/edge attributes (discrete or continuous),*
- *w optional weights on vertices/edges.*

A graph-based NLP model family *is a set of measurable maps*

$$\mathcal{F} = \{\, f_\theta : \mathcal{G} \to \mathcal{Y} \mid \theta \in \Theta \,\},$$

where each f_θ *is allowed to use the graph structure (e.g., paths, neighborhoods, walks, cuts, centrality, or message passing) in computing the output.*

Given a distribution $\mathbb{P}$ *over pairs* $(x, y) \in \mathcal{X} \times \mathcal{Y}$ *and a loss* $\mathcal{L} : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_{\geq 0}$, Graph-NLP *is the learning/optimization problem*

$$\min_{\theta \in \Theta} \mathbb{E}_{(x,y)\sim\mathbb{P}}\Big[\mathcal{L}\big(f_\theta(\Gamma(x)),\, y\big)\Big].$$

Any pipeline $x \mapsto \Gamma(x) \mapsto f_\theta(\Gamma(x))$ *that is designed and evaluated for NLP tasks (e.g., classification, extraction, retrieval, summarization, entailment) is called a* graph-based NLP method.

Definition 44 (HyperGraph-Based Natural Language Processing (HyperGraph-NLP))**.** *Fix a finite alphabet* Σ*, a set of text objects* $\mathcal{X} \subseteq \Sigma^*$*, and an output space* $\mathcal{Y}$*.*

A hypergraph construction (encoder) *is a mapping*

$$\Gamma_H : \mathcal{X} \to \mathcal{H},$$

where $\mathcal{H}$ *is a class of finite attributed hypergraphs. For each* $x \in \mathcal{X}$*,*

$$\Gamma_H(x) = H_x = (V_x, \mathcal{E}_x, \ell_V, \ell_{\mathcal{E}}, w),$$

where V_x *is a finite vertex set,* $\mathcal{E}_x \subseteq \mathcal{P}(V_x) \setminus \{\emptyset\}$ *is a family of hyperedges,* $\ell_V : V_x \to \mathcal{A}_V$ *and* $\ell_{\mathcal{E}} : \mathcal{E}_x \to \mathcal{A}_E$ *are attribute maps, and* w *denotes optional weights (on vertices and/or hyperedges).*

A hypergraph-based NLP model family *is a set of (measurable) maps*

$$\mathcal{F}_H = \{\, f_\theta : \mathcal{H} \to \mathcal{Y} \mid \theta \in \Theta \,\}.$$

Given a distribution $\mathbb{P}$ *on* $\mathcal{X} \times \mathcal{Y}$ *and a loss* $\mathcal{L} : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_{\geq 0}$, HyperGraph-NLP *is the learning problem*

$$\min_{\theta \in \Theta} \mathbb{E}_{(x,y)\sim\mathbb{P}}\Big[\mathcal{L}\big(f_\theta(\Gamma_H(x)), y\big)\Big].$$

Definition 45 (SuperHyperGraph-Based Natural Language Processing (SuperHyperGraph-NLP))**.** *Fix* $\Sigma, \mathcal{X}, \mathcal{Y}$ *as above and fix an integer* $n \geq 0$*.*

An n–SuperHyperGraph construction (encoder) *is a mapping*

$$\Gamma_{SH}^{(n)} : \mathcal{X} \to \mathcal{S}^{(n)},$$

where $\mathcal{S}^{(n)}$ *is a class of finite attributed* n*–SuperHyperGraphs. For each* $x \in \mathcal{X}$*,*

$$\Gamma_{SH}^{(n)}(x) = \mathrm{SHG}_x^{(n)} = (V_{0,x}, V_x, E_x, \ell_V, \ell_E, w),$$

such that

$$V_{0,x} \neq \emptyset \text{ is finite}, \qquad V_x \subseteq \mathcal{P}^n(V_{0,x}), \qquad E_x \subseteq \mathcal{P}(V_x) \setminus \{\emptyset\}.$$

Elements of V_x *are* n–supervertices *and elements of* E_x *are* n–superedges*. The maps* $\ell_V : V_x \to \mathcal{A}_V$ *and* $\ell_E : E_x \to \mathcal{A}_E$ *assign attributes, and* w *denotes optional weights.*

A superhypergraph-based NLP model family *is*

$$\mathcal{F}_{SH}^{(n)} = \{\, g_\theta : \mathcal{S}^{(n)} \to \mathcal{Y} \mid \theta \in \Theta \,\}.$$

Given $\mathbb{P}$ *and* $\mathcal{L}$ *as above,* SuperHyperGraph-NLP *is*

$$\min_{\theta \in \Theta} \mathbb{E}_{(x,y)\sim\mathbb{P}}\Big[\mathcal{L}\big(g_\theta(\Gamma_{SH}^{(n)}(x)), y\big)\Big].$$

Definition 46 (Graph-to-hypergraph inclusion)**.** *Let* $G = (V, E)$ *be a (simple, undirected) graph with* $E \subseteq \{\{u, v\} \subseteq V \mid u \neq v\}$*. Define the associated* 2-uniform hypergraph

$$\iota(G) = (V, \mathcal{E}) \quad by \quad \mathcal{E} = \{\{u, v\} \mid \{u, v\} \in E\} \subseteq \mathcal{P}(V) \setminus \{\emptyset\}.$$

Theorem 10 (HyperGraph-NLP generalizes Graph-NLP)**.** *Every instance of Graph-Based Natural Language Processing is a special case of HyperGraph-Based Natural Language Processing.*

Proof. Consider any Graph-NLP pipeline consisting of a graph encoder $\Gamma_G : \mathcal{X} \to \mathcal{G}$ and a model family $\mathcal{F}_G = \{f_\theta : \mathcal{G} \to \mathcal{Y}\}$, where $\mathcal{G}$ is a class of finite attributed graphs.

Define a hypergraph encoder by composition:

$$\Gamma_H = \iota \circ \Gamma_G : \mathcal{X} \to \mathcal{H},$$

where $\mathcal{H}$ contains (at least) all 2-uniform attributed hypergraphs obtained from graphs in $\mathcal{G}$. Define a hypergraph model family on this subclass by

$$f_\theta^H(\iota(G)) = f_\theta(G) \quad \text{for every } G \in \mathcal{G}.$$

Then for every (x, y),

$$\mathcal{L}\big(f_\theta^H(\Gamma_H(x)), y\big) = \mathcal{L}\big(f_\theta^H(\iota(\Gamma_G(x))), y\big) = \mathcal{L}\big(f_\theta(\Gamma_G(x)), y\big).$$

Taking expectation over $(x, y) \sim \mathbb{P}$ shows that the optimization objective of the constructed HyperGraph-NLP instance is identical to the original Graph-NLP objective. Hence Graph-NLP is a special case of HyperGraph-NLP. □

Definition 47 (Hypergraph-to-superhypergraph inclusion at level $n = 0$)**.** *Let* $H = (V, \mathcal{E})$ *be a hypergraph with* $\mathcal{E} \subseteq \mathcal{P}(V) \setminus \{\emptyset\}$*. Define*

$$\jmath_0(H) = \mathrm{SHG}^{(0)} = (V_0, V, E) \quad by \quad V_0 = V,\ V = V_0 \subseteq \mathcal{P}^0(V_0),\ E = \mathcal{E}.$$

Definition 48 (Singleton lifting of graphs/hypergraphs to level n)**.** *For* $n \geq 0$*, define* $\sigma_n : V_0 \to \mathcal{P}^n(V_0)$ *recursively by*

$$\sigma_0(v) = v, \qquad \sigma_{k+1}(v) = \{\sigma_k(v)\} \ \ (k \geq 0).$$

Given a hypergraph $H = (V_0, \mathcal{E})$*, define its* n-lift

$$\jmath_n(H) = \mathrm{SHG}^{(n)} = (V_0, V^{(n)}, E^{(n)}),$$

where

$$V^{(n)} = \{\sigma_n(v) \mid v \in V_0\} \subseteq \mathcal{P}^n(V_0), \qquad E^{(n)} = \Big\{ \{\sigma_n(v) \mid v \in e\} \ \Big| \ e \in \mathcal{E} \Big\}.$$

(Attributes and weights are transported along σ_n *in the obvious way.)*

Theorem 11 (SuperHyperGraph-NLP generalizes Graph-NLP and HyperGraph-NLP)**.** *For every* $n \geq 0$*, every instance of HyperGraph-NLP is a special case of SuperHyperGraph-NLP at level* n*. Consequently, every instance of Graph-NLP is also a special case of SuperHyperGraph-NLP.*

Proof. Fix $n \geq 0$.

Step 1 (HyperGraph-NLP ⇒ SuperHyperGraph-NLP). Take any HyperGraph-NLP pipeline with encoder $\Gamma_H : \mathcal{X} \to \mathcal{H}$ and model family $\mathcal{F}_H = \{f_\theta : \mathcal{H} \to \mathcal{Y}\}$. Define a superhypergraph encoder by

$$\Gamma_{SH}^{(n)} = \jmath_n \circ \Gamma_H : \mathcal{X} \to \mathcal{S}^{(n)}.$$

Define a superhypergraph model family on the image of $\jmath_n$ by

$$g_\theta(\jmath_n(H)) = f_\theta(H) \quad \text{for every } H \in \mathcal{H}.$$

Then for every (x, y),

$$\mathcal{L}\big(g_\theta(\Gamma^{(n)}_{SH}(x)), y\big) = \mathcal{L}\big(g_\theta(\jmath_n(\Gamma_H(x))), y\big) = \mathcal{L}\big(f_\theta(\Gamma_H(x)), y\big).$$

Thus the expected risk objective is preserved, and HyperGraph-NLP is a special case of SuperHyperGraph-NLP.

Step 2 (Graph-NLP $\Rightarrow$ SuperHyperGraph-NLP). By Theorem above, Graph-NLP is a special case of HyperGraph-NLP via the inclusion ι into 2-uniform hypergraphs. Composing with Step 1 yields a SuperHyperGraph-NLP instance (at level n) with identical objective value for all θ. Therefore SuperHyperGraph-NLP generalizes both Graph-NLP and HyperGraph-NLP. $\square$

10. Conclusion

This paper extended several core frameworks—including GCN, GraphRAG, causal graphs, graph embedding, and graph generation—by integrating the SuperHyperGraph perspective. Future work will include computational experiments and quantitative evaluation on real-world datasets. We also hope that further research will explore extensions of the concepts presented here using Fuzzy Sets [150], Rough Sets [151,152], Neutrosophic Sets [79,153,154], Uncertain Sets[78,155], and Plithogenic Sets [156,157].

Funding: This study was conducted without any financial support from external organizations or grants.

Acknowledgments: We would like to express our sincere gratitude to everyone who provided valuable insights, support, and encouragement throughout this research. We also extend our thanks to the readers for their interest and to the authors of the referenced works, whose scholarly contributions have greatly influenced this study. Lastly, we are deeply grateful to the publishers and reviewers who facilitated the dissemination of this work.

Data Availability Statement: Since this research is purely theoretical and mathematical, no empirical data or computational analysis was utilized. Researchers are encouraged to expand upon these findings with data-oriented or experimental approaches in future studies.

Ethical Statement: As this study does not involve experiments with human participants or animals, no ethical approval was required.

Conflicts of Interest: The authors declare that they have no conflicts of interest related to the content or publication of this paper.

Code Availability: No code or software was developed for this study.

Clinical Trial: This study did not involve any clinical trials.

Consent to Participate: Not applicable.

Use of Generative AI and AI-Assisted Tools: I use generative AI and AI-assisted tools for tasks such as English grammar checking, and I do not employ them in any way that violates ethical standards.

Disclaimer: This work presents theoretical ideas and frameworks that have not yet been empirically validated. Readers are encouraged to explore practical applications and further refine these concepts. Although care has been taken to ensure accuracy and appropriate citations, any errors or oversights are unintentional. The perspectives and interpretations expressed herein are solely those of the authors and do not necessarily reflect the viewpoints of their affiliated institutions.

References

1. Diestel, R. *Graph theory*; Springer (print edition); Reinhard Diestel (eBooks), 2024.
2. Gross, J.L.; Yellen, J.; Anderson, M. *Graph theory and its applications*; Chapman and Hall/CRC, 2018.

3. Wagner, S.; Wang, H. *Introduction to chemical graph theory*; Chapman and Hall/CRC, 2018.
4. Zhang, Y. Graph Neural Network-Based User Preference Model for Social Network Access Control. *Informatica* **2025**, *49*.
5. Feng, Y.; You, H.; Zhang, Z.; Ji, R.; Gao, Y. Hypergraph neural networks. In Proceedings of the Proceedings of the AAAI conference on artificial intelligence, 2019, Vol. 33, pp. 3558–3565.
6. Gao, Y.; Zhang, Z.; Lin, H.; Zhao, X.; Du, S.; Zou, C. Hypergraph learning: Methods and practices. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2020**, *44*, 2548–2566.
7. Feng, Y.; Han, J.; Ying, S.; Gao, Y. Hypergraph isomorphism computation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2024**.
8. Banerjee, A.; Mishra, R.; Parui, S. On the Spectra of Threshold Hypergraphs. *arXiv preprint arXiv:2207.02528* **2022**.
9. Chiarelli, N.; Milanič, M. Total domishold graphs: a generalization of threshold graphs, with connections to threshold hypergraphs. *Discrete Applied Mathematics* **2014**, *179*, 1–12.
10. Keevash, P. Hypergraph turan problems. *Surveys in combinatorics* **2011**, *392*, 83–140.
11. Ramos, E.L.H.; Ayala, L.R.A.; Macas, K.A.S. Study of Factors that Influence a Victim's Refusal to Testify for Sexual Reasons Due to External Influence Using Plithogenic n-SuperHyperGraphs. *Operational Research Journal* **2025**, *46*, 328–337.
12. Ghods, M.; Rostami, Z.; Smarandache, F. Introduction to Neutrosophic Restricted SuperHyperGraphs and Neutrosophic Restricted SuperHyperTrees and several of their properties. *Neutrosophic Sets and Systems* **2022**, *50*, 480–487.
13. Fujita, T.; Smarandache, F. Neutrosophic Soft n-Super-HyperGraphs with Real-World Applications. *European Journal of Pure and Applied Mathematics* **2025**, *18*, 6621. https://doi.org/10.29020/nybg.ejpam.v18i3.6621.
14. Alqahtani, M. Intuitionistic Fuzzy Quasi-Supergraph Integration for Social Network Decision Making. *International Journal of Analysis and Applications* **2025**, *23*, 137–137.
15. Smarandache, F. *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.
16. Hamidi, M.; Taghinezhad, M. *Application of Superhypergraphs-Based Domination Number in Real World*; Infinite Study, 2023.
17. Hamidi, M.; Smarandache, F.; Davneshvar, E. Spectrum of superhypergraphs via flows. *Journal of Mathematics* **2022**, *2022*, 9158912.
18. Bravo, J.C.M.; Piedrahita, C.J.B.; Bravo, M.A.M.; Pilacuan-Bonete, L.M. Integrating SMED and Industry 4.0 to optimize processes with plithogenic n-SuperHyperGraphs. *Neutrosophic Sets and Systems* **2025**, *84*, 328–340.
19. Tarjan, R.E. Depth-First Search and Linear Graph Algorithms. *SIAM J. Comput.* **1972**, *1*, 146–160.
20. Mehlhorn, K. Graph Algorithm and NP-Completeness. 1984.
21. Holland, P.W.; Leinhardt, S. An exponential family of probability distributions for directed graphs. *Journal of the american Statistical association* **1981**, *76*, 33–50.
22. Fujita, T. Extensions of multidirected graphs: Fuzzy, neutrosophic, plithogenic, rough, soft, hypergraph, and superhypergraph variants. *International journal of topology* **2025**, *2*, 11.
23. George, B.; Jose, J.; Thumbakara, R.K.; et al. Introducing soft directed hypergraphs: a fusion of soft set theory and directed hypergraphs. *Utilitas Mathematica* **2024**, *121*.
24. George, B.; Jose, J.; Thumbakara, R.K. Soft Directed Hypergraphs and Their AND & OR Operations. *Mathematical Forum* **2022**, *30*.
25. Fujita, T.; Smarandache, F. A Concise Study of Some Superhypergraph Classes. *Neutrosophic Sets and Systems* **2024**, *77*, 548–593.
26. Wei, E.L.; Tang, W.L.; Ye, D. Nowhere-zero 15-flow in 3-edge-connected bidirected graphs. *Acta Mathematica Sinica, English Series* **2014**, *30*, 649–660.
27. Xu, R.; Zhang, C.Q. On flows in bidirected graphs. *Discrete mathematics* **2005**, *299*, 335–343.
28. Wei, E.; Tang, W.; Wang, X. Flows in 3-edge-connected bidirected graphs. *Frontiers of Mathematics in China* **2011**, *6*, 339–348.
29. Berge, C. *Hypergraphs: combinatorics of finite sets*; Vol. 45, Elsevier, 1984.
30. Khan, B.; Wu, J.; Yang, J.; Ma, X. Heterogeneous hypergraph neural network for social recommendation using attention network. *ACM Transactions on Recommender Systems* **2025**, *3*, 1–22.
31. Hu, B.; Huang, W.; Zheng, T.; Song, M.; Li, Y. A general heterogeneous hypergraph neural network for node classification. In Proceedings of the 2023 International Joint Conference on Neural Networks (IJCNN). IEEE, 2023, pp. 1–7.

32. Jin, H.; Zhou, K.; Yin, J.; You, L.; Zhou, Z. Multi-Granular Attention based Heterogeneous Hypergraph Neural Network. *arXiv preprint arXiv:2505.04340* **2025**.
33. Fujita, T.; Smarandache, F. Superhypergraph Neural Networks and Plithogenic Graph Neural Networks: Theoretical Foundations. *Advancing Uncertain Combinatorics through Graphization, Hyperization, and Uncertainization: Fuzzy, Neutrosophic, Soft, Rough, and Beyond* **2025**, *5*, 577.
34. Yi, O.; Guo, B.; Tang, X.; He, X.; Xiong, J.; Yu, Z. Learning Cross-Domain Representation with Multi-Graph Neural Network. *ArXiv* **2019**, *abs/1905.10095*.
35. Lee, L.H.; Lu, Y. Multiple embeddings enhanced multi-graph neural networks for Chinese healthcare named entity recognition. *IEEE Journal of Biomedical and Health Informatics* **2021**, *25*, 2801–2810.
36. Song, Y.; Ye, H.; Li, M.; Cao, F. Deep multi-graph neural networks with attention fusion for recommendation. *Expert Systems with Applications* **2022**, *191*, 116240.
37. Wientjes, M. Hyperlink prediction with Modular Directed Hypergraph Neural Networks. PhD thesis, Master's thesis, Eindhoven University of Technology, 2021.
38. Yang, G.; Jin, T.; Dou, L. Heterogeneous directed hypergraph neural network over abstract syntax tree (ast) for code classification. *arXiv preprint arXiv:2305.04228* **2023**.
39. Tran, L.H.; Tran, L.H. Directed hypergraph neural network. *ArXiv* **2020**, *abs/2008.03626*.
40. Fujita, T. Multi-SuperHyperGraph Neural Networks: A Generalization of Multi-HyperGraph Neural Networks. *Neutrosophic Computing and Machine Learning* **2025**, *39*, 328–347.
41. Zass, R.; Shashua, A. Probabilistic graph and hypergraph matching. In Proceedings of the 2008 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2008, pp. 1–8.
42. Nenadov, R. Probabilistic hypergraph containers. *Israel Journal of Mathematics* **2024**, *261*, 879–897.
43. Fujita, T. Review of Probabilistic HyperGraph and Probabilistic SuperHyperGraph. *Spectrum of Operational Research* **2026**, *3*, 319–338. https://doi.org/https://doi.org/10.31181/sor31202651.
44. Acid, S.; de Campos, L.M.; Castellano, F.J.G. Learning Bayesian Network Classifiers: Searching in a Space of Partially Directed Acyclic Graphs. *Machine Learning* **2005**, *59*, 213–235.
45. Tariq, R.; Aadil, F.; Malik, M.F.; Ejaz, S.; Khan, M.U.; Khan, M.F. Directed Acyclic Graph Based Task Scheduling Algorithm for Heterogeneous Systems. In Proceedings of the Intelligent Systems with Applications, 2018.
46. Fujita, T. Directed Acyclic SuperHypergraphs (DASH): A General Framework for Hierarchical Dependency Modeling. *Neutrosophic Knowledge* **2025**, *6*, 72–86.
47. Venhorst, J.; Núnez, S.; Kruse, C.G. Design of a high fragment efficiency library by molecular graph theory. *ACS Medicinal Chemistry Letters* **2010**, *1*, 499–503.
48. Alhamoud, K.; Ghunaim, Y.; Alshehri, A.S.; Li, G.; Ghanem, B.; You, F. Leveraging 2D molecular graph pretraining for improved 3D conformer generation with graph neural networks. *Computers & Chemical Engineering* **2024**, *183*, 108622.
49. Rahman, A.; Poirel, C.L.; Badger, D.J.; Murali, T. Reverse engineering molecular hypergraphs. In Proceedings of the Proceedings of the ACM Conference on Bioinformatics, Computational Biology and Biomedicine, 2012, pp. 68–75.
50. Chen, J.; Schwaller, P. Molecular hypergraph neural networks. *The Journal of Chemical Physics* **2024**, *160*.
51. Fujita, T. Molecular Fuzzy graphs, hypergraphs, and superhypergraphs. *Journal of Intelligent Decision and Computational Modelling* **2025**, *1*, 158–171.
52. Fujita, T. An Introduction and Reexamination of Molecular Hypergraph and Molecular N-SuperHypergraph. *Asian Journal of Physical and Chemical Sciences* **2025**, *13*, 1–38.
53. Schmidt, J.; Pettersson, L.; Verdozzi, C.; Botti, S.; Marques, M.A. Crystal graph attention networks for the prediction of stable materials. *Science advances* **2021**, *7*, eabi7948.
54. Wang, X.; He, X.; Cao, Y.; Liu, M.; Chua, T.S. Kgat: Knowledge graph attention network for recommendation. In Proceedings of the Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining, 2019, pp. 950–958.
55. Ding, K.; Wang, J.; Li, J.; Li, D.; Liu, H. Be more with less: Hypergraph attention networks for inductive text classification. *arXiv preprint arXiv:2011.00387* **2020**.
56. Wang, J.; Ding, K.; Zhu, Z.; Caverlee, J. Session-based recommendation with hypergraph attention networks. In Proceedings of the Proceedings of the 2021 SIAM international conference on data mining (SDM). SIAM, 2021, pp. 82–90.
57. Fujita, T.; Mehmood, A. SuperHyperGraph Attention Networks. *Neutrosophic Computing and Machine Learning* **2025**, *40*, 10–27.

58. Han, H.; Wang, Y.; Shomer, H.; Guo, K.; Ding, J.; Lei, Y.; Halappanavar, M.; Rossi, R.A.; Mukherjee, S.; Tang, X.; et al. Retrieval-augmented generation with graphs (graphrag). *arXiv preprint arXiv:2501.00309* **2024**.
59. Han, H.; Ma, L.; Shomer, H.; Wang, Y.; Lei, Y.; Guo, K.; Hua, Z.; Long, B.; Liu, H.; Aggarwal, C.C.; et al. Rag vs. graphrag: A systematic evaluation and key insights. *arXiv preprint arXiv:2502.11371* **2025**.
60. Luo, H.; Chen, G.; Zheng, Y.; Wu, X.; Guo, Y.; Lin, Q.; Feng, Y.; Kuang, Z.; Song, M.; Zhu, Y.; et al. HyperGraphRAG: Retrieval-Augmented Generation via Hypergraph-Structured Knowledge Representation. *arXiv preprint arXiv:2503.21322* **2025**.
61. Wang, X.; Huang, W.; Zhao, B.; Li, S. Scientific collaborator recommendation via hypergraph embedding. *Information Processing & Management* **2026**, *63*, 104423.
62. Dong, X.; Ding, H.; Gao, D.; Wang, J.; Zheng, G. Joint Node–Hyperedge Contribution Hypergraph Embedding for Scraper Conveyor Sprocket Bearings Fault Diagnosis. *Available at SSRN 5390382*.
63. Zhu, Y.; Du, Y.; Wang, Y.; Xu, Y.; Zhang, J.; Liu, Q.; Wu, S. A survey on deep graph generation: Methods and applications. In Proceedings of the Learning on Graphs Conference. PMLR, 2022, pp. 47–1.
64. Ou, X.; Boyer, W.F.; McQueen, M.A. A scalable approach to attack graph generation. In Proceedings of the Proceedings of the 13th ACM conference on Computer and communications security, 2006, pp. 336–345.
65. Yang, J.; Lu, J.; Lee, S.; Batra, D.; Parikh, D. Graph r-cnn for scene graph generation. In Proceedings of the Proceedings of the European conference on computer vision (ECCV), 2018, pp. 670–685.
66. Zhao, L.; Song, Y.; Zhang, C.; Liu, Y.; Wang, P.; Lin, T.; Deng, M.; Li, H. T-GCN: A temporal graph convolutional network for traffic prediction. *IEEE transactions on intelligent transportation systems* **2019**, *21*, 3848–3858.
67. Guo, S.; Lin, Y.; Feng, N.; Song, C.; Wan, H. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In Proceedings of the Proceedings of the AAAI conference on artificial intelligence, 2019, Vol. 33, pp. 922–929.
68. Zhang, S.; Tong, H.; Xu, J.; Maciejewski, R. Graph convolutional networks: a comprehensive review. *Computational Social Networks* **2019**, *6*.
69. Bing, R.; Yuan, G.; Zhang, Y.; Wang, S.; Li, B.; Zhou, Y. An Efficient Multi-View Heterogeneous Hypergraph Convolutional Network for Heterogeneous Information Network Representation Learning. *IEEE Transactions on Big Data* **2024**.
70. Wang, J.; Li, H.; Qu, G.; Cecil, K.M.; Dillman, J.R.; Parikh, N.A.; He, L. Dynamic weighted hypergraph convolutional network for brain functional connectome analysis. *Medical image analysis* **2023**, *87*, 102828.
71. Nong, L.; Peng, J.; Zhang, W.; Lin, J.; Qiu, H.; Wang, J. Adaptive multi-hypergraph convolutional networks for 3D object classification. *IEEE Transactions on Multimedia* **2022**, *25*, 4842–4855.
72. Wang, Y.; Sun, Y.; Liu, Z.; Sarma, S.E.; Bronstein, M.M.; Solomon, J.M. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (tog)* **2019**, *38*, 1–12.
73. Beck, F.; Burch, M.; Diehl, S.; Weiskopf, D. The State of the Art in Visualizing Dynamic Graphs. *EuroVis (STARs)* **2014**.
74. Wang, S.; Wang, K.; Li, X. Dynamic Hypergraph Neural Networks for Flotation Condition Recognition Based on Group-View Composite Features. *IEEE Transactions on Instrumentation and Measurement* **2025**.
75. Zhang, Z.; Lin, H.; Gao, Y.; BNRist, K. Dynamic hypergraph structure learning. In Proceedings of the IJCAI, 2018, pp. 3162–3169.
76. Yin, N.; Feng, F.; Luo, Z.; Zhang, X.; Wang, W.; Luo, X.; Chen, C.; Hua, X.S. Dynamic hypergraph convolutional network. In Proceedings of the 2022 IEEE 38th International Conference on Data Engineering (ICDE). IEEE, 2022, pp. 1621–1634.
77. Fujita, T. SuperHyperGraph Neural Network and Dynamic SuperHyperGraph Neural Network. *International Journal of Complexity in Applied Science and Technology (IJCAST)* **2025**. Accepted.
78. Fujita, T.; Smarandache, F. A Unified Framework for U-Structures and Functorial Structure: Managing Super, Hyper, SuperHyper, Tree, and Forest Uncertain Over/Under/Off Models. *Neutrosophic Sets and Systems* **2025**, *91*, 337–380.
79. Broumi, S.; Talea, M.; Bakali, A.; Smarandache, F. Single valued neutrosophic graphs. *Journal of New theory* **2016**, *10*, 86–101.
80. Rosenfeld, A. Fuzzy graphs. In *Fuzzy sets and their applications to cognitive and decision processes*; Elsevier, 1975; pp. 77–95.
81. Mordeson, J.N.; Nair, P.S. *Fuzzy graphs and fuzzy hypergraphs*; Vol. 46, Physica, 2012.
82. Fujita, T. HyperWeighted Graph, SuperHyperWeighted Graph, and MultiWeighted Graph. *Pure Mathematics for Theoretical Computer Science* **2025**, *5*, 21–33.

83. Mathew, S.; Sunitha, M. Cycle connectivity in weighted graphs. *Proyecciones (Antofagasta)* **2011**, *30*, 1–17.
84. Banerjee, S.; Mukherjee, A.; Panigrahi, P.K. Quantum blockchain using weighted hypergraph states. *Physical Review Research* **2020**, *2*, 013322.
85. Fujita, T. Modeling Complex Hierarchical Systems with Weighted and Signed Superhypergraphs: Foundations and Applications. *Open Journal of Discrete Applied Mathematics (ODAM)* **2025**, *8*, 20–39. https://doi.org/10.30538/psrp-odam2025.0120.
86. Jose, B.K.; Tuza, Z. Hypergraph domination and strong independence. *Applicable Analysis and Discrete Mathematics* **2009**, *3*, 347–358.
87. Feng, S.; Heath, E.; Jefferson, B.; Joslyn, C.; Kvinge, H.; Mitchell, H.D.; Praggastis, B.; Eisfeld, A.J.; Sims, A.C.; Thackray, L.B.; et al. Hypergraph models of biological networks to identify genes critical to pathogenic viral response. *BMC bioinformatics* **2021**, *22*, 287.
88. Fujita, T. *Advancing Uncertain Combinatorics through Graphization, Hyperization, and Uncertainization: Fuzzy, Neutrosophic, Soft, Rough, and Beyond*; Biblio Publishing, 2025.
89. Marcos, B.V.S.; Willner, M.F.; Rosa, B.V.C.; Yissel, F.F.R.M.; Roberto, E.R.; Puma, L.D.B.; Fernández, D.M.M. Using plithogenic n-SuperHyperGraphs to assess the degree of relationship between information skills and digital competencies. *Neutrosophic Sets and Systems* **2025**, *84*, 513–524.
90. Amable, N.H.; De Salazar, E.E.V.; Isaac, M.G.M.; Sánchez, O.C.O.; Palma, J.M.S. Representation of motivational dynamics in school environments through Plithogenic n-SuperHyperGraphs with family participation. *Neutrosophic Sets and Systems* **2025**, *92*, 570–583.
91. Nacaroglu, Y.; Akgunes, N.; Pak, S.; Cangul, I.N. Some graph parameters of power set graphs. *Advances & Applications in Discrete Mathematics* **2021**, *26*.
92. Shalu, M.; Yamini, S.D. Counting maximal independent sets in power set graphs. *Indian Institute of Information Technology Design & Manufacturing (IIITD&M) Kancheepuram, India* **2014**.
93. Jech, T. *Set theory: The third millennium edition, revised and expanded*; Springer, 2003.
94. Bretto, A. Hypergraph theory. *An introduction. Mathematical Engineering. Cham: Springer* **2013**, *1*.
95. Smarandache, F. Foundation of SuperHyperStructure & Neutrosophic SuperHyperStructure. *Neutrosophic Sets and Systems* **2024**, *63*, 21.
96. Khali, H.E.; GÜNGÖR, G.D.; Zaina, M.A.N. Neutrosophic SuperHyper Bi-Topological Spaces: Original Notions and New Insights. *Neutrosophic Sets and Systems* **2022**, *51*, 3.
97. Smarandache, F. *Introduction to the n-SuperHyperGraph-the most general form of graph today*; Infinite Study, 2022.
98. Yu, H.; Gan, A.; Zhang, K.; Tong, S.; Liu, Q.; Liu, Z. Evaluation of retrieval-augmented generation: A survey. In Proceedings of the CCF Conference on Big Data. Springer, 2024, pp. 102–120.
99. Salemi, A.; Zamani, H. Evaluating retrieval quality in retrieval-augmented generation. In Proceedings of the Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2024, pp. 2395–2400.
100. Wang, X.; Wang, Z.; Gao, X.; Zhang, F.; Wu, Y.; Xu, Z.; Shi, T.; Wang, Z.; Li, S.; Qian, Q.; et al. Searching for best practices in retrieval-augmented generation. In Proceedings of the Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, 2024, pp. 17716–17736.
101. Jiang, Z.; Xu, F.F.; Gao, L.; Sun, Z.; Liu, Q.; Dwivedi-Yu, J.; Yang, Y.; Callan, J.; Neubig, G. Active retrieval augmented generation. In Proceedings of the Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, 2023, pp. 7969–7992.
102. Peng, B.; Zhu, Y.; Liu, Y.; Bo, X.; Shi, H.; Hong, C.; Zhang, Y.; Tang, S. Graph retrieval-augmented generation: A survey. *ACM Transactions on Information Systems* **2024**.
103. Edge, D.; Trinh, H.; Cheng, N.; Bradley, J.; Chao, A.; Mody, A.; Truitt, S.; Metropolitansky, D.; Ness, R.O.; Larson, J. From local to global: A graph rag approach to query-focused summarization. *arXiv preprint arXiv:2404.16130* **2024**.
104. Wu, J.; Zhu, J.; Qi, Y.; Chen, J.; Xu, M.; Menolascina, F.; Grau, V. Medical graph rag: Towards safe medical large language model via graph retrieval-augmented generation. *arXiv preprint arXiv:2408.04187* **2024**.
105. Cordeiro, D.; Mounié, G.; Perarnau, S.; Trystram, D.; Vincent, J.M.; Wagner, F. Random graph generation for scheduling simulations. In Proceedings of the 3rd International ICST Conference on Simulation Tools and Techniques (SIMUTools 2010). ICST, 2010, p. 10.
106. Liao, R.; Li, Y.; Song, Y.; Wang, S.; Hamilton, W.; Duvenaud, D.K.; Urtasun, R.; Zemel, R. Efficient graph generation with graph recurrent attention networks. *Advances in neural information processing systems* **2019**, *32*.

107. Kavvadias, D.; Stavropoulos, E. An efficient algorithm for the transversal hypergraph generation. *Journal of Graph Algorithms and Applications* **2005**, *9*, 239–264.
108. Lin, Z.; Yan, Q.; Liu, W.; Wang, S.; Wang, M.; Tan, Y.; Yang, C. Automatic hypergraph generation for enhancing recommendation with sparse optimization. *IEEE Transactions on Multimedia* **2023**, *26*, 5680–5693.
109. Gailhard, D.; Tartaglione, E.; Naviner, L.; Giraldo, J.H. HYGENE: A diffusion-based hypergraph generation method. In Proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence, 2025, Vol. 39, pp. 16682–16690.
110. Wen, W.; Yu, T. HyperPLR: Hypergraph Generation through Projection, Learning, and Reconstruction. In Proceedings of the The Thirteenth International Conference on Learning Representations, 2025.
111. Spirtes, P.; Glymour, C. An algorithm for fast recovery of sparse causal graphs. *Social science computer review* **1991**, *9*, 62–72.
112. Helmert, M. A Planning Heuristic Based on Causal Graph Analysis. In Proceedings of the ICAPS, 2004, Vol. 16, pp. 161–170.
113. Jiralerspong, T.; Chen, X.; More, Y.; Shah, V.; Bengio, Y. Efficient causal graph discovery using large language models. *arXiv preprint arXiv:2402.01207* **2024**.
114. Helmert, M.; Geffner, H. Unifying the Causal Graph and Additive Heuristics. In Proceedings of the ICAPS, 2008, pp. 140–147.
115. Xu, C.; Huang, H.; Yoo, S. Scalable causal graph learning through a deep neural network. In Proceedings of the Proceedings of the 28th ACM international conference on information and knowledge management, 2019, pp. 1853–1862.
116. Li, F.; Li, X.; Wen, S.; Bao, J. Multi-modal causal hypergraph reasoning for enhancing collaborative diagnosis of equipment composite failures. *Advanced Engineering Informatics* **2025**, *68*, 103611.
117. Harit, A.; Sun, Z.; Yu, J. From News to Returns: A Granger-Causal Hypergraph Transformer on the Sphere. In Proceedings of the Proceedings of the 6th ACM International Conference on AI in Finance, 2025, pp. 674–682.
118. Gôlo, M.P.S.; Marcacini, R.M. One-class edge classification through heterogeneous hypergraph for causal discovery. *Scientific Reports* **2025**.
119. Sun, Z.; Harit, A.; Lio, P. Actionable Interpretability via Causal Hypergraphs: Unravelling Batch Size Effects in Deep Learning. *arXiv preprint arXiv:2506.17826* **2025**.
120. Yang, H.; Gu, Y.; Zhu, J.; Hu, K.; Zhang, X. PGCN-TCA: Pseudo graph convolutional network with temporal and channel-wise attention for skeleton-based action recognition. *IEEE Access* **2020**, *8*, 10040–10047.
121. Lee, K.; Rhee, W. DDP-GCN: Multi-graph convolutional network for spatiotemporal traffic forecasting. *Transportation Research Part C: Emerging Technologies* **2022**, *134*, 103466.
122. Zhu, D.; Zhang, Z.; Cui, P.; Zhu, W. Robust Graph Convolutional Networks Against Adversarial Attacks. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* **2019**.
123. Xia, T.; Lin, J.; Li, Y.; Feng, J.; Hui, P.; Sun, F.; Guo, D.; Jin, D. 3dgcn: 3-dimensional dynamic graph convolutional network for citywide crowd flow prediction. *ACM Transactions on Knowledge Discovery from Data (TKDD)* **2021**, *15*, 1–21.
124. Liu, T.; Hu, Y.; Wang, B.; Sun, Y.; Gao, J.; Yin, B. Hierarchical graph convolutional networks for structured long document classification. *IEEE transactions on neural networks and learning systems* **2022**, *34*, 8071–8085.
125. Heilman, A.J.; Gong, W.; Yan, Q. Crystal Hypergraph Convolutional Networks. *arXiv preprint arXiv:2411.12616* **2024**.
126. Wang, Q.; Mao, Z.; Wang, B.; Guo, L. Knowledge graph embedding: A survey of approaches and applications. *IEEE transactions on knowledge and data engineering* **2017**, *29*, 2724–2743.
127. Ou, M.; Cui, P.; Pei, J.; Zhang, Z.; Zhu, W. Asymmetric Transitivity Preserving Graph Embedding. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* **2016**.
128. Cai, H.; Zheng, V.W.; Chang, K.C.C. A comprehensive survey of graph embedding: Problems, techniques, and applications. *IEEE transactions on knowledge and data engineering* **2018**, *30*, 1616–1637.
129. Wang, Z.; Zhang, J.; Feng, J.; Chen, Z. Knowledge graph embedding by translating on hyperplanes. In Proceedings of the Proceedings of the AAAI conference on artificial intelligence, 2014, Vol. 28.
130. Hayat, M.K.; Xue, S.; Wu, J.; Yang, J. Heterogeneous hypergraph embedding for node classification in dynamic networks. *IEEE Transactions on Artificial Intelligence* **2024**.
131. Dong, X.; Ding, H.; Gao, D.; Wang, J.; Zheng, G. Joint Node–Hyperedge Contribution Hypergraph Embedding for Scraper Conveyor Sprocket Bearings Fault Diagnosis. *Available at SSRN 5390382*.

132. Fatemi, B.; Taslakian, P.; Vazquez, D.; Poole, D. Knowledge hypergraph embedding meets relational algebra. *Journal of Machine Learning Research* **2023**, *24*, 1–34.
133. Goyal, P.; Ferrara, E. Graph embedding techniques, applications, and performance: A survey. *Knowledge-Based Systems* **2018**, *151*, 78–94.
134. Xu, M. Understanding graph embedding methods and their applications. *SIAM Review* **2021**, *63*, 825–853.
135. Chowdhary, K.; Chowdhary, K. Natural language processing. *Fundamentals of artificial intelligence* **2020**, pp. 603–649.
136. Bird, S.; Klein, E.; Loper, E. *Natural language processing with Python: analyzing text with the natural language toolkit*; " O'Reilly Media, Inc.", 2009.
137. Manning, C.D. *Foundations of statistical natural language processing*; The MIT Press, 1999.
138. Collobert, R.; Weston, J.; Bottou, L.; Karlen, M.; Kavukcuoglu, K.; Kuksa, P. Natural language processing (almost) from scratch. *Journal of machine learning research* **2011**, *12*, 2493–2537.
139. Manning, C.D.; Surdeanu, M.; Bauer, J.; Finkel, J.R.; Bethard, S.; McClosky, D. The Stanford CoreNLP natural language processing toolkit. In Proceedings of the Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations, 2014, pp. 55–60.
140. Adel, N. Fuzzy natural language similarity measures through computing with words. PhD thesis, Manchester Metropolitan University, 2022.
141. Glöckner, I.; Knoll, A. A formal theory of fuzzy natural language quantification and its role in granular computing. In *Granular computing: An emerging paradigm*; Springer, 2001; pp. 215–256.
142. Xie, J.; Zhao, L. Research on resume screening and career matching model based on fuzzy natural language processing. *Journal of Computational Methods in Sciences and Engineering* **2025**.
143. Fujita, T. Natural n-Superhyper Plithogenic Language. *Advancing Uncertain Combinatorics through Graphization, Hyperization, and Uncertainization: Fuzzy, Neutrosophic, Soft, Rough, and Beyond* **2025**, p. 294.
144. Fujita, T. Theoretical interpretations of large uncertain and hyper language models: Advancing natural uncertain and hyper language processing. *Advancing Uncertain Combinatorics through Graphization, Hyperization, and Uncertainization: Fuzzy, Neutrosophic, Soft, Rough, and Beyond* **2025**, p. 245.
145. Mihalcea, R.; Radev, D. *Graph-based natural language processing and information retrieval*; Cambridge university press, 2011.
146. Agarwal, V.; Joglekar, S.; Young, A.P.; Sastry, N. Graphnli: A graph-based natural language inference model for polarity prediction in online debates. In Proceedings of the Proceedings of the ACM Web Conference 2022, 2022, pp. 2729–2737.
147. Li, W.; Peng, R.; Wang, Y.; Yan, Z. Knowledge graph based natural language generation with adapted pointer-generator networks. *Neurocomputing* **2020**, *382*, 174–187.
148. Mihalcea, R.; Radev, D. Graph-Based Natural Language Processing and Information Retrieval, 2012.
149. Dumitriu, A.; Molony, C.; Daluwatte, C. Graph-based natural language processing for the pharmaceutical industry. In *Provenance in Data Science: From Data Models to Context-Aware Knowledge Graphs*; Springer, 2020; pp. 75–110.
150. Zadeh, L.A. Fuzzy sets. *Information and control* **1965**, *8*, 338–353.
151. Pawlak, Z. Rough sets. *International journal of computer & information sciences* **1982**, *11*, 341–356.
152. Pawlak, Z. Rough sets and intelligent data analysis. *Information sciences* **2002**, *147*, 1–12.
153. Hussain, S.S.; Durga, N.; Hossein, R.; Ganesh, G. New Concepts on Quadripartitioned Single-Valued Neutrosophic Graph with Real-Life Application. *International Journal of Fuzzy Systems* **2022**, *24*, 1515–1529.
154. Smarandache, F. A unifying field in Logics: Neutrosophic Logic. In *Philosophy*; American Research Press, 1999; pp. 1–141.
155. Fujita, T.; Smarandache, F. *A Dynamic Survey of Fuzzy, Intuitionistic Fuzzy, Neutrosophic, Plithogenic, and Extensional Sets*; Neutrosophic Science International Association (NSIA), 2025.
156. Smarandache, F. *Plithogenic set, an extension of crisp, fuzzy, intuitionistic fuzzy, and neutrosophic sets-revisited*; Infinite study, 2018.
157. Smarandache, F. Extension of soft set to hypersoft set, and then to plithogenic hypersoft set. *Neutrosophic sets and systems* **2018**, *22*, 168–170.