Exploration of Graph Classes and Concepts for SuperHypergraphs and n-th Power Mathematical Structures

Preprint · March 2025

DOI: 10.13140/RG.2.2.21482.02241

1 author:

Takaaki Fujita

Exploration of Graph Classes and Concepts for SuperHypergraphs and n-th Power Mathematical Structures

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

Abstract

A hypergraph extends this idea by allowing edges, referred to as hyperedges, to connect any number of vertices [30]. This paper explores superhypergraphs, an extension of hypergraphs incorporating superedges and supervertices. For example, Arboreal Superhypergraphs, Molecular superhypergraphs, and Probabilistic SuperHyperGraphs illustrate diverse structural types that can be modeled using superhypergraphs. We introduce the Generalized n-th Powerset, a formalized framework enabling broader mathematical applications while preserving the traditional n-th powerset structure. And we provide a brief exploration of *Natural Hyperlanguage Processing*, an extended framework of Natural Language Processing that leverages the concept of hyperlanguage for advanced applications. By extending hypergraph concepts to superhypergraphs, this work aims to advance their study and practical applicability.

Keywords: Superhypergraph, Hypergraph, Power set, nth Power set
MSC 2010 classifications: 05C65 - Hypergraphs, 68R10 - Graph theory in computer science

1 Short Introduction

1.1 Hypergraph and Superhypergraph

A graph is a mathematical structure used to represent relationships between entities through vertices and edges [85, 87]. Numerous applications of graph theory have been extensively studied [25, 41, 83, 142]. In graph theory, various graph classes have been extensively studied to suit the characteristics and structures of specific graphs [50].

A hypergraph extends this idea by allowing edges, referred to as hyperedges, to connect any number of vertices [30]. This structure can be seen as analogous to the power set in set theory. Hypergraphs are extensively studied and have found applications across a wide range of fields, including databases [171], neural networks [69, 100, 126], chemistry [230, 374], image representation [53, 179, 182], and VLSI design [60, 136, 193, 265, 300, 337]. Similar to general graphs, hypergraphs have been the subject of extensive research, with studies focusing on algorithms [124, 206, 276, 296], graph classes [7, 10, 13, 225, 226], and graph parameters [3, 4, 138, 140].

A superhypergraph extends the concept of a hypergraph by incorporating superedges and supervertices [114, 308, 309]. This structure can be likened to the n-th power set in set theory. Similarly, research has been conducted on algorithms [116], graph classes [112, 120, 151], and specific applications of superhypergraphs [118].

Due to their significance, superhyperstructures have been studied in contexts beyond graph theory as well [285, 311, 313, 315]. As a more abstracted graph concept compared to hypergraphs, the study of superhypergraphs is equally critical, and the author believes that further applications of superhypergraphs are highly promising.

1.2 Our Contribution in This Paper

This paper outlines our contributions to the field. While superhypergraphs have been explored in various studies, detailed research into their specific structures remains in its early stages. To address this, we aim to extend well-established hypergraph concepts to superhypergraphs. The natural progression from graphs to hypergraphs, with their mathematical structures and applications already being studied, makes it intuitive to further extend these concepts to superhypergraphs.

Some of the graph concepts discussed in this paper are listed below. Please refer to each subsection of the paper for further details.

- Arboreal SuperHypergraph: An Arboreal SuperHypergraph is a superhypergraph with a tree-like structure, representing hierarchical relationships among supervertices and superedges.
- Superhypergraph Morphism and Superhypergraph Isomorphism: Superhypergraph Morphism maps supervertices and superedges between superhypergraphs, preserving structure. Superhypergraph Isomorphism ensures structural equivalence between two superhypergraphs.
- Molecular n-superhypergraph: A Molecular n-SuperHypergraph extends molecular hypergraphs, modeling hierarchical molecular structures with n-level supervertices and superedges.
- Signed n-SuperHypergraph: A Signed n-SuperHypergraph assigns a positive or negative sign to each superedge, representing complex relationships in n-level superhypergraphs.
- Probabilistic SuperHyperGraph: A Probabilistic SuperHyperGraph assigns probabilities to superedges, modeling uncertainty and stochastic relationships in superhypergraph structures.
- Independent Set in a Superhypergraph: An Independent Set in a Superhypergraph is a subset of supervertices with no superedges fully contained within the subset.
- SuperHypergraph Ramsey numbers: SuperHypergraph Ramsey numbers determine the minimum supervertex count in a superhypergraph ensuring specific monochromatic substructures under edge-colorings.
- Multipartite SuperHypergraph: A Multipartite SuperHypergraph partitions supervertices into disjoint sets, ensuring no superedges connect supervertices within the same partition.
- SuperHypergraphic Sequence: A SuperHypergraphic Sequence lists supervertex degrees in a superhypergraph, representing the distribution of connections across its structure.
- Query n-superhypergraph: A Query n-SuperHypergraph models hierarchical query relationships, with supervertices representing data queries and dependencies.
- Superhypergraph Energy Functions: Superhypergraph Energy Functions measure the energy of a superhypergraph, derived from eigenvalues of its adjacency or incidence matrices.
- Transversal SuperHypergraph: A Transversal SuperHypergraph represents sets intersecting all superedges, modeling coverage relationships among supervertices in a superhypergraph.
- SuperHypernetwork: A SuperHypernetwork generalizes superhypergraphs, integrating supervertices and superedges to model multi-layered, interconnected systems and relationships.

Furthermore, we introduce the concept of the Generalized n-th Powerset to facilitate its application in various areas of mathematics. While the Generalized n-th Powerset retains the core mathematical framework of the traditional n-th powerset, it distinguishes itself by explicitly defining its structure, thereby enhancing its clarity and adaptability to a broader range of mathematical contexts. Finally, we provide a brief exploration of *Natural Hyperlanguage Processing*, an extended framework of Natural Language Processing that leverages the concept of hyperlanguage for advanced applications.

We hope that these contributions will support the development and dissemination of superhypergraph research and provide a solid foundation for future advancements in this field.

1.3 The Structure of the Paper

The format of this paper is described below.

2 Preliminaries and Definitions

This section introduces the essential background and definitions required for the concepts discussed in this paper. Readers interested in a more comprehensive understanding of graph theory are encouraged to explore standard references such as [85–87, 346]. Additionally, fundamental notions from set theory, which are relevant to this work, can be found in sources like [103, 155, 161, 176, 208]. For specific details about the operations and topics presented here, the cited references provide further elaboration.

2.1 Basic Concepts: Graphs and Hypergraphs

Graph theory is a pivotal mathematical tool for analyzing relationships between entities, represented as nodes (vertices) and their pairwise connections (edges). Hypergraphs expand upon this by introducing hyperedges, which can connect any number of vertices, making them suitable for representing more complex relationships [21, 22, 30, 139–141]. Below, we outline the definitions of graphs, subgraphs, and hypergraphs.

Definition 2.1 (Graph)**.** [87] A *graph* G is a mathematical structure represented as $G = (V, E)$, where:

- $V(G)$: The set of vertices (nodes).
- $E(G)$: The set of edges, where each edge connects two vertices, representing a relationship or interaction.

Definition 2.2 (Subgraph)**.** [87] Let $G = (V, E)$ be a graph. A *subgraph* $H = (V_H, E_H)$ of G is defined as follows:

- $V_H \subseteq V$: The vertex set of H is a subset of the vertex set of G.
- $E_H \subseteq E$: The edge set of H is a subset of the edge set of G.
- Every edge in E_H connects vertices within V_H.

Definition 2.3 (Hypergraph)**.** [30] A *hypergraph* $H = (V, E)$ generalizes the concept of a graph and is defined as:

- V: A set of vertices.
- E: A set of hyperedges, where each hyperedge $e \in E$ is a subset of V, i.e., $e \subseteq V$.

Properties:

- The hyperedge set E is a subset of the power set of V, i.e., $E \subseteq \mathcal{P}(V)$, where $\mathcal{P}(V)$ is the collection of all subsets of V.
- Unlike in traditional graphs, where edges connect exactly two vertices, hyperedges can connect any number of vertices, including a single vertex or the entire vertex set.

Proposition 2.4. *A hypergraph generalizes the concept of a graph by allowing edges, referred to as hyperedges, to connect more than two vertices.*

Proof. In a standard graph, each edge connects exactly two vertices. In contrast, a hypergraph extends this notion by permitting hyperedges to connect any subset of vertices, including sets with more than two elements. This broader structure encompasses traditional graphs as a special case where all hyperedges are limited to two vertices, thereby demonstrating the generalization. □

2.2 SuperHyperGraph

This subsection provides an overview of SuperHyperGraphs. A SuperHyperGraph is a class of graphs that achieves a higher level of generalization by utilizing superedges and supervertices. It serves as an extension of fundamental concepts such as graphs and hypergraphs (cf. [112, 112, 114, 117, 120, 131, 149, 151, 275, 308–310, 312, 315, 315, 316]). An n-SuperHyperGraph explicitly extends this concept, offering a more generalized framework for graph theory. The definitions and related concepts are detailed below.

Definition 2.5 (Powerset)**.** [279] The *powerset* of a set S, denoted $\mathcal{P}(S)$, is the set of all subsets of S, including the empty set and S itself. Formally,

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

Definition 2.6 (n-th powerset)**.** (cf. [301, 316]) The n-th powerset of H, denoted $P_n(H)$, is defined recursively as:

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

Similarly, the n-th non-empty powerset of H, denoted $P_n^*(H)$, is defined as:

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

Proposition 2.7. *A n-*th powerset *is a generalized concept of a powerset.*

Proof. This is evident. □

Definition 2.8 (n-SuperHyperGraph)**.** (cf. [308, 309]) Let V_0 be a finite set of base vertices. Define the n-th iterated power set of V_0 recursively as:

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

where $\mathcal{P}(A)$ denotes the power set of set A.

An *n-SuperHyperGraph* is an ordered pair $H = (V, E)$, where:

- $V \subseteq \mathcal{P}^n(V_0)$ is the set of *supervertices*, which are elements of the n-th power set of V_0.
- $E \subseteq \mathcal{P}^n(V_0)$ is the set of *superedges*, also elements of $\mathcal{P}^n(V_0)$.

Each supervertex $v \in V$ can be:

- A single vertex ($v \in V_0$),
- A subset of V_0 ($v \subseteq V_0$),
- A subset of subsets of V_0, up to n levels ($v \in \mathcal{P}^n(V_0)$),
- An indeterminate or fuzzy set(cf. [360]),
- The null set ($v = \emptyset$).

Each superedge $e \in E$ connects supervertices, potentially at different hierarchical levels up to n.

Proposition 2.9. *An n-SuperHyperGraph extends the concept of a hypergraph, incorporating higher-order structures and hierarchical relationships.*

Proof. The n-SuperHyperGraph generalizes a hypergraph by replacing vertices and edges with elements from the n-th iterated PowerSet. This hierarchical structure allows for the representation of relationships at multiple levels of abstraction, which directly extends the definition of a hypergraph. □

Proposition 2.10. *An n-SuperHyperGraph is a natural extension of a graph, enabling the representation of complex multi-level relationships.*

Proof. By definition, an n-SuperHyperGraph encompasses the classical graph as a special case when $n = 0$. The vertices and edges in a graph correspond to base-level elements in the n-th PowerSet. This embedding of graphs within n-SuperHyperGraphs demonstrates the generalization. □

Proposition 2.11. *[114] The structure of an n-SuperHyperGraph is built on the n-th iterated PowerSet, providing a robust framework for hierarchical modeling.*

Proof. This follows directly from the formal definition of the n-SuperHyperGraph, which recursively constructs its vertices and edges using the n-th PowerSet of a base set. For additional details, see [114]. □

A Superhypergraph and an n-SuperHyperGraph essentially share the same mathematical structure, with the primary difference being whether n is explicitly defined. Note that this distinction depends on the assumptions made in the paper.

We will now provide concrete examples of n-SuperHyperGraphs for $n = 0, 1, 2, 3$.

Example 2.12 (Case $n = 0$ of n-superhypergraph)**.** Let $V_0 = \{a, b, c\}$. Then:

$$\mathcal{P}^0(V_0) = V_0 = \{a, b, c\}.$$

An 0-SuperHyperGraph $H = (V, E)$ has:

- $V \subseteq V_0$.
- $E \subseteq V_0$.

Let $V = \{a, b\}$ and $E = \{c\}$.

Here, the supervertices are elements of V_0, and the superedges are also elements of V_0.

This case is basic, as both vertices and edges are simply elements of the base set V_0.

Example 2.13 (Case $n = 1$ of n-superhypergraph)**.** With the same $V_0 = \{a, b, c\}$, we have:

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

An 1-SuperHyperGraph $H = (V, E)$ has:

- $V \subseteq \mathcal{P}(V_0)$.
- $E \subseteq \mathcal{P}(V_0)$.

Let

- $V = \{\{a\}, \{b, c\}\}$.
- $E = \{\{a, b\}, \{c\}\}$.

In this case, the supervertices and superedges are subsets of V_0. This corresponds to a traditional hypergraph, where vertices are elements of $\mathcal{P}(V_0)$ (i.e., subsets of V_0).

Example 2.14 (Case $n = 2$ of n-superhypergraph)**.** Again, with $V_0 = \{a, b, c\}$, we compute:

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

First, list $\mathcal{P}(V_0)$ as before.

Then, $\mathcal{P}^2(V_0)$ is the set of all subsets of $\mathcal{P}(V_0)$.

An 2-SuperHyperGraph $H = (V, E)$ has:

- $V \subseteq \mathcal{P}^2(V_0)$.
- $E \subseteq \mathcal{P}^2(V_0)$.

Let

- $V = \{\{\{a\}, \{b\}\}, \{\{c\}, \{a, b\}\}\}$.
- $E = \{\{\{a, c\}, \{b, c\}\}\}$.

Here, the supervertices are subsets of $\mathcal{P}(V_0)$, i.e., sets whose elements are subsets of V_0.

For instance, $\{\{a\}, \{b\}\}$ is a supervertex consisting of two subsets of V_0: $\{a\}$ and $\{b\}$.

Example 2.15 (Case $n = 3$ of n-superhypergraph)**.** With $V_0 = \{a, b, c\}$, we have:

$$\mathcal{P}^3(V_0) = \mathcal{P}\left(\mathcal{P}\left(\mathcal{P}(V_0)\right)\right).$$

Elements of $\mathcal{P}^3(V_0)$ are subsets of $\mathcal{P}^2(V_0)$, which themselves are subsets of $\mathcal{P}(V_0)$.

An 3-SuperHyperGraph $H = (V, E)$ has:

- $V \subseteq \mathcal{P}^3(V_0)$.
- $E \subseteq \mathcal{P}^3(V_0)$.

Let

- $V = \{\{\{\{a\}, \{b\}\}, \{\{c\}\}\}\}$.
- $E = \{\{\{\{a, b\}\}, \{\{b, c\}\}\}\}$.

In this case, the supervertices are sets of elements from $\mathcal{P}^2(V_0)$, which are themselves sets of subsets of V_0.

For example, $\{\{\{a\}, \{b\}\}, \{\{c\}\}\}$ is a supervertex in V, where each element is a set of subsets of V_0.

3 Results in This Paper: Some Concepts for SuperHyperGraphs

In this section, we describe the results presented in this paper. We examine whether several hypergraph concepts can be extended to superhypergraphs. It is our hope that experts in the field will further explore practical applications of these extensions in the future.

3.1 Arboreal Superhypergraph

An Arboreal Hypergraph is a hypergraph with a tree-like structure, often used to model hierarchical relationships [30,52,78]. We extend this concept using superhypergraphs. The related definitions and theorems are provided below.

Definition 3.1 (Arboreal Hypergraph)**.** [30, 52, 78] A hypergraph $H = (V, E)$, where V is the set of vertices and E is the set of hyperedges, is called an *arboreal hypergraph* if it satisfies the following conditions:

1. H has the *Helly property*, meaning that for any collection of pairwise intersecting hyperedges, the entire collection has a non-empty intersection [88, 267].
2. For every cycle in H of length at least 3, there exist three hyperedges in the cycle that have a non-empty intersection.

Definition 3.2 (Co-Arboreal Hypergraph)**.** [52, 78] A hypergraph $H = (V, E)$ is called a *co-arboreal hypergraph* if it is the dual of an arboreal hypergraph. Formally:

1. H is *conformal*, i.e., every clique of the line graph of H corresponds to a hyperedge of H.
2. For every cycle in H of length at least 3, there exist three vertices in the cycle that are contained in the same hyperedge of H.

Definition 3.3 (Arboreal n-SuperHyperGraph)**.** An n-SuperHyperGraph $H = (V, E)$, where V is the set of supervertices and E is the set of superedges, is called an *Arboreal n-SuperHyperGraph* if it satisfies the following conditions:

1. H has the *Helly property*, meaning that for any collection of pairwise intersecting superedges, the entire collection has a non-empty intersection.
2. For every cycle in H of length at least 3, there exist three superedges in the cycle that have a non-empty intersection.

Definition 3.4 (Co-Arboreal n-SuperHyperGraph)**.** An n-SuperHyperGraph $H = (V, E)$ is called a *Co-Arboreal n-SuperHyperGraph* if it is the dual of an Arboreal n-SuperHyperGraph. Formally:

1. H is *conformal*, i.e., every clique of the line graph of H corresponds to a superedge of H.
2. For every cycle in H of length at least 3, there exist three supervertices in the cycle that are contained in the same superedge of H.

Theorem 3.5. *An Arboreal n-SuperHyperGraph generalizes the concept of an Arboreal Hypergraph.*

Proof. An Arboreal Hypergraph $H = (V, E)$ satisfies the Helly property and has the condition that every cycle of length at least 3 contains three hyperedges with a non-empty intersection. In the case of an Arboreal n-SuperHyperGraph, V and E are extended to elements of $\mathcal{P}^n(V_0)$, which encompasses standard vertices and edges as a special case when $n = 0$. Therefore, the conditions for the Helly property and cycles of length at least 3 are directly extended to the n-SuperHyperGraph structure, reducing to the original definition when $n = 0$. Thus, Arboreal n-SuperHyperGraphs generalize Arboreal Hypergraphs. □

Theorem 3.6. *A Co-Arboreal n-SuperHyperGraph generalizes the concept of a Co-Arboreal Hypergraph.*

Proof. A Co-Arboreal Hypergraph $H = (V, E)$ is the dual of an Arboreal Hypergraph and satisfies the conditions of conformality and that every cycle of length at least 3 contains three vertices in the same hyperedge. In a Co-Arboreal n-SuperHyperGraph, the vertices and edges are elements of $\mathcal{P}^n(V_0)$, thus extending the structural hierarchy. When $n = 0$, this structure naturally collapses to the definition of a Co-Arboreal Hypergraph. Therefore, Co-Arboreal n-SuperHyperGraphs generalize Co-Arboreal Hypergraphs. □

Theorem 3.7. *An Arboreal n-SuperHyperGraph is an n-SuperHyperGraph.*

Proof. An n-SuperHyperGraph $H = (V, E)$ has supervertices $V \subseteq \mathcal{P}^n(V_0)$ and superedges $E \subseteq \mathcal{P}^n(V_0)$. The definition of an Arboreal n-SuperHyperGraph imposes additional structural constraints on H, such as the Helly property and specific cycle intersection conditions. These properties do not alter the fundamental structure of H as an n-SuperHyperGraph because the elements of V and E remain subsets of $\mathcal{P}^n(V_0)$. Thus, an Arboreal n-SuperHyperGraph is an n-SuperHyperGraph. □

Theorem 3.8. *A Co-Arboreal n-SuperHyperGraph is an n-SuperHyperGraph.*

Proof. The dual of an Arboreal n-SuperHyperGraph, called a Co-Arboreal n-SuperHyperGraph, retains the supervertex and superedge structure of the original n-SuperHyperGraph. The dual operation swaps supervertices and superedges but does not modify their membership in $\mathcal{P}^n(V_0)$. Consequently, the structure of a Co-Arboreal n-SuperHyperGraph aligns with that of an n-SuperHyperGraph. □

3.2 Superhypergraph Morphism and Superhypergraph Isomorphism

A graph morphism is a mapping between graphs that preserves their structure and relationships [260, 262]. A graph isomorphism is a bijective mapping between graphs that preserves vertex adjacency [19, 20, 77, 101, 195, 235, 236, 274, 328]. These concepts have been extended to hypergraphs as hypergraph morphism [52] and hypergraph isomorphism [52, 99, 224, 249].

In this subsection, we investigate whether these notions can be further generalized to n-superhypergraphs. The related definitions and theorems are provided below.

Definition 3.9 (Hypergraph Morphism)**.** [52] Let $H = (V, E)$ and $H' = (V', E')$ be two hypergraphs without repeated hyperedges. A *morphism* of hypergraphs is a map $f : V \longrightarrow V'$ such that for every hyperedge $e \in E$, the image $f(e) \subseteq V'$ under f satisfies $f(e) \in E'$.

Definition 3.10 (Bijection)**.** (cf. [172]) A *bijection* is a function $f : A \longrightarrow B$ between two sets A and B that satisfies the following conditions:

- *Injective (One-to-One):* For all $x_1, x_2 \in A$, if $f(x_1) = f(x_2)$, then $x_1 = x_2$.
- *Surjective (Onto):* For every $y \in B$, there exists at least one $x \in A$ such that $f(x) = y$.

Definition 3.11 (Hypergraph Isomorphism)**.** [52] Two hypergraphs $H = (V, E)$ and $H' = (V', E')$ are *isomorphic*, denoted $H \cong H'$, if there exists:

- A bijection $f : V \longrightarrow V'$, and
- A bijection $\pi : I \longrightarrow J$ (where I and J are the index sets of E and E', respectively),

such that the induced map $g : E \longrightarrow E'$ defined by $g(e_i) = e'_{\pi(i)}$ satisfies:

$$g(e_i) = \{f(x) \mid x \in e_i\} \quad \text{for all } e_i \in E.$$

In this case, the pair (f, g) is called an *isomorphism of hypergraphs*.

Definition 3.12 (Hypergraph Automorphism)**.** [52] An *automorphism* of a hypergraph $H = (V, E)$ is an isomorphism (f, g) from H to itself. The set of all automorphisms of H, denoted $\mathrm{Aut}(H)$, forms a group under composition.

Definition 3.13 (n-SuperHyperGraph Morphism)**.** Let $H = (V, E)$ and $H' = (V', E')$ be two n-SuperHyperGraphs. A *morphism* $f : V \to V'$ between H and H' is a function such that for every superedge $e \in E$, the image $f(e) = \{f(v) \mid v \in e\} \subseteq V'$ satisfies $f(e) \in E'$.

In other words, f maps supervertices to supervertices and superedges to superedges via the induced map on edges.

Definition 3.14 (n-SuperHyperGraph Isomorphism)**.** Two n-SuperHyperGraphs $H = (V, E)$ and $H' = (V', E')$ are *isomorphic*, denoted $H \cong H'$, if there exists:

- A bijection $f : V \to V'$,

such that:

- For every superedge $e \in E$, the image $f(e) = \{f(v) \mid v \in e\} \in E'$.
- For every superedge $e' \in E'$, there exists $e \in E$ such that $f(e) = e'$.

In this case, f induces a bijection between E and E', and f is called an *isomorphism* of n-SuperHyperGraphs.

Definition 3.15 (n-SuperHyperGraph Automorphism)**.** An *automorphism* of an n-SuperHyperGraph $H = (V, E)$ is an isomorphism $f : V \to V$ from H to itself. The set of all automorphisms of H, denoted $\mathrm{Aut}(H)$, forms a group under composition.

Theorem 3.16. *An n-SuperHyperGraph Morphism generalizes the concept of a hypergraph morphism.*

Proof. When $n = 0$, the n-th iterated power set is $\mathcal{P}^0(V_0) = V_0$, so the supervertices are simply the base vertices V_0, and the superedges are subsets of V_0.

In this case, an n-SuperHyperGraph $H = (V, E)$ reduces to a standard hypergraph. The definition of an n-SuperHyperGraph morphism $f : V \to V'$ requires that for every edge $e \in E$, $f(e) \in E'$. This matches exactly the definition of a hypergraph morphism.

Therefore, n-SuperHyperGraph morphisms generalize hypergraph morphisms. □

Theorem 3.17. *An n-SuperHyperGraph Isomorphism generalizes the concept of a hypergraph isomorphism.*

Proof. Again, when $n = 0$, the n-SuperHyperGraph $H = (V, E)$ becomes a standard hypergraph with vertices V_0 and edges $E \subseteq \mathcal{P}(V_0)$.

An n-SuperHyperGraph isomorphism $f : V \to V'$ is a bijection such that $f(e) \in E'$ for all $e \in E$, and every edge in E' is the image of an edge in E. This coincides with the definition of a hypergraph isomorphism, where there is a bijection between the vertex sets that induces a bijection between the edge sets.

Therefore, n-SuperHyperGraph isomorphisms generalize hypergraph isomorphisms. □

Theorem 3.18. *An n-SuperHyperGraph Automorphism generalizes the concept of a hypergraph automorphism.*

Proof. When $n = 0$, an n-SuperHyperGraph automorphism $f : V \to V$ is a bijection from the vertex set to itself such that $f(e) \in E$ for all $e \in E$, meaning f maps edges to edges within the same hypergraph.

This matches the definition of a hypergraph automorphism, which is an isomorphism from a hypergraph to itself.

Therefore, n-SuperHyperGraph automorphisms generalize hypergraph automorphisms. □

3.3 Molecular n-superhypergraph

A Molecular Graph represents the structural formula of a molecule, modeling atoms as labeled nodes and bonds as labeled edges [128, 180, 194, 233, 263, 358, 367]. Molecular Graphs are closely related to Chemical Graphs [40, 127, 270, 329, 335]. A Molecular Hypergraph extends this concept, representing atoms as hyperedges and bonds as nodes connecting them [65, 185, 196, 198, 199, 253].

The formal definition is provided below.

Definition 3.19 (Atom)**.** (cf. [209, 336]) An *atom* is the basic unit of matter, consisting of a nucleus of protons and neutrons surrounded by electrons. In the context of molecular graphs, an atom is represented as a vertex labeled with its chemical symbol [343], such as H (hydrogen [177]) or C (carbon [327]).

Definition 3.20 (Bond)**.** (cf. [261, 287]) A *bond* is a connection between two atoms, representing the chemical interaction that holds them together. In molecular graphs, bonds are represented as edges labeled with their type, such as single, double, or triple bonds.

Definition 3.21. (cf. [194, 233, 263, 367]) A *Molecular Graph* is a graph $G = (V, E)$ that represents the structural formula of a molecule. In this representation:

- V: The vertex set represents the atoms in the molecule.
- E: The edge set represents the chemical bonds between pairs of atoms.

Each vertex $v \in V$ may have additional labels to denote the chemical element it represents (e.g., hydrogen, carbon, oxygen), and each edge $e \in E$ may have labels indicating the type of bond (e.g., single, double, or triple bonds).

Definition 3.22 (molecular hypergraph)**.** (cf. [65, 185, 196, 198, 199, 253]) A *molecular hypergraph* is a node and hyperedge-labeled hypergraph that models a molecule's atomic and bonding structure. Formally, a molecular hypergraph H is defined as an ordered quadruple $H = (V_H, E_H, \ell_H^{(V)}, \ell_H^{(E)})$, where:

- V_H is a finite set of nodes, representing bonds between atoms.
- E_H is a finite set of hyperedges, where each hyperedge $e \in E_H$ is a subset of V_H that represents an atom and its associated bonds.
- $\ell_H^{(V)} : V_H \to L_H^{(V)}$ is a node-labeling function, assigning a label to each node from a set $L_H^{(V)}$ of bond types.
- $\ell_H^{(E)} : E_H \to L_H^{(E)}$ is a hyperedge-labeling function, assigning a label to each hyperedge from a set $L_H^{(E)}$ of atomic properties.

Definition 3.23 (Molecular n-SuperHyperGraph)**.** Let V_0 be a finite set of base vertices representing bonds in a molecule. We define the n-th iterated power set of V_0 recursively as:

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

where $\mathcal{P}(A)$ denotes the power set of set A.

An *Molecular n-SuperHyperGraph* is defined as an ordered quadruple $H = (V_H, E_H, \ell_H^{(V)}, \ell_H^{(E)})$, where:

- $V_H \subseteq \mathcal{P}^n(V_0)$ is a finite set of *supernodes*, representing bonds or collections of bonds.
- $E_H \subseteq \mathcal{P}^n(V_0)$ is a finite set of *superhyperedges*, where each superhyperedge $e \in E_H$ connects elements of V_H at various hierarchical levels.
- $\ell_H^{(V)} : V_H \to L_H^{(V)}$ is a node-labeling function, assigning labels from a set $L_H^{(V)}$ of bond types or properties.

- $\ell_H^{(E)} : E_H \to L_H^{(E)}$ is a superedge-labeling function, assigning labels from a set $L_H^{(E)}$ of atomic or molecular properties.

Each supernode $v \in V_H$ can be:

- A single bond ($v \in V_0$),
- A subset of bonds ($v \subseteq V_0$),
- A higher-level collection up to n levels ($v \in \mathcal{P}^n(V_0)$),
- An indeterminate or fuzzy set (cf. [360]),
- The null set ($v = \emptyset$).

Theorem 3.24. *A Molecular n-SuperHyperGraph generalizes the concept of a Molecular Hypergraph.*

Proof. When $n = 0$, the n-th iterated power set reduces to $\mathcal{P}^0(V_0) = V_0$. Thus, the supernodes $V_H \subseteq V_0$ are simply the base nodes representing bonds, and the superhyperedges $E_H \subseteq V_0$ represent connections between these bonds.

In this case, the Molecular n-SuperHyperGraph $H = (V_H, E_H, \ell_H^{(V)}, \ell_H^{(E)})$ reduces to a standard Molecular Hypergraph, where:

- Nodes V_H represent bonds between atoms.
- Hyperedges E_H represent atoms connected via these bonds.
- Labeling functions $\ell_H^{(V)}$ and $\ell_H^{(E)}$ assign appropriate bond and atomic properties.

Therefore, the Molecular n-SuperHyperGraph encompasses the Molecular Hypergraph as a special case when $n = 0$, thereby generalizing it. □

Theorem 3.25. *Molecular n-SuperHyperGraphs are n-SuperHyperGraphs.*

Proof. A Molecular n-SuperHyperGraph $H = (V_H, E_H, \ell_H^{(V)}, \ell_H^{(E)})$ satisfies the structure of an n-SuperHyperGraph as follows:

1. By definition, $V_H \subseteq \mathcal{P}^n(V_0)$, where $\mathcal{P}^n(V_0)$ is the n-th iterated power set of the base vertex set V_0. Hence, V_H comprises supervertices that adhere to the hierarchical structure up to n levels.
2. Similarly, $E_H \subseteq \mathcal{P}^n(V_0)$, meaning that E_H contains superedges that align with the structure of n-SuperHyperGraphs.
3. The labeling functions $\ell_H^{(V)}$ and $\ell_H^{(E)}$ assign additional properties to vertices and edges but do not alter the structural definition of n-SuperHyperGraphs.

Thus, H meets all structural requirements of an n-SuperHyperGraph. □

3.4 Signed n-superhypergraph

A signed graph is a graph where each edge is assigned a positive or negative sign, modeling relationships with polarity [84, 164, 184, 227, 368]. The hypergraph counterpart is known as a signed hypergraph [152, 292, 293, 347, 359]. We extend these concepts using superhypergraphs. The related definitions and theorems are provided below.

Definition 3.26. [292] The *incidence matrix* of H, denoted by $\Phi(H)$, is a matrix of dimensions $|V(H)|\times|E(H)|$, where the entry $\Phi(H)_{i,j} = \varphi(v_i, e_j)$ indicates the incidence relationship between the i-th vertex and the j-th edge.

Definition 3.27. [292] A *signed hypergraph H* is formally defined as an ordered triple $H = (V(H), E(H), \varphi)$, where:

- $V(H)$ is a nonempty finite set of vertices.
- $E(H)$ is a nonempty finite set of edges, where each edge $e \in E(H)$ is a subset of $V(H)$, i.e., $e \subseteq V(H)$.
- $\varphi : V(H)\times E(H) \to \{-1, 0, 1\}$ is an *incidence function*, which assigns a value to each pair (v, e), where:
  - $\varphi(v, e) = 1$: v is positively incident with e.
  - $\varphi(v, e) = -1$: v is negatively incident with e.
  - $\varphi(v, e) = 0$: v is not incident with e.

Example 3.28. (cf. [292]) In the context of signed hypergraphs, the following special cases are well-known:

- A *signed graph* is a specific instance of a signed hypergraph where all edges have exactly two incident vertices, i.e., $\delta(e) = 2$ for all $e \in E(H)$.
- A *hypergraph* is a particular case of a signed hypergraph where the incidence function satisfies $\varphi(v, e) \in \{0, 1\}$ for all (v, e), meaning all incidences are positive.

Definition 3.29. Let V_0 be a finite set of base vertices. Define the n-th iterated power set of V_0 recursively as before.

A *Signed n-SuperHyperGraph* is defined as an ordered triple $H = (V(H), E(H), \varphi)$, where:

- $V(H) \subseteq \mathcal{P}^n(V_0)$ is a nonempty finite set of *supervertices*.
- $E(H) \subseteq \mathcal{P}^n(V_0)$ is a nonempty finite set of *superedges*.
- $\varphi : V(H) \times E(H) \to \{-1, 0, 1\}$ is an *incidence function*, assigning a value to each pair (v, e), where:
  - $\varphi(v, e) = 1$: v is positively incident with e.
  - $\varphi(v, e) = -1$: v is negatively incident with e.
  - $\varphi(v, e) = 0$: v is not incident with e.

Theorem 3.30. *A Signed n-SuperHyperGraph generalizes the concept of a Signed Hypergraph.*

Proof. When $n = 0$, $V(H) \subseteq \mathcal{P}^0(V_0) = V_0$ and $E(H) \subseteq \mathcal{P}^0(V_0) = V_0$. In this scenario, the supervertices and superedges are elements of the base set V_0.

The incidence function $\varphi : V_0 \times V_0 \to \{-1, 0, 1\}$ defines the relationships between vertices and edges as in a standard Signed Hypergraph.

Thus, the Signed n-SuperHyperGraph reduces to a Signed Hypergraph when $n = 0$, and therefore generalizes it. □

Theorem 3.31. *Signed n-SuperHyperGraphs are n-SuperHyperGraphs.*

Proof. A Signed n-SuperHyperGraph $H = (V(H), E(H), \varphi)$ satisfies the structure of an n-SuperHyperGraph as follows:

1. By definition, $V(H) \subseteq \mathcal{P}^n(V_0)$, where $\mathcal{P}^n(V_0)$ is the n-th iterated power set of the base vertex set V_0. Hence, $V(H)$ comprises supervertices that satisfy the hierarchical structure up to n levels.
2. Similarly, $E(H) \subseteq \mathcal{P}^n(V_0)$, meaning that $E(H)$ contains superedges that conform to the structure of n-SuperHyperGraphs.
3. The incidence function $\varphi : V(H) \times E(H) \to \{-1, 0, 1\}$ introduces signed relationships between supervertices and superedges but does not alter their structural definitions.

Thus, H meets all structural requirements of an n-SuperHyperGraph. □

3.5 Probabilistic n-SuperHyperGraph

A Probabilistic Graph is a graph where edges are assigned probabilities, capturing uncertainty in connections (cf. [96, 105, 154, 197, 203, 290]). The Probabilistic Hypergraph is an extension of this concept to hypergraphs, where hyperedges are associated with probabilities [166, 202, 217, 222, 248]. Various studies have explored its applications and properties. This concept is further generalized to n-SuperHyperGraphs. The related definitions and theorems are outlined below.

Definition 3.32. (cf. [178, 281]) Probability is a measure quantifying the likelihood of an event occurring, ranging from 0 (impossible) to 1 (certain).

Definition 3.33 (Probabilistic Graph)**.** [105, 197] A *Probabilistic Graph* is defined as a triplet $G = (V, E, A)$, where:

- V is a finite set of vertices.
- $E \subseteq \binom{V}{2}$ is a set of edges, where each $e \in E$ is an unordered pair of vertices from V.
- $A : V \times V \to [0, 1]$ is an *affinity matrix* or *probability matrix*, where $A(i, j)$ represents the probability or weight of connection between vertices $v_i, v_j \in V$.

Edge Weight: For each edge $e = \{v_i, v_j\} \in E$, the weight $w(e)$ is defined as:

$$w(e) = A(v_i, v_j).$$

Vertex Degree: The degree of a vertex $v \in V$ is defined as:

$$d(v) = \sum_{u \in V, \{v,u\} \in E} w(\{v, u\}).$$

Adjacency Matrix: The adjacency matrix M of the probabilistic graph is given by:

$$M(i, j) = \begin{cases} A(v_i, v_j), & \text{if } \{v_i, v_j\} \in E, \\ 0, & \text{otherwise.} \end{cases}$$

Definition 3.34 (Centroid in Hypergraphs)**.** [166] Let V be a finite set of vertices, and $A : V \times V \to [0, 1]$ be a similarity matrix, where $A(i, j)$ quantifies the similarity between vertices $v_i, v_j \in V$.

A vertex $v_j \in V$ is called the *centroid* of a hyperedge $e \subseteq V$ if:

1. v_j is chosen based on a predefined criterion, such as:

- *Maximum similarity* to other vertices:

$$v_j = \arg\max_{v \in V} \sum_{v_i \in e} A(v, v_i).$$

- *Predefined property*, such as an initial label or domain-specific ranking.

2. The hyperedge e is formed as:

$$e = \{v_j\} \cup \{v_i \mid v_i \in \text{neighbors of } v_j \text{ based on a similarity threshold or } k\text{-nearest neighbors}\}.$$

Definition 3.35 (Probabilistic Hypergraph)**.** [166] A *Probabilistic Hypergraph* is defined as a triplet $G = (V, E, A)$, where:

- V is a finite set of vertices.
- $E \subseteq \mathcal{P}(V)$ is a set of hyperedges, where each $e \in E$ is a subset of V.
- $A : V \times V \to [0, 1]$ is an *affinity matrix* that quantifies the similarity or probability of connection between vertices. Specifically, $A(i, j)$ represents the similarity between vertices $v_i, v_j \in V$.

The *incidence matrix* H of the probabilistic hypergraph is a $|V| \times |E|$ matrix defined as:

$$H(i, j) = \begin{cases} A(v_j, v_i), & \text{if } v_i \in e_j \text{ and } v_j \text{ is the centroid of } e_j, \\ 0, & \text{otherwise.} \end{cases}$$

Hyperedge Weight: For each hyperedge $e \in E$, the weight $w(e)$ is computed as:

$$w(e) = \sum_{v_i \in e} A(v_j, v_i),$$

where v_j is the centroid vertex of the hyperedge e.

Vertex Degree: The degree of a vertex $v \in V$ is defined as:

$$d(v) = \sum_{e \in E} w(e) \cdot H(v, e).$$

Hyperedge Degree: The degree of a hyperedge $e \in E$ is given by:

$$\delta(e) = \sum_{v \in e} H(v, e).$$

Definition 3.36 (Probabilistic n-SuperHyperGraph)**.** Let V_0 be a finite set of base vertices. Define the n-th iterated power set of V_0 recursively as:

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

where $\mathcal{P}(A)$ denotes the power set of the set A.

An *n-SuperHyperGraph* is an ordered pair $H = (V, E)$, where:

- $V \subseteq \mathcal{P}^n(V_0)$ is the set of *supervertices*.
- $E \subseteq \mathcal{P}^n(V_0)$ is the set of *superedges*.

A *Probabilistic n-SuperHyperGraph* is defined as a triplet $G = (V, E, A)$, where:

- V and E are as defined above.
- $A : V \times V \to [0, 1]$ is an *affinity function* assigning a probability or similarity measure between pairs of supervertices.

The *incidence matrix* H is a $|V| \times |E|$ matrix defined by:

$$H(i, j) = \begin{cases} A(v_j, v_i), & \text{if } v_i \in e_j \text{ and } v_j \text{ is the centroid of } e_j, \\ 0, & \text{otherwise.} \end{cases}$$

superedge Weight: For each superedge $e \in E$, the weight $w(e)$ is calculated as:

$$w(e) = \sum_{v_i \in e} A(v_j, v_i),$$

where v_j is the centroid supervertex of the superedge e.

Vertex Degree: The degree of a vertex $v \in V$ is defined as:

$$d(v) = \sum_{e \in E} w(e) \cdot H(v, e).$$

superedge Degree: The degree of a superedge $e \in E$ is given by:

$$\delta(e) = \sum_{v \in e} H(v, e).$$

Theorem 3.37. *A Probabilistic n-SuperHyperGraph is an n-SuperHyperGraph.*

Proof. By definition, a Probabilistic n-SuperHyperGraph $G = (V, E, A)$ possesses supervertices $V \subseteq \mathcal{P}^n(V_0)$ and superedges $E \subseteq \mathcal{P}^n(V_0)$, fulfilling the criteria of an n-SuperHyperGraph $H = (V, E)$. The introduction of the affinity function A and the probabilistic incidence matrix H adds probabilistic characteristics but does not alter the fundamental structure of supervertices and superedges. Therefore, G retains the structure of an n-SuperHyperGraph. □

Theorem 3.38. *A Probabilistic n-SuperHyperGraph generalizes the Probabilistic HyperGraph.*

Proof. When $n = 0$, the n-th iterated power set simplifies to $\mathcal{P}^0(V_0) = V_0$, so the supervertices and superedges reduce to elements and subsets of the base vertex set V_0. In this scenario, the Probabilistic n-SuperHyperGraph $G = (V, E, A)$ becomes a Probabilistic HyperGraph with vertex set V_0, hyperedge set $E \subseteq \mathcal{P}(V_0)$, and affinity function $A : V_0 \times V_0 \to [0, 1]$. The definitions of the incidence matrix H, hyperedge weights $w(e)$, and degrees $d(v)$ and $\delta(e)$ coincide with those in the Probabilistic HyperGraph. Thus, the Probabilistic n-SuperHyperGraph generalizes the Probabilistic HyperGraph. □

Question 3.39. Is it possible to define a Bayesian n-superhypergraph as an extension of Bayesian hypergraphs [174, 175, 342]? Additionally, can the concept of a Markov chain in hypergraphs [49, 220] be extended to n-superhypergraphs? What are the potential mathematical structures and applications of such an extension?

3.6 Independent Set in a Superhypergraph

An independent set in a graph is a set of vertices such that no two vertices in the set are connected by an edge [144, 223]. Similarly, an independent set in a hypergraph is a subset of vertices that does not contain any hyperedge as a subset, extending the concept of independence to higher-dimensional relationships [23, 42, 153, 181, 201]. This concept can be further defined in the context of a superhypergraph. The relevant definitions and theorem are presented below.

Definition 3.40 (Independent Set in a Hypergraph). [23] Let $H = (V(H), E(H))$ be a hypergraph, where $V(H)$ is the set of vertices and $E(H) \subseteq 2^{V(H)}$ is the set of hyperedges. A subset $I \subseteq V(H)$ is called an *independent set* in H if I does not contain any hyperedge of H as a subset. Formally,

$$I \text{ is independent} \iff \forall e \in E(H),\ e \nsubseteq I.$$

Definition 3.41 (Independent Set in an n-SuperHyperGraph). Let $H = (V, E)$ be an n-SuperHyperGraph. A subset $I \subseteq V$ is called an *independent set* in H if I does not contain any superedge $e \in E$ as a subset. Formally,

$$I \text{ is independent} \iff \forall e \in E,\ e \nsubseteq I.$$

Theorem 3.42. *The concept of an independent set in an n-SuperHyperGraph generalizes the notion of an independent set in a hypergraph. In particular, a hypergraph is equivalent to a 1-SuperHyperGraph.*

Proof. Let $H = (V, E)$ be a hypergraph. By definition, $V \subseteq V_0$ and $E \subseteq 2^{V_0}$, where V_0 is the base set of vertices. A hypergraph can be interpreted as a 1-SuperHyperGraph, since:

$$\mathcal{P}^1(V_0) = \mathcal{P}(V_0).$$

For a 1-SuperHyperGraph $H = (V, E)$, the vertices and edges satisfy $V, E \subseteq \mathcal{P}^1(V_0)$, and the independence condition $I \subseteq V$ with $e \nsubseteq I$ for all $e \in E$ is exactly the same as the definition of independence in a hypergraph.

For $n > 1$, the vertices and edges $V, E \subseteq \mathcal{P}^n(V_0)$ involve higher levels of hierarchical relationships. However, the independence condition $e \nsubseteq I$ remains consistent across all levels of n. Thus, the definition of independence in n-SuperHyperGraphs generalizes the concept from hypergraphs.

Therefore, a hypergraph is specifically a 1-SuperHyperGraph, and the concept of independence is naturally extended to n-SuperHyperGraphs for $n \geq 1$. □

3.7 n-SuperHypergraph Ramsey numbers

The *Graph Ramsey Number* is the smallest N such that any red-blue edge coloring of K_N contains a red K_s or a blue K_t [28, 64, 97, 143, 268]. The *Hypergraph Ramsey Number* is the smallest N such that any red-blue coloring of k-element subsets of $[N]$ contains a monochromatic k-uniform hypergraph of size s or t [75, 76, 89, 200, 244, 245]. These concepts are extended to superhypergraphs. The relevant definitions and theorems are presented below.

Definition 3.43 (Complete Graph). (cf. [2, 68]) A *complete graph*, denoted K_n, is a graph where:

- The vertex set $V(K_n)$ consists of n vertices: $V(K_n) = \{v_1, v_2, \ldots, v_n\}$.
- The edge set $E(K_n)$ contains all possible $\binom{n}{2}$ edges, where each edge connects two distinct vertices v_i and v_j $(i \neq j)$.

In K_n, every vertex has a degree of $n - 1$, and the graph is maximally connected.

Definition 3.44 (Graph edge coloring). [58, 160, 371] In general, graph edge coloring is the assignment of colors to the edges of a graph such that no two edges sharing the same vertex have the same color.

Definition 3.45 (Graph Ramsey Number). [28, 64, 97, 143, 268] The *Graph Ramsey Number*, denoted $R(s, t)$, is the smallest positive integer N such that any red-blue coloring of the edges of a complete graph K_N on N vertices contains:

- A red K_s (a complete subgraph of s vertices with all edges colored red), or
- A blue K_t (a complete subgraph of t vertices with all edges colored blue).

Formally,

$$R(s, t) = \min \{N \mid \forall \text{ red-blue edge colorings of } K_N, \exists \text{ a red } K_s \text{ or a blue } K_t\}.$$

Definition 3.46 (k-Uniform Hypergraph)**.** [76, 159, 247, 278] A k-uniform hypergraph $H = (V, E)$ is a hypergraph where:

- V is the set of vertices.
- $E \subseteq \binom{V}{k}$, the set of all k-element subsets of V. Each $e \in E$ is called a k-uniform hyperedge.

Definition 3.47 (Monochromatic k-Uniform Hypergraph)**.** [76] A k-uniform hypergraph $H = (V, E)$ is said to be *monochromatic* under a coloring if all hyperedges $e \in E$ are assigned the same color.

More formally, let $\chi : \binom{V}{k} \to \{c_1, c_2, \ldots, c_m\}$ be a coloring function assigning one of m colors to each k-tuple of V. The k-uniform hypergraph $H = (V, E)$ is monochromatic if there exists a color $c \in \{c_1, c_2, \ldots, c_m\}$ such that:

$$\forall e \in E, \ \chi(e) = c.$$

Definition 3.48 (Monochromatic Subset in a k-Uniform Hypergraph)**.** [76] Given a k-uniform hypergraph $H = (V, E)$ with a coloring $\chi : \binom{V}{k} \to \{c_1, c_2, \ldots, c_m\}$, a subset $S \subseteq V$ is called a *monochromatic subset* if:

$$\forall e \in \binom{S}{k}, \ \chi(e) = c,$$

for some fixed color $c \in \{c_1, c_2, \ldots, c_m\}$.

Definition 3.49 (Hypergraph Ramsey Numbers)**.** [76] Let k, s, and n be positive integers. The k-uniform hypergraph Ramsey number, denoted $r_k(s, n)$, is the smallest positive integer N such that, for every red-blue coloring of the k-element subsets of an N-element set $[N]$, one of the following holds:

1. There exists a subset $S \subseteq [N]$ with $|S| = s$ such that every k-tuple of S is red.
2. There exists a subset $T \subseteq [N]$ with $|T| = n$ such that every k-tuple of T is blue.

Formally,

$$r_k(s, n) = \min \left\{ N \mid \forall \text{ red-blue coloring of } \binom{[N]}{k}, \exists \text{ monochromatic } k\text{-uniform hypergraph with size } s \text{ or } n \right\}.$$

Definition 3.50 (k-Uniform n-SuperHypergraph)**.** Let $n \geq 1$ and $k \geq 1$ be integers, and let V_0 be a finite set. Let $V = \mathcal{P}^{n-1}(V_0)$ be the set of vertices.

A *k-uniform n-SuperHypergraph* is a hypergraph $H = (V, E)$, where:

- $V = \mathcal{P}^{n-1}(V_0)$ is the vertex set.
- $E \subseteq \binom{V}{k}$, the set of all k-element subsets of V.

Definition 3.51 (Monochromatic k-Uniform n-SuperHypergraph)**.** Let $H = (V, E)$ be a k-uniform n-SuperHypergraph, and let $\chi : \binom{V}{k} \to \{c_1, c_2, \ldots, c_m\}$ be a coloring function assigning one of m colors to each edge $e \in E$. We say that H is *monochromatic* if there exists a color $c \in \{c_1, c_2, \ldots, c_m\}$ such that:

$$\forall e \in E, \quad \chi(e) = c.$$

Definition 3.52 (n-SuperHypergraph Ramsey Numbers)**.** Let $n \geq 1$, $k \geq 1$, and s, t be positive integers. The n-SuperHypergraph Ramsey number, denoted $r_n^{(k)}(s, t)$, is the smallest positive integer N such that, for every red-blue coloring χ of the edges in $\binom{V}{k}$ with $V = \mathcal{P}^{n-1}(V_0)$ and $|V_0| = N$, one of the following holds:

1. There exists a subset $S \subseteq V$ with $|S| = s$ such that all k-element subsets of S are colored red.

2. There exists a subset $T \subseteq V$ with $|T| = t$ such that all k-element subsets of T are colored blue.

Formally,

$$r_n^{(k)}(s,t) = \min \left\{ N \,\middle|\, \forall \text{ red-blue coloring } \chi \text{ of } \binom{V}{k}, \exists \text{ monochromatic } k\text{-uniform } n\text{-SuperHypergraph of size } s \text{ or } t \right\}.$$

Theorem 3.53. *The concept of n-SuperHypergraph Ramsey numbers generalizes hypergraph Ramsey numbers. In particular, when $n = 1$, the n-SuperHypergraph Ramsey number $r_1^{(k)}(s,t)$ coincides with the classical hypergraph Ramsey number $r_k(s,t)$.*

Proof. When $n = 1$, we have:

$$\mathcal{P}^{n-1}(V_0) = \mathcal{P}^0(V_0) = V_0.$$

Thus, the vertex set is $V = V_0$.

The edge set is $E \subseteq \binom{V}{k} = \binom{V_0}{k}$.

This corresponds exactly to a classical k-uniform hypergraph on the vertex set V_0.

In the classical hypergraph Ramsey problem, we consider the smallest integer N such that any red-blue coloring of the edges of the complete k-uniform hypergraph on N vertices contains a monochromatic complete k-uniform hypergraph of size s in red or t in blue.

Therefore, $r_1^{(k)}(s,t) = r_k(s,t)$.

This shows that the n-SuperHypergraph Ramsey numbers generalize the classical hypergraph Ramsey numbers. □

Question 3.54. Is it possible to propose Anti-Ramsey theorems [95] in the context of n-SuperHypergraphs?

3.8 Tripartite n-SuperHypergraph and Multipartite n-SuperHypergraph

In general, a *tripartite graph* is a graph in which the vertex set is divided into three disjoint subsets, with no edges connecting vertices within the same subset [251,291,373,380]. Tripartite graphs have been extensively studied for practical applications in fields such as personalized recommendation systems [221,373]. A tripartite graph can also be viewed as an extended version of a bipartite graph [17, 90, 145]. A *multipartite graph* is a graph where the vertex set is partitioned into k disjoint subsets, ensuring that no two vertices within the same subset are adjacent [46,81,98]. These concepts, when extended to hypergraphs, lead to the notions of *Tripartite Hypergraphs* [133, 134, 170, 216, 372] and *Multipartite Hypergraphs* [1, 47]. A more structured version, the *k-Uniform Multipartite Hypergraph*, has also been widely studied in this context [47]. This subsection introduces a further generalization of these concepts to *superhypergraphs*, as described below. It is worth noting that in this paper, the definition of a tripartite hypergraph follows the user–item–tag tripartite hypergraph model proposed in [372].

Definition 3.55 (Tripartite Hypergraph). [372] A *tripartite hypergraph* is a hypergraph $G = (V, H)$ where:

- $V = U \cup R \cup T$, where U, R, and T are disjoint vertex sets representing users, resources, and tags, respectively.
- $H \subseteq U \times R \times T$, the set of hyperedges, where each hyperedge $h = (u, r, t)$ consists of one user $u \in U$, one resource $r \in R$, and one tag $t \in T$.

Definition 3.56 (Properties of a Tripartite Hypergraph). [372] Given a tripartite hypergraph $G = (V, H)$:

- The *hyperdegree* of a node $v \in V$ is the number of hyperedges in H that contain v.

- The *clustering coefficient* of a node $v \in V$ is the ratio of the actual number of hyperedges involving v to the maximum possible number of such hyperedges, based on the degrees of its neighbors [36, 289].
- The *average distance* is the average shortest path length between two random nodes in G, considering paths that traverse hyperedges [71, 72].

Definition 3.57 (k-Uniform Multipartite Hypergraph)**.** (cf. [47]) A k-uniform multipartite hypergraph is a hypergraph $H = (V, E)$, where:

- $V = V_1 \cup V_2 \cup \cdots \cup V_k$ is the vertex set, partitioned into k disjoint subsets $V_1, V_2, \ldots, V_k$, called the *vertex classes*.
- $E \subseteq V_1 \times V_2 \times \cdots \times V_k$, the set of hyperedges, where each hyperedge $e \in E$ is a k-tuple such that $|e \cap V_i| = 1$ for all $i = 1, 2, \ldots, k$.

Definition 3.58 (Tripartite n-SuperHypergraph)**.** Let V_0 be a finite set. The n-th iterated power set of V_0, denoted $\mathcal{P}^n(V_0)$, is defined recursively as:

$$\mathcal{P}^0(V_0) = V_0, \quad \mathcal{P}^{k+1}(V_0) = \mathcal{P}\left(\mathcal{P}^k(V_0)\right), \quad \text{for } k \geq 0,$$

where $\mathcal{P}(A)$ denotes the power set of the set A.

And let $n \geq 1$, and let U_0, R_0, and T_0 be finite, disjoint base sets representing users, resources, and tags, respectively.

Define the vertex classes:

$$U = \mathcal{P}^{n-1}(U_0), \quad R = \mathcal{P}^{n-1}(R_0), \quad T = \mathcal{P}^{n-1}(T_0).$$

A *Tripartite n-SuperHypergraph* is a hypergraph $G = (V, H)$, where:

- $V = U \cup R \cup T$ is the vertex set, partitioned into three disjoint classes.
- $H \subseteq U \times R \times T$ is the set of hyperedges, where each hyperedge $h = (u, r, t)$ consists of one supervertex $u \in U$, one supervertex $r \in R$, and one supervertex $t \in T$.

Definition 3.59 (k-Uniform Multipartite n-SuperHypergraph)**.** Let $n \geq 1$, $k \geq 1$, and let $V_{0,1}, V_{0,2}, \ldots, V_{0,k}$ be finite, disjoint base sets.

Define the vertex classes:

$$V_i = \mathcal{P}^{n-1}(V_{0,i}), \quad \text{for } i = 1, 2, \ldots, k.$$

A *k-Uniform Multipartite n-SuperHypergraph* is a hypergraph $H = (V, E)$, where:

- $V = V_1 \cup V_2 \cup \cdots \cup V_k$ is the vertex set, partitioned into k disjoint classes.
- $E \subseteq V_1 \times V_2 \times \cdots \times V_k$ is the set of hyperedges, where each hyperedge $e = (v_1, v_2, \ldots, v_k)$ consists of one supervertex $v_i \in V_i$ from each vertex class.

Theorem 3.60. *The concept of a Tripartite n-SuperHypergraph generalizes that of a Tripartite Hypergraph. Specifically, when $n = 1$, a Tripartite n-SuperHypergraph reduces to a Tripartite Hypergraph.*

Proof. When $n = 1$, we have:

$$\mathcal{P}^{n-1}(U_0) = \mathcal{P}^0(U_0) = U_0, \quad \mathcal{P}^{n-1}(R_0) = R_0, \quad \mathcal{P}^{n-1}(T_0) = T_0.$$

Therefore, the vertex classes are:

$$U = U_0, \quad R = R_0, \quad T = T_0.$$

The hyperedges are subsets of $U \times R \times T$, where each hyperedge $h = (u, r, t)$ consists of one element from each of U_0, R_0, and T_0.

This matches the definition of a Tripartite Hypergraph, where $V = U_0 \cup R_0 \cup T_0$, and $H \subseteq U_0 \times R_0 \times T_0$.

Therefore, the Tripartite n-SuperHypergraph with $n = 1$ is equivalent to a Tripartite Hypergraph. □

Theorem 3.61. *The concept of a k-Uniform Multipartite n-SuperHypergraph generalizes that of a k-Uniform Multipartite Hypergraph. Specifically, when $n = 1$, a k-Uniform Multipartite n-SuperHypergraph reduces to a k-Uniform Multipartite Hypergraph.*

Proof. When $n = 1$, we have:

$$V_i = \mathcal{P}^{n-1}(V_{0,i}) = \mathcal{P}^0(V_{0,i}) = V_{0,i}, \quad \text{for } i = 1, 2, \ldots, k.$$

Therefore, the vertex classes are $V_i = V_{0,i}$, and the vertex set is $V = V_{0,1} \cup V_{0,2} \cup \cdots \cup V_{0,k}$.

The hyperedges are subsets of $V_1 \times V_2 \times \cdots \times V_k$, where each hyperedge $e = (v_1, v_2, \ldots, v_k)$ consists of one element from each $V_{0,i}$.

This matches the definition of a k-Uniform Multipartite Hypergraph, where the vertex set is partitioned into k classes, and each hyperedge consists of one vertex from each class.

Therefore, the k-Uniform Multipartite n-SuperHypergraph with $n = 1$ is equivalent to a k-Uniform Multipartite Hypergraph. □

3.9 SuperHypergraphic Sequence

In this subsection, we explore the concept of a SuperHypergraphic Sequence. In mathematics, a sequence is an ordered list of elements, typically numbers, following a specific rule [129, 205]. The *degree sequence* of a graph or hypergraph is defined as the list of vertex degrees, where each degree represents the number of edges incident to the corresponding vertex [37, 63, 237, 238, 241]. A *hypergraphic sequence* is a sequence of non-negative integers that satisfies specific combinatorial conditions, ensuring the existence of a corresponding hypergraph [218, 218, 239, 283]. We extend these notions to n-SuperHyperGraphs. The related definitions and theorems are presented below.

Definition 3.62 (Degree (Recall))**.** [283] The *degree* of a vertex $v \in V$, denoted as $d(v)$, is the number of hyperedges in E that contain v, formally defined as:

$$d(v) = |\{e \in E \mid v \in e\}|.$$

Definition 3.63. [283] A hypergraph H is called *simple* if it contains no repeated hyperedges. Moreover, if every hyperedge in E contains exactly r vertices, the hypergraph is called an *r-uniform hypergraph.*

Definition 3.64. [283] The *degree sequence* of a hypergraph H is the vector of degrees of all vertices, represented as:

$$d(H) = (d(v_1), d(v_2), \ldots, d(v_n)).$$

Given an n-dimensional integer vector $d = (d_1, d_2, \ldots, d_n)$, it is said to be a *hypergraphic sequence* if there exists a simple hypergraph H with $d(H) = d$.

Definition 3.65 (Degree of a Supervertex)**.** In an n-SuperHyperGraph $H = (V, E)$, the *degree* of a supervertex $v \in V$, denoted $d(v)$, is defined as the number of superedges in E that contain v:

$$d(v) = |\{e \in E \mid v \in e\}|.$$

Definition 3.66 (n-SuperHypergraphic Sequence)**.** Given a finite set V of supervertices, an m-tuple of non-negative integers $d = (d(v_1), d(v_2), \ldots, d(v_m))$ is called an *n-SuperHypergraphic Sequence* if there exists an n-SuperHyperGraph $H = (V, E)$ such that for each supervertex $v_i \in V$, the degree $d(v_i)$ equals the given degree in the sequence, i.e.,

$$d(v_i) = |\{e \in E \mid v_i \in e\}| \quad \text{for } i = 1, 2, \ldots, m.$$

Theorem 3.67. *An n-SuperHypergraphic Sequence generalizes the concept of a hypergraphic sequence. Specifically, when $n = 0$, the n-SuperHypergraphic Sequence reduces to a hypergraphic sequence.*

Proof. Case 1 ($n = 0$): When $n = 0$, the n-th iterated power set is $\mathcal{P}^0(V_0) = V_0$. Thus, the supervertices are the base vertices $V = V_0$, and the superedges are subsets of V_0, i.e., $E \subseteq \mathcal{P}(V_0)$.

An n-SuperHyperGraph $H = (V, E)$ becomes a standard hypergraph in this case. The degree of each vertex $v \in V$ is calculated as:

$$d(v) = |\{e \in E \mid v \in e\}|,$$

which matches the definition of vertex degrees in hypergraphs.

Therefore, the degree sequence $d = (d(v_1), d(v_2), \ldots, d(v_m))$ is a hypergraphic sequence.

Case 2 ($n > 0$): For $n > 0$, the supervertices $V \subseteq \mathcal{P}^n(V_0)$ include higher-order elements from the iterated power set. The degrees of supervertices are defined similarly:

$$d(v) = |\{e \in E \mid v \in e\}| \quad \text{for all } v \in V.$$

This extends the concept of a degree sequence to n-SuperHyperGraphs, capturing the degrees of supervertices at various hierarchical levels.

Since the definition of an n-SuperHypergraphic Sequence encompasses the standard hypergraphic sequence when $n = 0$, and generalizes it for $n > 0$, it follows that the n-SuperHypergraphic Sequence is a generalization of the hypergraphic sequence. □

3.10 Query n-superhypergraph

A *Query Hypergraph* is a mathematical structure utilized in information retrieval to represent relationships between query concepts [29, 322, 370]. This concept is extended to n-SuperHyperGraphs, resulting in the definition of a *Query n-SuperHyperGraph*. The related definitions and theorems are provided below.

Definition 3.68. [29] A Query Hypergraph $H = (V, E, \varphi)$ is defined as follows:

- *Vertices (V)*: The vertex set $V = Q \cup \{D\}$, where:
  - Q is the set of query concepts, which may include terms, phrases, or other linguistic structures derived from a query Q.
  - D represents a document in the retrieval corpus.
- *Hyperedges (E)*: A hyperedge $e \in E$ connects a subset of query concepts $k \subseteq Q$ with the document D. Formally:

$$e = (k, D), \quad k \subseteq Q.$$

- *Weights (φ)*: Each hyperedge $e = (k, D)$ is associated with a weight $\varphi(e)$, which represents the relevance or importance of the relationship between the query concept set k and the document D.

Definition 3.69 (Query n-SuperHyperGraph)**.** Let V_0 be the base set of query concepts derived from a query Q, and let D represent a document in the retrieval corpus. Define the n-th iterated power set of V_0 recursively as:

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

The *Query n-SuperHyperGraph* $H = (V, E, \varphi)$ is defined as follows:

- *Vertices (V)*: The vertex set V consists of supervertices, which are elements of the n-th iterated power set of the base set V_0 augmented with the document D:

$$V = \mathcal{P}^n(V_0) \cup \{D\}.$$

- *Superedges (E)*: The superedge set E consists of subsets of V, connecting supervertices at various hierarchical levels. Each superedge $e \in E$ is defined as:

$$e = (k, D), \quad k \in \mathcal{P}^n(V_0).$$

- *Weights (φ)*: Each superedge $e = (k, D)$ is associated with a weight $\varphi(e)$, representing the relevance or importance of the relationship between the supervertex set k and the document D.

Theorem 3.70. *The Query n-SuperHyperGraph generalizes the Query Hypergraph.*

Proof. When $n = 0$, the n-th iterated power set reduces to $\mathcal{P}^0(V_0) = V_0$, the base set of query concepts. In this case:

- The vertices V become $V = V_0 \cup \{D\}$, matching the vertex set in the Query Hypergraph.
- The superedges E are defined as $e = (k, D)$ with $k \in \mathcal{P}^0(V_0) = V_0$, so $k \subseteq V_0$. This matches the hyperedges in the Query Hypergraph, which connect subsets of query concepts $k \subseteq Q$ with the document D.
- The weights $\varphi(e)$ remain unchanged.

Therefore, the Query n-SuperHyperGraph reduces to the Query Hypergraph when $n = 0$. For $n > 0$, it extends the structure to include higher-level supervertices and superedges, thus generalizing the Query Hypergraph. □

Theorem 3.71. *A Query n-SuperHyperGraph possesses the structure of an n-SuperHyperGraph.*

Proof. By definition, an n-SuperHyperGraph $H = (V, E)$ consists of:

- *Vertices (V)*: Elements of the n-th iterated power set $\mathcal{P}^n(V_0)$, where V_0 is the base set.
- *Edges (E)*: Subsets of $\mathcal{P}^n(V_0)$, connecting supervertices at different hierarchical levels.

For a Query n-SuperHyperGraph $H = (V, E, \varphi)$, we have:

- *Vertices (V)*: Defined as $\mathcal{P}^n(V_0) \cup \{D\}$, where V_0 is the set of query concepts and D is the document. The additional element D does not alter the hierarchical structure of $\mathcal{P}^n(V_0)$, as it can be treated as a singleton set $\{D\} \subseteq \mathcal{P}^n(V_0)$.
- *Superedges (E)*: Defined as $e = (k, D)$ for $k \in \mathcal{P}^n(V_0)$. These superedges are subsets of V and connect elements within $\mathcal{P}^n(V_0) \cup \{D\}$, preserving the hierarchical structure of $\mathcal{P}^n(V_0)$.

The weights $\varphi(e)$ do not affect the structural composition of the vertices and superedges, as they are additional metadata associated with each superedge.

Thus, the Query n-SuperHyperGraph $H = (V, E, \varphi)$ satisfies the structural requirements of an n-SuperHyperGraph $H' = (V', E')$, with:

$$V' = \mathcal{P}^n(V_0), \quad E' = \mathcal{P}^n(V_0).$$

Therefore, a Query n-SuperHyperGraph possesses the structure of an n-SuperHyperGraph. □

3.11 Superhypergraph Energy Functions

Hypergraph Energy Functions are mathematical tools designed to quantify relationships in hypergraphs by optimizing node and edge embeddings for downstream tasks [61, 341]. This concept is extended to superhypergraphs, and the corresponding definitions are provided below.

Definition 3.72 (Hyperedge Regularization)**.** (cf. [338, 341]) Hyperedge Regularization is a technique that enforces similarity or consistency among nodes within the same hyperedge in a hypergraph. Mathematically, for a hypergraph $H = (V, E)$, the regularization term for a hyperedge $e \in E$ is often defined as:

$$R(e) = \sum_{i,j \in e} \|\mathbf{y}_i - \mathbf{y}_j\|^2,$$

where $\mathbf{y}_i$ and $\mathbf{y}_j$ are embeddings of nodes i and j, and $\|\cdot\|$ denotes the norm. This term penalizes differences in embeddings among nodes within the hyperedge e, promoting structural coherence.

Definition 3.73 (Hypergraph Energy Function)**.** [341] Let $H = (V, E)$ be a hypergraph, where V is the set of nodes, E is the set of hyperedges, and $B \in \mathbb{R}^{|V| \times |E|}$ is the binary incidence matrix such that $B_{ik} = 1$ if node $v_i \in e_k$, and $B_{ik} = 0$ otherwise. Define:

- $Y \in \mathbb{R}^{|V| \times d}$: Node embeddings where each row y_i represents the embedding of node v_i.
- $Z \in \mathbb{R}^{|E| \times d}$: Hyperedge embeddings where each row z_k represents the embedding of hyperedge e_k.
- $g_1(Y)$: A node regularization term ensuring smoothness or specific properties of Y.
- $g_2(Z)$: A hyperedge regularization term ensuring smoothness or specific properties of Z.
- $g_3(Y, Z)$: A structural term that encodes the relationships between nodes and hyperedges in the hypergraph.

The *hypergraph energy function* is defined as:

$$\mathcal{L}(Y, Z) = g_1(Y) + g_2(Z) + g_3(Y, Z),$$

where $g_3(Y, Z)$ can take the form:

$$g_3(Y, Z) = \lambda_0 \sum_{e_k \in E} \sum_{v_i, v_j \in e_k} \|y_i - y_j\|^2 + \lambda_1 \sum_{e_k \in E} \sum_{v_i \in e_k} \|y_i - z_k\|^2.$$

Here, λ_0 and λ_1 are weighting factors that balance the contributions of the terms.

Definition 3.74 (n-SuperHypergraph Energy Function)**.** Let $H = (V, E)$ be an n-SuperHyperGraph, where V is the set of supervertices and E is the set of superedges. Let d be the dimensionality of the embeddings.

- For each supervertex $v \in V$, let $y_v \in \mathbb{R}^d$ be its embedding.
- For each superedge $e \in E$, let $z_e \in \mathbb{R}^d$ be its embedding.
- Let $x_v \in \mathbb{R}^{d_v}$ be the feature vector associated with supervertex v.
- Let $u_e \in \mathbb{R}^{d_e}$ be the feature vector associated with superedge e.
- Let $f_v(\cdot; W_v)$ and $f_e(\cdot; W_e)$ be learnable functions (e.g., neural networks) parameterized by weights W_v and W_e, mapping features to embeddings in $\mathbb{R}^d$.

The *n-SuperHypergraph Energy Function* is defined as:

$$\mathcal{L}(Y, Z) = \sum_{v \in V} \|y_v - f_v(x_v; W_v)\|^2 + \sum_{e \in E} \|z_e - f_e(u_e; W_e)\|^2 + \lambda_0 \sum_{e \in E} \sum_{v, w \in e} \|y_v - y_w\|^2 + \lambda_1 \sum_{e \in E} \sum_{v \in e} \|y_v - z_e\|^2,$$

where λ_0 and λ_1 are non-negative hyperparameters controlling the importance of each term.

Theorem 3.75. *The n-SuperHypergraph Energy Function generalizes the hypergraph energy function. Specifically, when $n = 0$, the n-SuperHypergraph Energy Function reduces to the standard hypergraph energy function.*

Proof. When $n = 0$, the n-th iterated power set reduces to $\mathcal{P}^0(V_0) = V_0$. Thus, the supervertices and superedges become elements and subsets of the base vertex set V_0, respectively.

In this case:

- The set of supervertices $V \subseteq V_0$ is simply the set of vertices in the hypergraph.
- The set of superedges $E \subseteq \mathcal{P}(V_0)$ is the set of hyperedges in the hypergraph.
- The embeddings y_v for $v \in V$ correspond to the node embeddings in the hypergraph.
- The embeddings z_e for $e \in E$ correspond to the hyperedge embeddings in the hypergraph.

The incidence matrix $B \in \mathbb{R}^{|V| \times |E|}$ is defined as:

$$B_{v,e} = \begin{cases} 1, & \text{if } v \in e, \\ 0, & \text{otherwise.} \end{cases}$$

The energy function simplifies to:

$$\mathcal{L}(Y, Z) = \sum_{v \in V} \|y_v - f_v(x_v; W_v)\|^2 + \sum_{e \in E} \|z_e - f_e(u_e; W_e)\|^2 + \lambda_0 \sum_{e \in E} \sum_{v,w \in e} \|y_v - y_w\|^2 + \lambda_1 \sum_{e \in E} \sum_{v \in e} \|y_v - z_e\|^2,$$

which is exactly the standard hypergraph energy function.

Therefore, the n-SuperHypergraph Energy Function generalizes the hypergraph energy function. □

3.12 Transversal n-SuperHypergraph

A transversal graph is a type of graph where every edge intersects all subsets of edges, ensuring that no subset remains disjoint from the edge set [31, 67, 147, 240].

Similarly, a transversal hypergraph is defined as a hypergraph where every hyperedge represents a minimal hitting set that intersects all hyperedges of the original hypergraph [92, 93, 146, 148, 191, 323, 333].

This concept is extended to the domain of n-SuperHyperGraphs. The related definitions and theorems are provided below.

Definition 3.76 (Transversal)**.** (cf. [92, 93, 148, 191, 323]) Let $H = (V, E)$ be a hypergraph. A set $T \subseteq V$ is called a *transversal* (or hitting set) of H if:

$$T \cap E_i \neq \emptyset, \quad \forall E_i \in E.$$

A transversal T is *minimal* if no proper subset $T' \subset T$ is a transversal of H.

Definition 3.77 (Transversal Hypergraph)**.** (cf. [92, 93, 148, 191, 323]) Let $H = (V, E)$ be a hypergraph. The *transversal hypergraph* of H, denoted $\mathrm{Tr}(H)$, is defined as the hypergraph:

$$\mathrm{Tr}(H) = (V, \mathcal{T}),$$

where $\mathcal{T}$ is the family of all minimal transversals of H.

Definition 3.78 (Base Set)**.** For any element $x \in V \cup E$ of an n-SuperHyperGraph $H = (V, E)$, the *base set* of x, denoted $\mathrm{Base}(x)$, is defined recursively as:

- If $x \in V_0$, then $\mathrm{Base}(x) = \{x\}$.
- If x is a set, i.e., $x \in \mathcal{P}^k(V_0)$ for $k \geq 1$, then:

$$\mathrm{Base}(x) = \bigcup_{y \in x} \mathrm{Base}(y).$$

Definition 3.79 (Incidence in n-SuperHyperGraph)**.** In an n-SuperHyperGraph $H = (V, E)$, a supervertex $v \in V$ and a superedge $e \in E$ are said to be *incident* if:

$$\mathrm{Base}(v) \cap \mathrm{Base}(e) \neq \emptyset.$$

Definition 3.80 (Transversal in n-SuperHyperGraph)**.** A set $T \subseteq V$ is called a *transversal* (or *hitting set*) of an n-SuperHyperGraph $H = (V, E)$ if for every superedge $e \in E$, there exists a supervertex $v \in T$ such that v is incident to e; that is:

$$\mathrm{Base}(v) \cap \mathrm{Base}(e) \neq \emptyset.$$

A transversal T is *minimal* if no proper subset $T' \subset T$ is a transversal of H.

Definition 3.81 (Transversal n-SuperHyperGraph)**.** Given an n-SuperHyperGraph $H = (V, E)$, the *Transversal n-SuperHyperGraph* of H, denoted $\mathrm{Tr}(H)$, is defined as:

$$\mathrm{Tr}(H) = (V, \mathcal{T}),$$

where $\mathcal{T}$ is the set of all minimal transversals of H.

Theorem 3.82. *The Transversal n-SuperHyperGraph generalizes the Transversal Hypergraph. Specifically, when $n = 0$, the Transversal n-SuperHyperGraph reduces to the classical Transversal Hypergraph.*

Proof. When $n = 0$, the n-SuperHyperGraph $H = (V, E)$ becomes a standard hypergraph:

- The 0-th iterated power set is $\mathcal{P}^0(V_0) = V_0$.
- The supervertices $V \subseteq V_0$ are simply the vertices of the hypergraph.
- The superedges $E \subseteq \mathcal{P}^0(V_0) = V_0$ become subsets of V_0, i.e., hyperedges.

The base set of any vertex $v \in V$ is:

$$\mathrm{Base}(v) = \{v\}, \quad \text{since } v \in V_0.$$

The base set of any edge $e \in E$ is:

$$\mathrm{Base}(e) = \bigcup_{u \in e} \mathrm{Base}(u) = e.$$

The incidence relation simplifies to:

$$\mathrm{Base}(v) \cap \mathrm{Base}(e) = \{v\} \cap e \neq \emptyset \iff v \in e.$$

Therefore, a transversal $T \subseteq V$ satisfies:

$$T \cap e \neq \emptyset, \quad \forall e \in E,$$

which is the classical definition of a transversal (hitting set) in a hypergraph.

The minimal transversals in H correspond to the minimal hitting sets in the hypergraph. Consequently, the Transversal n-SuperHyperGraph $\mathrm{Tr}(H) = (V, \mathcal{T})$ reduces to the classical Transversal Hypergraph, where $\mathcal{T}$ is the set of all minimal transversals.

Thus, the Transversal n-SuperHyperGraph generalizes the Transversal Hypergraph. □

Theorem 3.83. *A Transversal n-SuperHyperGraph possesses the structural properties of an n-SuperHyperGraph.*

Proof. Let $H = (V, E)$ be an n-SuperHyperGraph, where $V \subseteq \mathcal{P}^n(V_0)$ is the set of n-level supervertices, and $E \subseteq \mathcal{P}^n(V_0)$ is the set of n-level superedges.

By definition, the Transversal n-SuperHyperGraph $\mathrm{Tr}(H) = (V, \mathcal{T})$ is formed by computing the family $\mathcal{T}$, which consists of all minimal transversals of H.

The vertices V of $\mathrm{Tr}(H)$ are identical to those of the original n-SuperHyperGraph H, and thus $V \subseteq \mathcal{P}^n(V_0)$.

Each edge $T \in \mathcal{T}$ is a minimal transversal of H. A transversal $T \subseteq V$ ensures that $T \cap e \neq \emptyset$ for every $e \in E$. Since $T \subseteq V$ and $V \subseteq \mathcal{P}^n(V_0)$, we have $T \subseteq \mathcal{P}^n(V_0)$. Hence, $\mathcal{T} \subseteq \mathcal{P}^n(V_0)$.

The set $\mathcal{T}$ is a subset of the n-th iterated power set $\mathcal{P}^n(V_0)$, which aligns with the edge definition of an n-SuperHyperGraph. Therefore, $\mathrm{Tr}(H)$ adheres to the structural constraints of an n-SuperHyperGraph.

Thus, $\mathrm{Tr}(H)$ satisfies the vertex and edge definitions of an n-SuperHyperGraph, confirming that it retains the structural properties of n-SuperHyperGraphs. □

3.13 n-SuperHypernetwork

A hypernetwork is a related concept to hypergraphs, employing similar principles to represent relationships in networks [14, 16, 163, 264]. Extensive research has been conducted in this area. This concept is extended to n-SuperHypernetworks, which provide a more general and hierarchical framework. Relevant definitions and theorems are detailed below.

Definition 3.84 (Hypernetwork)**.** [14,16,163] A *hypernetwork* is defined as a hypergraph $G = (V, E)$ equipped with a node type mapping function $\varphi : V \to A$, where:

- V is the set of nodes,
- E is the set of hyperedges, where each $e \in E$ is a non-empty subset of V,
- A is the set of node types,
- $\varphi(v) \in A$ specifies the type of each node $v \in V$.

A hyperedge $e \in E$ represents a *tuplewise relationship* among the nodes in e. The following additional properties can be used to classify hypernetworks:

1. *Homogeneous vs. Heterogeneous Hypernetwork*:
   - The hypernetwork is *homogeneous* if $|A| = 1$, i.e., all nodes are of the same type.
   - The hypernetwork is *heterogeneous* if $|A| > 1$, i.e., nodes can belong to multiple types [162, 339, 369].
2. *Uniformity*:
   - The hypernetwork is k*-uniform* if every hyperedge $e \in E$ satisfies $|e| = k$, i.e., all hyperedges contain exactly k nodes.

The *neighbors* of a node $v \in V$ are defined as:

$$N_G(v) = \{u \in V \mid \exists e \in E \text{ such that } v \in e \text{ and } u \in e\}.$$

Definition 3.85 (Hypernetwork Representation)**.** [14, 16, 163] Given a hypernetwork $G = (V, E)$, the goal of *hypernetwork representation learning* is to learn:

1. A *node embedding function* $f : V \to \mathbb{R}^d$, which maps each node $v \in V$ to a low-dimensional vector $f(v) \in \mathbb{R}^d$ (cf. [59, 297]),

2. A *tuplewise similarity function* $s_{\text{tuple}} : T \to [0, 1]$, where T is the set of possible tuples of nodes in V, to measure the relationships among nodes in tuples (cf. [330, 381]).

The representation $f(v)$ should preserve both global and local structural information of the hypernetwork, including:

- *Pairwise relationships*, reflecting the similarity between two nodes(cf. [66, 350]),
- *Tuplewise relationships*, capturing the interactions among more than two nodes within a hyperedge.

Definition 3.86 (n-SuperHypernetwork)**.** Let V_0 be a finite set of base nodes. The n-th iterated power set of V_0 is defined recursively as:

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

where $\mathcal{P}(A)$ denotes the power set of set A.

An *n-SuperHypernetwork* is an ordered triple $H = (V, E, \varphi)$, where:

- $V \subseteq \mathcal{P}^n(V_0)$ is the set of *supernodes*.
- $E \subseteq \mathcal{P}^n(V_0)$ is the set of *superedges*.
- $\varphi : V \to A$ is a *node type mapping function*, with A being the set of node types.

Each supernode $v \in V$ can be:

- A single node ($v \in V_0$),
- A subset of V_0 ($v \subseteq V_0$),
- A subset of subsets of V_0, up to n levels ($v \in \mathcal{P}^n(V_0)$).

Similarly, each superedge $e \in E$ connects supernodes, potentially at different hierarchical levels up to n.

Definition 3.87 (n-SuperHypernetwork Representation)**.** Given an n-SuperHypernetwork $H = (V, E, \varphi)$, the goal of *n-SuperHypernetwork representation learning* is to learn:

1. A *node embedding function* $f : V \to \mathbb{R}^d$, which maps each supernode $v \in V$ to a low-dimensional vector $f(v) \in \mathbb{R}^d$.
2. A *tuplewise similarity function* $s_{\text{tuple}} : T \to [0, 1]$, where T is the set of possible tuples (e.g., superedges) in V, to measure the relationships among nodes in tuples.

The representations aim to preserve both global and local structural information of the n-SuperHypernetwork, including:

- *Pairwise relationships*, reflecting similarities between supernodes.
- *Tuplewise relationships*, capturing interactions among multiple supernodes within superedges.

Theorem 3.88. *When $n = 0$, the n-SuperHypernetwork reduces to a hypernetwork, and the n-SuperHypernetwork representation reduces to the hypernetwork representation. Therefore, the definitions of n-SuperHypernetwork and its representation generalize those of hypergraphs and hypernetworks.*

Proof. Consider $n = 0$. Then, the 0-th iterated power set is:

$$\mathcal{P}^0(V_0) = V_0.$$

Thus, the set of supernodes and superedges become:

$$V \subseteq \mathcal{P}^0(V_0) = V_0, \quad E \subseteq \mathcal{P}^0(V_0) = V_0.$$

This means:

- The supernodes V are simply elements of V_0, i.e., the base nodes themselves.
- The superedges E are subsets of V_0. Since $E \subseteq V_0$, each edge $e \in E$ is a node in V_0, which does not align with the standard hyperedge definition. This suggests that we should consider $n = 1$ for a meaningful hyperedge structure.

Now, consider $n = 1$:

$$\mathcal{P}^1(V_0) = \mathcal{P}(V_0),$$

the standard power set of V_0.

Then:

$$V \subseteq \mathcal{P}^1(V_0) = \mathcal{P}(V_0), \quad E \subseteq \mathcal{P}^1(V_0) = \mathcal{P}(V_0).$$

In this case:

- The supernodes V are subsets of V_0, i.e., sets of nodes.
- The superedges E are subsets of V_0, i.e., hyperedges in the classical sense.

If we restrict $V = V_0$, then the supernodes are the base nodes themselves, and the superedges $E \subseteq \mathcal{P}(V_0)$ are standard hyperedges connecting nodes in V_0.

Thus, the n-SuperHypernetwork $H = (V, E, \varphi)$ with $n = 1$ reduces to a traditional hypernetwork, where:

- $V \subseteq V_0$ is the set of nodes.
- $E \subseteq \mathcal{P}(V_0)$ is the set of hyperedges.
- $\varphi : V \to A$ maps nodes to their types.

Regarding the representation, the n-SuperHypernetwork representation learning aims to learn embeddings $f : V \to \mathbb{R}^d$ and a tuplewise similarity function s_{tuple}. When $n = 1$, this reduces to learning node embeddings and similarity functions for hypernetworks, as commonly done in hypernetwork representation learning.

Therefore, the definitions of n-SuperHypernetwork and n-SuperHypernetwork representation generalize the classical definitions of hypergraphs and hypernetworks. □

Question 3.89. Can the relationships between the aforementioned network concepts and Graph Neural Networks [156–158, 288, 294, 351, 352, 369, 377, 378], Hypergraph Neural Networks [188, 210, 212, 214, 219, 341, 349, 357, 379], and Superhypergraph Neural Networks [118] be formalized into theorems and proven? Additionally, is it possible to combine them to develop some form of practical applications?

3.14 Introduction to Other Known Superhypergraph Classes

Several other classes of superhypergraphs are already known. To facilitate the future development of research in superhypergraphs, we present the definitions of these classes below for reference. These can be seen as extensions of analogous concepts in hypergraphs.

3.14.1 Directed Superhypergraph and Bidirected Superhypergraph

A Directed Graph is a graph in which orientations are assigned to edges in a standard graph [5, 130]. Similarly, in the context of hypergraphs, Directed Hypergraphs are well-studied structures [123]. A Directed Superhypergraph is an extension of this concept, assigning orientations to the edges of a Superhypergraph. The formal definition is provided below [120].

Definition 3.90 (Directed n-SuperHyperGraph)**.** [120] A *Directed n-SuperHyperGraph* is defined as a tuple:

$$H = (V, E),$$

where:

- $V \subseteq \mathcal{P}^n(V_0)$ is the set of *supervertices*, where V_0 is a finite set of base vertices and $\mathcal{P}^n(V_0)$ represents the n-th iterated power set of V_0.
- $E \subseteq \{(T, H) \mid T, H \subseteq V\}$ is the set of *directed superhyperedges*, where each $e = (T, H)$ satisfies:
  - $T \subseteq V$: the *tail set*, representing source supervertices.
  - $H \subseteq V$: the *head set*, representing target supervertices.

A directed superhyperedge $e = (T, H)$ generalizes the concept of edges in directed graphs and hypergraphs, allowing connections between multiple source and target supervertices.

Question 3.91. Can the superhypergraph classes introduced in this paper be extended to Directed Superhypergraphs? Furthermore, what potential mathematical structures and applications could arise from such an extension?

A mixed graph combines undirected and directed edges, enabling both two-way and one-way connections between vertices [108, 282]. This framework has been further generalized to mixed hypergraphs [326], which adapt the concept to hypergraphs, with their mathematical characteristics studied extensively.

Definition 3.92 (Mixed n-SuperHyperGraph)**.** [120] A *Mixed n-SuperHyperGraph* is defined as a tuple:

$$H = (V, S, E, A),$$

where:

- $V \subseteq \mathcal{P}^n(V_0)$ is the set of *supervertices*.
- $S \subseteq \mathcal{P}^n(V_0)$ is the set of subsets of supervertices, called *supervertex sets*.
- $E \subseteq \mathcal{P}(S)$ is the set of undirected *superedges*.
- $A \subseteq \{(Z, z) \mid Z \subseteq S, z \in S, Z \cap \{z\} = \emptyset\}$ is the set of directed *superedges*, where each directed superedge $a = (Z, z)$ consists of:
  - Z: the *tail set*, a non-empty subset of supervertex sets.
  - z: the *head*, a supervertex set.

Mixed superhypergraphs combine undirected and directed edges, allowing flexible representation of both directional and non-directional relationships.

The idea of a bidirected graph [15, 91, 130] has gained attention in recent years. To expand on this, we outline the definitions of bidirected hypergraphs and bidirected superhypergraphs, which extend the principles of bidirected graphs.

Definition 3.93 (Bidirected n-SuperHyperGraph)**.** [120] A *Bidirected n-SuperHyperGraph* is defined as a triple:

$$H = (V, E, \tau),$$

where:

- $V \subseteq \mathcal{P}^n(V_0)$ is the set of *supervertices*.
- $E \subseteq \mathcal{P}(V)$ is the set of *superedges*.
- $\tau : V \times E \to \{-1, 0, 1\}$ is the *bidirection function*, assigning orientations to the incidence of supervertices and superedges:
  - $\tau(v, e) = 1$: Superedge e is directed *toward* supervertex v.
  - $\tau(v, e) = -1$: Superedge e is directed *away from* supervertex v.
  - $\tau(v, e) = 0$: Supervertex v is not incident to superedge e.

This structure allows independent orientations for each supervertex with respect to each incident superedge, generalizing the concept of bidirectionality in graphs.

3.14.2 Multi-Superhypergraph and Pseudo-Superhypergraph

A notable type of graph is the multigraph, characterized by its allowance for multiple edges (often called parallel edges) connecting the same pair of vertices [62, 102, 204, 234]. This concept is further extended to hypergraphs, resulting in the multi-hypergraph, which permits the existence of parallel hyperedges. Both multigraphs and multi-hypergraphs are widely utilized across various fields, including the study of neural networks [32,207,213,259,321,345]. This idea has been extended to superhypergraphs, leading to the concept of the *multi-superhypergraph*, which was defined in [120]. The formal definition is provided below.

Definition 3.94 (Multi-n-SuperHyperGraph)**.** [120] A *Multi-n-SuperHyperGraph* is defined as a triple:

$$H = (V, S, E),$$

where:

- $V \subseteq \mathcal{P}^n(V_0)$ is a finite set of *supervertices*, where V_0 is a finite set of base vertices and $\mathcal{P}^n(V_0)$ represents the n-th iterated power set of V_0.
- S is a multiset of non-empty subsets of V, called *multi-supervertices*. Each multi-supervertex $s \in S$ satisfies $s \subseteq V$, and multiple occurrences of the same subset s are permitted in S.
- E is a multiset of non-empty subsets of S, called *multi-superedges*. Each multi-superedge $e \in E$ satisfies $e \subseteq S$, and multiple occurrences of the same subset e are permitted in E.

This structure extends the n-SuperHyperGraph by allowing repeated subsets within the sets of supervertices and superedges, enabling richer modeling of relationships and connections.

A pseudograph is a graph variant that permits both parallel edges and self-loops, where an edge connects a vertex to itself [24,48,204]. This flexibility allows for the depiction of more intricate relationships and complex network structures compared to traditional graph models [334,353]. By extending this concept to hypergraphs, a pseudo-hypergraph is introduced, enabling the representation of even more sophisticated connections and interactions [18, 51, 211]. Building on these advancements, the notion of a pseudo-superhypergraph, which generalizes the pseudo-hypergraph to superhypergraphs, has been defined in [120]. The formal definition is provided below.

Definition 3.95 (Pseudo-n-SuperHyperGraph)**.** [120] A *Pseudo-n-SuperHyperGraph* is defined as a triple:

$$H = (V, S, E),$$

where:

- $V \subseteq \mathcal{P}^n(V_0)$ is a finite set of *supervertices*, where V_0 is a finite set of base vertices and $\mathcal{P}^n(V_0)$ represents the n-th iterated power set of V_0.
- S is a multiset of elements from V, called *pseudo-supervertices*. Each pseudo-supervertex $s \in S$ is a multiset of supervertices from V, allowing:
  - Repetition of the same supervertex within a pseudo-supervertex s.
  - Repetition of the same pseudo-supervertex across S.
- E is a multiset of elements from S, called *pseudo-superedges*. Each pseudo-superedge $e \in E$ is a multiset of pseudo-supervertices from S, allowing:
  - Repetition of the same pseudo-supervertex within a pseudo-superedge e.
  - Repetition of the same pseudo-superedge across E.

This structure generalizes the n-SuperHyperGraph by incorporating multisets, enabling repeated elements at multiple levels of the hierarchy.

3.14.3 Dynamic Superhypergraph

In fields such as Neural Networks, dynamic graph concepts like Dynamic Graphs [26,27,192,340] and Dynamic Hypergraphs [189,332,356,375] are well-known. Extending these concepts to superhypergraphs, the Dynamic Superhypergraph has also been introduced [118]. The definition is presented below.

Definition 3.96. [118] A *Dynamic SuperHypergraph* is a sequence of n-SuperHyperGraphs $\{H^{(l)} = (V^{(l)}, E^{(l)})\}_{l=0}^{L}$, where each layer l represents a SuperHyperGraph at a specific time or iteration, and:

- $V^{(l)} \subseteq \mathcal{P}^n(V_0)$ is the set of supervertices at layer l, where V_0 is the base set of vertices, and $\mathcal{P}^n(V_0)$ is the n-th iterated power set of V_0.
- $E^{(l)} \subseteq \mathcal{P}^n(V_0)$ is the set of superedges at layer l.

The evolution of the SuperHyperGraph from layer l to $l + 1$ may depend on the features or embeddings of the supervertices at layer l.

Question 3.97. Inspired by the concept of HyperStorylines in Dynamic Hypergraphs, is it possible to explore the application of SuperHyperStorylines within Dynamic Superhypergraphs?

3.14.4 Quasi superhypergraph

A Quasi-SuperHyperGraph is a graph that is almost a Quasi-SuperHyperGraph [150]. The formal definition is provided below.

Definition 3.98 (Quasi-n-SuperHyperGraph)**.** [150] A *Quasi-n-SuperHyperGraph* is a triple:

$$H = (V, S, \Phi),$$

where:

- $V \subseteq \mathcal{P}^n(V_0)$ is a set of *supervertices*, where V_0 is a finite base set, and $\mathcal{P}^n(V_0)$ represents its n-th iterated power set.
- $S = \{S_i\}_{i=1}^{k} \subseteq \mathcal{P}(V)$ is a family of subsets of V, called *super-supervertices*.
- $\Phi = \{\varphi_{i,j} \mid i \neq j\}$ is a set of mappings $\varphi_{i,j} : S_i \to S_j$, called *quasi-superedges*, representing directed connections between super-supervertices.

3.14.5 Superhypertree

A Superhypertree is the tree version of a Superhypergraph. In recent years, the graph width parameter known as Superhypertree-width has also been defined and studied. The formal definition is provided below [112].

Definition 3.99 (n-SuperHyperTree)**.** [112, 131] An *n-SuperHyperTree* is an n-SuperHyperGraph SHT $= (V, E)$ satisfying the following conditions:

1. *Host Tree Condition*: There exists a tree $T = (V_T, E_T)$, called the *host tree*, such that:
   - The vertex set of T is $V_T = V$.
   - Each superedge $e \in E$ corresponds to a connected subtree of T.
2. *Acyclicity Condition*: The host tree T is acyclic, ensuring that SHT does not contain cycles.
3. *Connectedness Condition*: For any $v, w \in V$, there exists a sequence of superedges $e_1, e_2, \ldots, e_k \in E$ such that:
$$v \in e_1, \quad w \in e_k, \quad \text{and} \quad e_i \cap e_{i+1} \neq \emptyset \quad \text{for } 1 \leq i < k.$$

3.15 General Plithogenic n-SuperHyperGraph

The concept of a Plithogenic Graph [106, 108, 187, 298, 299, 306, 325] serves as a generalization of various types of graphs, including Fuzzy Graphs [33, 125, 135, 190, 243, 252, 277, 280, 324, 344], Neutrosophic Graphs [11, 12, 57, 111, 113, 165, 186, 284], Vague Graphs [8, 9, 43–45, 271, 272, 286], Intuitionistic Fuzzy Graphs [6, 173, 331, 376], and Pentapartitioned Neutrosophic Graphs [79, 167, 168, 266]. It is particularly known for its flexibility in handling uncertainty by allowing a customizable number of parameters to represent various degrees of vagueness and ambiguity. The General Plithogenic Graph is an extended framework that relaxes the constraints of a Plithogenic Graph, thereby offering a more versatile graph structure [111, 250]. The General Plithogenic n-SuperHyperGraph is a further extension, applying the principles of the General Plithogenic Graph to the domain of SuperHyperGraphs, thus combining the hierarchical structure of n-SuperHyperGraphs with the flexibility of Plithogenic Graphs [107, 309].

Definition 3.100 (General Plithogenic n-SuperHyperGraph)**.** [107] Let V_0 be a finite set of base vertices. Define the n-th iterated power set of V_0 recursively as:

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

where $\mathcal{P}(A)$ denotes the power set of the set A.

A *General Plithogenic n-SuperHyperGraph* is an octuple:

$$H^{(n)GP} = (V, E, A_V, A_E, \mathrm{DAF}_V, \mathrm{DAF}_E, \mathrm{DCF}_V, \mathrm{DCF}_E),$$

with the following conditions:

- $V \subseteq \mathcal{P}^n(V_0)$ is the set of *supervertices*, where each supervertex is an element of the n-th iterated power set of V_0. Thus, a supervertex can be:
  - A single vertex $v \in V_0$,
  - A subset of V_0,
  - A subset of subsets of V_0, up to n levels, i.e., $v \in \mathcal{P}^n(V_0)$,
  - An indeterminate or fuzzy set (cf. [360]),
  - The null set $\emptyset$.
- $E \subseteq \mathcal{P}^n(V_0)$ is the set of *superedges*, where each superedge is also an element of $\mathcal{P}^n(V_0)$. Each superedge connects supervertices potentially at multiple hierarchical levels up to n.
- A_V is a finite set of attributes associated with the supervertices.

- A_E is a finite set of attributes associated with the superedges.
- $\mathrm{DAF}_V : V \times A_V \to [0,1]^s$ is the *Degree of Appurtenance Function* for supervertices, assigning to each pair (v, a_V), with $v \in V$ and $a_V \in A_V$, a membership degree in $[0,1]^s$.
- $\mathrm{DAF}_E : E \times A_E \to [0,1]^s$ is the *Degree of Appurtenance Function* for superedges, assigning to each pair (e, a_E), with $e \in E$ and $a_E \in A_E$, a membership degree in $[0,1]^s$.
- $\mathrm{DCF}_V : A_V \times A_V \to [0,1]^t$ is the *Degree of Contradiction Function* for vertex attributes, satisfying:

$$\mathrm{DCF}_V(a,a) = 0, \quad \mathrm{DCF}_V(a,b) = \mathrm{DCF}_V(b,a), \quad \forall a, b \in A_V.$$

- $\mathrm{DCF}_E : A_E \times A_E \to [0,1]^t$ is the *Degree of Contradiction Function* for edge attributes, satisfying:

$$\mathrm{DCF}_E(a,a) = 0, \quad \mathrm{DCF}_E(a,b) = \mathrm{DCF}_E(b,a), \quad \forall a, b \in A_E.$$

The degrees of appurtenance assigned by DAF_V and DAF_E may be adjusted or interpreted through the DCF_V and DCF_E functions, reflecting plithogenic synthesis of attributes, where multiple conditions (attributes) combine, potentially with contradictory influences, to determine the final membership degrees of supervertices and superedges.

Example 3.101. (cf. [111]) The following examples illustrate specific cases of General Plithogenic n-SuperHyperGraphs:

- When $s = t = 1$, the G^{PGSH} is called a *Plithogenic Fuzzy n-SuperHyperGraph*.
- When $s = 2, t = 1$, the G^{PGSH} is called a *Plithogenic Intuitionistic Fuzzy n-SuperHyperGraph*. Also the G^{PGSH} is called a *Plithogenic Vague n-SuperHyperGraph*.
- When $s = 3, t = 1$, the G^{PGSH} is called a *Plithogenic Neutrosophic n-SuperHyperGraph*.
- When $s = 4, t = 1$, the G^{PGSH} is called a *Plithogenic Quadripartitioned Neutrosophic n-SuperHyperGraph* (cf. [169, 269, 295]).
- When $s = 5, t = 1$, the G^{PGSH} is called a *Plithogenic Pentapartitioned Neutrosophic n-SuperHyperGraph* (cf. [35, 80, 229]).
- When $s = 6, t = 1$, the G^{PGSH} is called a *Plithogenic Hexapartitioned Neutrosophic n-SuperHyperGraph* (cf. [254]).
- When $s = 7, t = 1$, the G^{PGSH} is called a *Plithogenic Heptapartitioned Neutrosophic n-SuperHyperGraph* (cf. [56, 246]).
- When $s = 8, t = 1$, the G^{PGSH} is called a *Plithogenic Octapartitioned Neutrosophic n-SuperHyperGraph*.
- When $s = 9, t = 1$, the G^{PGSH} is called a *Plithogenic Nonapartitioned Neutrosophic n-SuperHyperGraph*.

4 Discussion: Generalized n-th Powerset (Power Mathematical structure)

This section briefly introduces the concept of the Generalized n-th Powerset. We believe that this structure can be applied not only in graph theory and set theory but also in other fields. It is our hope that further studies will explore its applications and implications. Relevant definitions and theorems are provided below.

Definition 4.1 (Generalized n-th Powerset)**.** Let H be a set or a mathematical structure, and let $P(H)$ denote the classical powerset of H. Define the n-th generalized powerset of H, denoted $G_n(H)$, recursively as:

$$G_1(H) = G(H),$$

$$G_{n+1}(H) = G\big(G_n(H)\big) \quad \text{for } n \geq 1,$$

where $G(H)$ is a generalized powerset operator that incorporates additional constraints, properties, or structures. Examples of $G(H)$ include:

- *Labeled subsets:* $G(H) = \{(A, \ell_A) \mid A \subseteq H, \ell_A \in L\}$, where L is a set of labels.
- *Weighted subsets [354]:* $G(H) = \{(A, w_A) \mid A \subseteq H, w_A \in \mathbb{R}\}$, where weights w_A are assigned to subsets.
- *Soft subsets [242]:* Let U be a universe and E a set of parameters. A soft subset over U is a pair (F, A), where $A \subseteq E$ and $F : A \to P(U)$. For each $e \in A$, $F(e) \subseteq U$ represents the set of elements satisfying parameter e.
- *Graph subsets:* $G(H) = \{(G, V_G, E_G) \mid V_G \subseteq V(H), E_G \subseteq E(H)\}$, where $G = (V_G, E_G)$ is a subgraph of H.
- *Structured subsets:* Subsets with internal structures, such as orderings, multisets, or graph-like properties.
- *Filtered subsets:* Subsets satisfying a predicate $P(A)$, such that $G(H) = \{A \subseteq H \mid P(A)\}$.
- *Fuzzy subsets [360]:* $G(H) = \{(A, \mu_A) \mid A \subseteq H, \mu_A : A \to [0, 1]\}$, where μ_A defines the degree of membership for each element in A.
- *Rough subsets [255]:* Defined in terms of lower and upper approximations, $G(H) = \{(A, \underline{A}, \overline{A}) \mid A \subseteq H\}$, where:

$$\underline{A} = \{x \in H \mid P(x) \text{ is definitely true}\}, \quad \overline{A} = \{x \in H \mid P(x) \text{ is possibly true}\}.$$

- *Neutrosophic subsets [302]:* $G(H) = \{(A, T_A, I_A, F_A) \mid A \subseteq H, T_A, I_A, F_A : A \to [0, 1]\}$, where:

$$0 \leq T_A(x) + I_A(x) + F_A(x) \leq 3 \quad \text{for all } x \in A,$$

  and $T_A(x)$, $I_A(x)$, and $F_A(x)$ represent the degrees of truth, indeterminacy, and falsity, respectively.
- *Plithogenic subsets [307, 318]:* $G(H) = \{(A, v, Pv, pdf, pCF) \mid A \subseteq H\}$, where:
  - v is an attribute.
  - Pv is the range of possible values for v.
  - $pdf : A \times Pv \to [0, 1]^s$ is the *Degree of Appurtenance Function (DAF).*
  - $pCF : Pv \times Pv \to [0, 1]^t$ is the *Degree of Contradiction Function (DCF)* satisfying:

$$pCF(a, a) = 0, \quad pCF(a, b) = pCF(b, a) \quad \text{for all } a, b \in Pv.$$

Theorem 4.2. *The Generalized n-th Powerset can represent the structure of supervertices and superedges in an n-SuperHyperGraph.*

Proof. Let $H = V_0$ be the base set of vertices in a graph or hypergraph. The n-th powerset $P_n(H)$ recursively defines the n-level structure of subsets of V_0, where:

$$P_1(H) = P(V_0), \quad P_2(H) = P(P(V_0)), \quad \ldots, \quad P_n(H) = P(P_{n-1}(H)).$$

Each level $P_k(H)$ contains subsets that correspond to vertices, supervertices, or higher-level structures.

Similarly, consider the set $E(H)$ of edges or hyperedges in H. The n-th powerset $P_n(E(H))$ describes the hierarchical structure of edges, superedges, and their generalizations.

By including additional constraints, such as graph structures (V_G, E_G) for each subset, we can construct subsets that represent specific subgraphs or induced structures within the n-th powerset hierarchy.

For example:

- At $n = 0$, the vertices are elements of V_0 and edges are subsets of V_0.
- At $n = 1$, $P(V_0)$ defines supervertices as subsets of V_0, and $P(E(H))$ defines superedges as subsets of $E(H)$.

- At $n = 2$, $P(P(V_0))$ includes higher-order structures, such as subsets of supervertices, which are themselves subsets of V_0.

Since the n-th generalized powerset incorporates additional structures like labels, weights, and fuzzy memberships, it can represent complex relationships within supervertices and superedges, generalizing their structure.

Thus, the Generalized n-th Powerset fully encapsulates the hierarchy of supervertices and superedges. □

Definition 4.3 (Generalized Non-Empty n-th Powerset)**.** Define the n-th generalized non-empty powerset of H, denoted $G_n^*(H)$, recursively as:

$$G_1^*(H) = G^*(H),$$

$$G_{n+1}^*(H) = G^*\big(G_n^*(H)\big),$$

where $G^*(H)$ is the non-empty subset operator under the generalized powerset $G(H)$, satisfying $G^*(H) \subseteq G(H) \setminus \{\emptyset\}$.

Definition 4.4 (Fuzzy, Neutrosophic, and Plithogenic n-th Powerset)**.** (cf. [309]) Let H be a set or a mathematical structure. Define the n-th fuzzy, neutrosophic, and plithogenic powersets of H, denoted $F_n(H)$, $N_n(H)$, and $Pn_n(H)$, respectively, as follows:

$$F_1(H) = F(H), \quad F_{n+1}(H) = F(F_n(H)),$$

$$N_1(H) = N(H), \quad N_{n+1}(H) = N(N_n(H)),$$

$$Pn_1(H) = Pn(H), \quad Pn_{n+1}(H) = Pn(Pn_n(H)).$$

Here:

- $F(H) = \{(A, \mu_A) \mid A \subseteq H, \mu_A : A \to [0, 1]\}$.
- $N(H) = \{(A, T_A, I_A, F_A) \mid A \subseteq H, T_A, I_A, F_A : A \to [0, 1]\}$.
- $Pn(H) = \{(A, v, Pv, pdf, pCF) \mid A \subseteq H\}$, with attributes and functions as defined above.

Example 4.5. Let $H = \{a, b, c\}$. Define $G(H)$ as the set of labeled subsets:

$$G(H) = \{(A, \ell) \mid A \subseteq H, \ell \in L\},$$

where $L = \{\text{"red"}, \text{"blue"}\}$. The first generalized powerset $G_1(H)$ is given by:

$$G_1(H) = \{(\emptyset, \ell), (\{a\}, \ell), (\{b\}, \ell), (\{a, b\}, \ell), \cdots \mid \ell \in L\}.$$

For higher n, the elements of $G_n(H)$ are labeled subsets of $G_{n-1}(H)$, creating hierarchical structures with additional labels.

Example 4.6. Let $H = \{a, b, c\}$. Define $F(H)$ as the set of fuzzy subsets:

$$F(H) = \{(A, \mu_A) \mid A \subseteq H, \mu_A : A \to [0, 1]\}.$$

For example:

$$(A, \mu_A) = \big(\{a, b\}, \mu_A\big), \quad \mu_A(a) = 0.8,\ \mu_A(b) = 0.5.$$

5 Future Tasks

This section outlines the future directions of this research. Building upon the various graph concepts introduced earlier, we aim to explore their applications and underlying mathematical structures in greater depth.

5.1 Adding Conditions of Uncertain Sets to Superhyperconcepts

We plan to examine how these concepts evolve when incorporating the frameworks of Fuzzy Sets [280,360–366], Neutrosophic Sets [110, 111, 119, 121, 122, 302–304, 317], Soft Sets [228, 242], Hypersoft Sets [109, 115, 305, 314], superhypersoft sets [55, 82, 285, 313, 319], Hyperfuzzy sets [114, 132, 183, 320], HyperNeutrosophic sets [114], and Rough Sets [255–257, 257, 258]. These extensions will provide valuable insights into the theoretical and practical implications of these graph structures.

5.2 n-Superhyperword and n-Superhyperlanguage

In this subsection, we define the notions of a hyperlanguage and an n-superhyperlanguage. Intuitively, a hyperlanguage [38, 39, 104] generalizes the concept of a language by allowing its elements to be sets of words rather than individual words. We then extend this idea hierarchically to n-superhyperlanguages, which are based on iterated power sets of the set of words. Although this definition is still in its conceptual stage, it is formally presented below. We anticipate that future research will explore the mathematical structures and applications of these concepts.

Definition 5.1 (Hyperword and Hyperlanguage)**.** [38, 39, 104, 273] Let Σ be a finite alphabet, and let Σ^* denote the set of all finite words over Σ.

1. A *hyperword* over Σ is a nonempty subset of Σ^*. In other words, a hyperword is an element of the power set $\mathcal{P}(\Sigma^*)$.

2. A *hyperlanguage* over Σ is a set of hyperwords over Σ. Thus, a hyperlanguage H is a subset of $\mathcal{P}(\Sigma^*)$. Formally:

$$H \subseteq \mathcal{P}(\Sigma^*).$$

A hyperlanguage can therefore be viewed as a *set of sets of words* over Σ.

Definition 5.2 (n-Superhyperword and n-Superhyperlanguage)**.** We now generalize this construction to multiple levels. Define the iterated power sets as follows:

$$\mathcal{P}^0(\Sigma^*) := \Sigma^*, \quad \mathcal{P}^{k+1}(\Sigma^*) := \mathcal{P}(\mathcal{P}^k(\Sigma^*)), \text{ for all } k \geq 0.$$

1. An *n-superhyperword* over Σ is an element of $\mathcal{P}^n(\Sigma^*)$. In particular:

$$\mathcal{P}^1(\Sigma^*) = \mathcal{P}(\Sigma^*) \text{ consists of hyperwords,}$$

$$\mathcal{P}^2(\Sigma^*) = \mathcal{P}(\mathcal{P}(\Sigma^*)) \text{ consists of sets of hyperwords, and so forth.}$$

2. An *n-superhyperlanguage* over Σ is a subset of $\mathcal{P}^n(\Sigma^*)$. Formally:

$$L \subseteq \mathcal{P}^n(\Sigma^*).$$

Thus, an n-superhyperlanguage is a *set of* $(n-1)$*-superhyperwords*, generalizing the concept of a hyperlanguage to n-th level power sets of words.

Theorem 5.3. *The notion of an n-superhyperlanguage generalizes the notion of a hyperlanguage. In particular:*

A hyperlanguage is precisely a 1*-superhyperlanguage.*

Proof. By Definition 5.1, a hyperlanguage is a subset of $\mathcal{P}(\Sigma^*)$. Note that $\mathcal{P}^1(\Sigma^*) = \mathcal{P}(\Sigma^*)$. Thus, a hyperlanguage $H \subseteq \mathcal{P}(\Sigma^*)$ is exactly a 1-superhyperlanguage.

In other words, setting $n = 1$ in Definition 5.2 recovers the definition of a hyperlanguage. Hence, n-superhyperlanguages form a hierarchy of increasingly complex structures, with hyperlanguages occupying the first level of this hierarchy. □

5.3 Natural HyperLanguage Processing and n-superhyperlanguage Processing

Natural Language Processing (NLP) has been extensively studied in various contexts and applications [34, 54, 70, 73, 74, 94, 137, 215, 231, 232, 348, 355].

In this subsection, we introduce an extension of NLP utilizing the concepts of hyperlanguage and n-superhyperlanguage, leading to the frameworks of *Natural Hyperlanguage Processing* and *n-Superhyperlanguage Processing*. Since these definitions are currently at the conceptual stage, it is anticipated that future studies will explore more refined definitions, as well as research and development into methods of implementation and practical applications.

Definition 5.4 (Natural Language Processing (NLP))**.** (cf. [34,70,231]) Let Σ be a finite alphabet representing the vocabulary of a natural language, and let Σ^* denote the set of all finite sequences (words) over Σ. A *language* $\mathcal{L}$ is a subset $\mathcal{L} \subseteq \Sigma^*$.

An NLP system is a tuple:

$$\mathcal{N} = (\Sigma, \mathcal{L}, \mathcal{P}, \mathcal{M}, \mathcal{T}),$$

where:

1. Σ: A finite alphabet of symbols.
2. $\mathcal{L} \subseteq \Sigma^*$: The language, defined by some grammar $\mathcal{G}$.
3. $\mathcal{P} : \mathcal{L} \to [0, 1]$: A probability model [281] assigning probabilities to each $w \in \mathcal{L}$:

$$\mathcal{P}(w) = P(w \mid \theta),$$

   where θ represents model parameters.
4. $\mathcal{M} : \mathcal{L} \to O$: A mapping function that transforms each $w \in \mathcal{L}$ into a structured output $o \in O$ (e.g., a parse tree, a translation).
5. $\mathcal{T} : \mathcal{L} \times \mathcal{L} \to \mathbb{R}$: A similarity measure between pairs of words or sentences.

We now define Natural Hyperlanguage Processing, which extends NLP to operate on hyperlanguages rather than languages.

Definition 5.5 (Natural Hyperlanguage Processing (NHP))**.** Let Σ be a finite alphabet, and let $\mathcal{H} \subseteq \mathcal{P}(\Sigma^*)$ be a hyperlanguage (a set of sets of words).

A Natural Hyperlanguage Processing system is a tuple:

$$\mathcal{N}^{HL} = (\Sigma, \mathcal{H}, \mathcal{P}^{HL}, \mathcal{M}^{HL}, \mathcal{T}^{HL}),$$

where:

1. Σ: A finite alphabet.
2. $\mathcal{H} \subseteq \mathcal{P}(\Sigma^*)$: A hyperlanguage.
3. $\mathcal{P}^{HL} : \mathcal{H} \to [0, 1]$: A probability model assigning probabilities to *hyperwords* $H \in \mathcal{H}$.
4. $\mathcal{M}^{HL} : \mathcal{H} \to O$: A mapping function transforming each hyperword $H \in \mathcal{H}$ into a structured output $o \in O$.
5. $\mathcal{T}^{HL} : \mathcal{H} \times \mathcal{H} \to \mathbb{R}$: A similarity measure defined between pairs of hyperwords.

We further generalize to n-superhyperlanguages.

Definition 5.6 (Natural n-Superhyperlanguage Processing (NnSHP))**.** Let Σ be a finite alphabet, and let $\mathcal{H}^{(n)} \subseteq \mathcal{P}^n(\Sigma^*)$ be an n-superhyperlanguage.

A Natural n-Superhyperlanguage Processing system is a tuple:

$$\mathcal{N}^{(n)} = (\Sigma, \mathcal{H}^{(n)}, \mathcal{P}^{(n)}, \mathcal{M}^{(n)}, \mathcal{T}^{(n)}),$$

where:

1. Σ: A finite alphabet.
2. $\mathcal{H}^{(n)} \subseteq \mathcal{P}^n(\Sigma^*)$: An n-superhyperlanguage.
3. $\mathcal{P}^{(n)} : \mathcal{H}^{(n)} \to [0, 1]$: A probability model assigning probabilities to n-superhyperwords.
4. $\mathcal{M}^{(n)} : \mathcal{H}^{(n)} \to O$: A mapping function from n-superhyperwords to structured outputs.
5. $\mathcal{T}^{(n)} : \mathcal{H}^{(n)} \times \mathcal{H}^{(n)} \to \mathbb{R}$: A similarity measure on n-superhyperwords.

Theorem 5.7. *Natural Hyperlanguage Processing (NHP) generalizes Natural Language Processing (NLP).*

Proof. Consider an NHP system $\mathcal{N}^{HL} = (\Sigma, \mathcal{H}, \mathcal{P}^{HL}, \mathcal{M}^{HL}, \mathcal{T}^{HL})$ where $\mathcal{H} \subseteq \mathcal{P}(\Sigma^*)$.

If we restrict $\mathcal{H}$ so that every hyperword is a singleton set, i.e., for every $H \in \mathcal{H}$, $H = \{w\}$ for some $w \in \Sigma^*$, then there is a bijection between hyperwords in $\mathcal{H}$ and words in a language $\mathcal{L} \subseteq \Sigma^*$.

Under this restriction:

$$\mathcal{H} \cong \mathcal{L}, \quad \text{with } H = \{w\} \leftrightarrow w.$$

In this case, $\mathcal{N}^{HL}$ reduces to:

$$(\Sigma, \mathcal{L}, \mathcal{P}^{HL}, \mathcal{M}^{HL}, \mathcal{T}^{HL}),$$

which is structurally identical to the NLP definition $(\Sigma, \mathcal{L}, \mathcal{P}, \mathcal{M}, \mathcal{T})$.

Thus, NLP is a special case of NHP, proving that NHP generalizes NLP. □

Theorem 5.8. *Natural n-Superhyperlanguage Processing (NnSHP) generalizes both NLP and NHP.*

Proof. By definition, an n-superhyperlanguage $\mathcal{H}^{(n)} \subseteq \mathcal{P}^n(\Sigma^*)$.

For $n = 1$, we have $\mathcal{H}^{(1)} \subseteq \mathcal{P}(\Sigma^*)$, which is a hyperlanguage. Thus, an N1SHP system:

$$\mathcal{N}^{(1)} = (\Sigma, \mathcal{H}^{(1)}, \mathcal{P}^{(1)}, \mathcal{M}^{(1)}, \mathcal{T}^{(1)})$$

coincides with an NHP system:

$$\mathcal{N}^{HL} = (\Sigma, \mathcal{H}, \mathcal{P}^{HL}, \mathcal{M}^{HL}, \mathcal{T}^{HL}).$$

Hence, NHP is a special case of NnSHP at $n = 1$.

From Theorem 5.7, we know NHP generalizes NLP. Since NnSHP generalizes NHP, it also generalizes NLP. Concretely, by setting $n = 1$ and then restricting hyperwords to singletons, we recover the NLP scenario.

Thus, NnSHP includes both NHP and NLP as special cases, proving that NnSHP generalizes both NLP and NHP. □

Funding

This study did not receive financial support from any external sources.

Acknowledgments

We extend our heartfelt gratitude to all those whose guidance and encouragement played a crucial role in the completion of this research. We also express our sincere thanks to the readers for their interest in and engagement with this work. Lastly, we deeply appreciate the authors of the references cited, whose foundational contributions have significantly enriched this study.

Data Availability

This paper is purely mathematical and theoretical in nature. Therefore, no data analysis was performed as part of this study. We hope future researchers will consider conducting data analysis or related investigations as necessary.

Ethical Approval

This paper is focused on mathematical and theoretical research. As such, it does not involve any studies on human participants or animals.

Conflicts of Interest

The authors declare that there are no conflicts of interest related to the publication of this paper.

Disclaimer

This study primarily addresses theoretical advancements and has not been applied or tested in practical settings. Future research may aim to validate and refine the proposed methods through empirical studies. Although every effort has been made to ensure the accuracy and proper citation of references, unintentional errors or omissions may exist. Readers are encouraged to independently verify the cited materials. The views and interpretations expressed in this paper are solely those of the authors and do not necessarily reflect the perspectives of their affiliated institutions.

References

[1] Ahmad Abu-Khazneh. *Matchings and covers of multipartite hypergraphs*. PhD thesis, London School of Economics and Political Science, 2016.

[2] Louigi Addario-Berry, Nicolas Broutin, Christina Goldschmidt, and Grégory Miermont. The scaling limit of the minimum spanning tree of the complete graph. 2017.

[3] Isolde Adler, Tomáš Gavenčiak, and Tereza Klimošová. Hypertree-depth and minors in hypergraphs. *Theoretical Computer Science*, 463:84–95, 2012.

[4] Isolde Adler, Georg Gottlob, and Martin Grohe. Hypertree width and related hypergraph invariants. *European Journal of Combinatorics*, 28(8):2167–2181, 2007.

[5] Alfred V. Aho, Michael R Garey, and Jeffrey D. Ullman. The transitive reduction of a directed graph. *SIAM Journal on Computing*, 1(2):131–137, 1972.

[6] Muhammad Akram and Noura Omair Alshehri. Intuitionistic fuzzy cycles and intuitionistic fuzzy trees. *The Scientific World Journal*, 2014(1):305836, 2014.

[7] Muhammad Akram and Wieslaw A. Dudek. Intuitionistic fuzzy hypergraphs with applications. *Inf. Sci.*, 218:182–193, 2013.

[8] Muhammad Akram, Feng Feng, Shahzad Sarwar, and Youne Bae Jun. Certain types of vague graphs. *University Politehnica of Bucharest Scientific Bulletin Series A*, 76(1):141–154, 2014.

[9] Muhammad Akram, A Nagoor Gani, and A Borumand Saeid. Vague hypergraphs. *Journal of Intelligent & Fuzzy Systems*, 26(2):647–653, 2014.

[10] Muhammad Akram and Anam Luqman. Certain networks models using single-valued neutrosophic directed hypergraphs. *Journal of Intelligent & Fuzzy Systems*, 33(1):575–588, 2017.

[11] Muhammad Akram, Hafsa M Malik, Sundas Shahzadi, and Florentin Smarandache. Neutrosophic soft rough graphs with application. *Axioms*, 7(1):14, 2018.

[12] Muhammad Akram and Gulfam Shahzadi. *Operations on single-valued neutrosophic graphs*. Infinite Study, 2017.

[13] Muhammad Akram, Sundas Shahzadi, and AB Saeid. Single-valued neutrosophic hypergraphs. *TWMS Journal of Applied and Engineering Mathematics*, 8(1):122–135, 2018.

[14] Sinan G Aksoy, Cliff Joslyn, Carlos Ortiz Marrero, Brenda Praggastis, and Emilie Purvine. Hypernetwork science via high-order hypergraph walks. *EPJ Data Science*, 9(1):16, 2020.

[15] Kazutoshi Ando, Satoru Fujishige, and Toshio Nemoto. Decomposition of a bidirected graph into strongly connected components and its signed poset structure. *Discrete Applied Mathematics*, 68(3):237–248, 1996.

[16] Alessia Antelmi, Gennaro Cordasco, Mirko Polato, Vittorio Scarano, Carmine Spagnuolo, and Dingqi Yang. A survey on hypergraph representation learning. *ACM Computing Surveys*, 56(1):1–38, 2023.

[17] Armen S Asratian, Tristan MJ Denley, and Roland Häggkvist. *Bipartite graphs and their applications*, volume 131. Cambridge university press, 1998.

[18] Bethany Austhof and Sean English. Nearly-regular hypergraphs and saturation of berge stars. *arXiv preprint arXiv:1812.00472*, 2018.

[19] László Babai. Graph isomorphism in quasipolynomial time. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 684–697, 2016.

[20] László Babai, Paul Erdos, and Stanley M Selkow. Random graph isomorphism. *SIaM Journal on computing*, 9(3):628–635, 1980.

[21] M Amin Bahmanian and Mateja Šajna. Hypergraphs: connection and separation. *arXiv preprint arXiv:1504.04274*, 2015.

[22] Mohammad A Bahmanian and Mateja Sajna. Connection and separation in hypergraphs. *Theory and Applications of Graphs*, 2(2):5, 2015.

[23] József Balogh, Robert Morris, and Wojciech Samotij. Independent sets in hypergraphs. *Journal of the American Mathematical Society*, 28(3):669–709, 2015.

[24] Hans-Jürgen Bandelt and Henry Martyn Mulder. Pseudo-median graphs: decomposition via amalgamation and cartesian multiplication. *Discrete mathematics*, 94(3):161–180, 1991.

[25] Jørgen Bang-Jensen and Gregory Z Gutin. *Digraphs: theory, algorithms and applications*. Springer Science & Business Media, 2008.

[26] Claudio DT Barros, Matheus RF Mendonça, Alex B Vieira, and Artur Ziviani. A survey on embedding dynamic graphs. *ACM Computing Surveys (CSUR)*, 55(1):1–37, 2021.

[27] Fabian Beck, Michael Burch, Stephan Diehl, and Daniel Weiskopf. The state of the art in visualizing dynamic graphs. *EuroVis (STARs)*, 2014.

[28] Jozsef Beck. On size ramsey number of paths, trees, and circuits. i. *Journal of Graph Theory*, 7(1):115–129, 1983.

[29] Michael Bendersky and W Bruce Croft. Modeling higher-order term dependencies in information retrieval using query hypergraphs. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 941–950, 2012.

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

[31] Robert Berke. *Colorings and transversals of graphs*. PhD thesis, ETH Zurich, 2008.

[32] Alaa Bessadok, Mohamed Ali Mahjoub, and Islem Rekik. Brain multigraph prediction using topology-aware adversarial graph neural network. *Medical image analysis*, 72:102090, 2021.

[33] Anushree Bhattacharya and Madhumangal Pal. A fuzzy graph theory approach to the facility location problem: A case study in the indian banking system. *Mathematics*, 11(13):2992, 2023.

[34] Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.", 2009.

[35] Pranab Biswas, Surapati Pramanik, and Bibhas Chandra Giri. Single valued bipolar pentapartitioned neutrosophic set and its application in madm strategy. 2022.

[36] Mindaugas Bloznelis. Degree and clustering coefficient in sparse random intersection graphs. 2013.

[37] B´ ela Bollobás, Oliver Riordan, Joel Spencer, and Gábor Tusnády. The degree sequence of a scale-free random graph process. *Random Structures & Algorithms*, 18(3):279–290, 2001.

[38] Borzoo Bonakdarpour and Sarai Sheinvald. Automata for hyperlanguages. *arXiv preprint arXiv:2002.09877*, 2020.

[39] Borzoo Bonakdarpour and Sarai Sheinvald. Finite-word hyperlanguages. *Information and Computation*, 295:104944, 2023.

[40] Danail Bonchev. *Chemical graph theory: introduction and fundamentals*. Routledge, 2018.

[41] John Adrian Bondy, Uppaluri Siva Ramachandra Murty, et al. *Graph theory with applications*, volume 290. Macmillan London, 1976.

[42] Endre Boros, Khaled Elbassioni, Vladimir Gurvich, and Leonid Khachiyan. Generating maximal independent sets for hypergraphs with bounded edge-intersections. In *Latin American Symposium on Theoretical Informatics*, pages 488–498. Springer, 2004.

[43] RA Borzooei and HOSSEIN RASHMANLOU. Degree of vertices in vague graphs. *Journal of applied mathematics & informatics*, 33(5_6):545–557, 2015.

[44] RA Borzooei and Hossein Rashmanlou. More results on vague graphs. *UPB Sci. Bull. Ser. A*, 78(1):109–122, 2016.

[45] Rajab Ali Borzooei, Hossein Rashmanlou, Sovan Samanta, and Madhumangal Pal. Regularity of vague graphs. *Journal of Intelligent & Fuzzy Systems*, 30(6):3681–3689, 2016.

[46] Florian Boudin. Unsupervised keyphrase extraction with multipartite graphs. *arXiv preprint arXiv:1803.08721*, 2018.

[47] Candida Bowtell and Richard Mycroft. Matchings in multipartite hypergraphs. *arXiv preprint arXiv:2403.05219*, 2024.

[48] Luis Boza, Eugenio Manuel Fedriani, Juan Núñez, Ana María Pacheco, and María Trinidad Villar. Directed pseudo-graphs and lie algebras over finite fields. *Czechoslovak Mathematical Journal*, 64:229–239, 2014.

[49] Jeremy T Bradley, Nicholas J Dingle, William J Knottenbelt, and Helen J Wilson. Hypergraph-based parallel computation of passage time densities in large semi-markov models. *Linear Algebra and its Applications*, 386:311–334, 2004.

[50] Andreas Brandstädt, Van Bang Le, and Jeremy P Spinrad. *Graph classes: a survey*. SIAM, 1999.

[51] Lukas Braun. Invariant rings of sums of fundamental representations of sln and colored hypergraphs. *Advances in Mathematics*, 389:107929, 2021.

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

[53] Alain Bretto and Luc Gillibert. Hypergraph-based image representation. In *Workshop on Graph Based Representations in Pattern Recognition*, 2005.

[54] Eric Brill. Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging. *Computational linguistics*, 21(4):543–565, 1995.

[55] Oswaldo Edison García Brito, Andrea Sofía Ribadeneira Vacacela, Carmen Hortensia Sánchez Burneo, and Mónica Cecilia Jimbo Galarza. English for specific purposes in the medical sciences to strengthen the professional profile of the higher education medicine student: a knowledge representation using superhypersoft sets. *Neutrosophic Sets and Systems*, 74:84–91, 2024.

[56] S Broumi and Tomasz Witczak. Heptapartitioned neutrosophic soft set. *International Journal of Neutrosophic Science*, 18(4):270–290, 2022.

[57] Said Broumi, Mohamed Talea, Assia Bakali, and Florentin Smarandache. Interval valued neutrosophic graphs. *Critical Review, XII*, 2016:5–33, 2016.

[58] Anita C Burris and Richard H Schelp. Vertex-distinguishing proper edge-colorings. *Journal of graph theory*, 26(2):73–82, 1997.

[59] Hongyun Cai, Vincent W Zheng, and Kevin Chen-Chuan Chang. A comprehensive survey of graph embedding: Problems, techniques, and applications. *IEEE transactions on knowledge and data engineering*, 30(9):1616–1637, 2018.

[60] Andrew E. Caldwell, Andrew B. Kahng, and Igor L. Markov. Design and implementation of move-based heuristics for vlsi hypergraph partitioning. *ACM J. Exp. Algorithmics*, 5:5, 2000.

[61] Kauê Cardoso and Vilmar Trevisan. Energies of hypergraphs. *arXiv preprint arXiv:1912.03224*, 2019.

[62] Gary Chartrand. *Introductory graph theory*. Courier Corporation, 2012.

[63] Sourav Chatterjee, Persi Diaconis, and Allan Sly. Random graphs with a given degree sequence. 2011.

[64] GT Chen and Richard H Schelp. Graphs with linearly bounded ramsey numbers. *Journal of Combinatorial Theory, Series B*, 57(1):138–149, 1993.

[65] Junwu Chen and Philippe Schwaller. Molecular hypergraph neural networks. *The Journal of Chemical Physics*, 160(14), 2024.

[66] Yan-Ying Chen, Winston H Hsu, and Hong-Yuan Mark Liao. Discovering informative social subgraphs and predicting pairwise relationships from group photos. In *Proceedings of the 20th ACM international conference on Multimedia*, pages 669–678, 2012.

[67] Yangyang Cheng and Katherine Staden. Transversals via regularity. *arXiv preprint arXiv:2306.03595*, 2023.

[68] Paul Chew. There is a planar graph almost as good as the complete graph. In *Proceedings of the second annual symposium on Computational geometry*, pages 169–177, 1986.

[69] Eli Chien, Chao Pan, Jianhao Peng, and Olgica Milenkovic. You are allset: A multiset function framework for hypergraph neural networks. *ArXiv*, abs/2106.13264, 2021.

[70] KR1442 Chowdhary and KR Chowdhary. Natural language processing. *Fundamentals of artificial intelligence*, pages 603–649, 2020.

[71] Fan Chung and Linyuan Lu. The average distances in random graphs with given expected degrees. *Proceedings of the National Academy of Sciences*, 99(25):15879–15882, 2002.

[72] FRK Chung. The average distance and the independence. *J. Graph Theory*, 12:229–235, 1988.

[73] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167, 2008.

[74] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *Journal of machine learning research*, 12:2493–2537, 2011.

[75] David Conlon, Jacob Fox, and Benny Sudakov. Ramsey numbers of sparse hypergraphs. *Random Structures & Algorithms*, 35(1):1–14, 2009.

[76] David Conlon, Jacob Fox, and Benny Sudakov. Hypergraph ramsey numbers. *Journal of the American Mathematical Society*, 23(1):247–266, 2010.

[77] Derek G Corneil and Calvin C Gotlieb. An efficient algorithm for graph isomorphism. *Journal of the ACM (JACM)*, 17(1):51–64, 1970.

[78] Roohallah Daneshpayeh and Sirus Jahanpanah. On tree graphs and arboreal hypergraphs.

[79] Suman Das, Rakhal Das, and Surapati Pramanik. Single valued pentapartitioned neutrosophic graphs. *Neutrosophic Sets and Systems*, 50(1):225–238, 2022.

[80] Suman Das, Rakhal Das, and Binod Chandra Tripathy. Topology on rough pentapartitioned neutrosophic set. *Iraqi Journal of Science*, 2022.

[81] Milind Dawande, Pinar Keskinocak, Jayashankar M Swaminathan, and Sridhar Tayur. On bipartite and multipartite clique problems. *Journal of Algorithms*, 41(2):388–403, 2001.

[82] María Eugenia Lucena de Ustáriz, Francisco Javier Ustáriz-Fajardo, Adriana Monserrath Monge Moreno, Adriana Isabel Rodríguez Basantes, Lisbeth Josefina Reales Chacón, Rosa Elisa Cruz Tenempaguay, Verónica Paulina Cáceres Manzano, and Nathaly Kassandra Moscoso Moreno. Antibacterial and antifungal properties of hedyosmum cuatrecazanum occhioni essential oil: A promising natural alternative studied using neutrosophic superhypersoft sets. *Neutrosophic Sets and Systems*, 74:14–23, 2024.

[83] Narsingh Deo. *Graph theory with applications to engineering and computer science*. Courier Dover Publications, 2016.

[84] Tyler Derr, Yao Ma, and Jiliang Tang. Signed graph convolutional networks. *2018 IEEE International Conference on Data Mining (ICDM)*, pages 929–934, 2018.

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

[86] Reinhard Diestel. Graph theory 3rd ed. *Graduate texts in mathematics*, 173(33):12, 2005.

[87] Reinhard Diestel. *Graph theory*. Springer (print edition); Reinhard Diestel (eBooks), 2024.

[88] Mitre C Dourado, Fábio Protti, and Jayme L Szwarcfiter. Complexity aspects of the helly property: Graphs and hypergraphs. *The Electronic Journal of Combinatorics*, pages DS17–Jun, 2012.

[89] Andrzej Dudek, Steven La Fleur, Dhruv Mubayi, and Vojtech Rodl. On the size-ramsey number of hypergraphs. *Journal of Graph Theory*, 86(1):104–121, 2017.

[90] Andrew L Dulmage and Nathan S Mendelsohn. Coverings of bipartite graphs. *Canadian Journal of Mathematics*, 10:517–534, 1958.

[91] Jack Edmonds and Ellis L Johnson. Matching: A well-solved class of integer linear programs. In *Combinatorial Optimization-Eureka, You Shrink! Papers Dedicated to Jack Edmonds 5th International Workshop Aussois, France, March 5–9, 2001 Revised Papers*, pages 27–30. Springer, 2003.

[92] Thomas Eiter and Georg Gottlob. Identifying the minimal transversals of a hypergraph and related problems. *SIAM Journal on Computing*, 24(6):1278–1304, 1995.

[93] Thomas Eiter and Georg Gottlob. Hypergraph transversal computation and related problems in logic and ai. In *European Workshop on Logics in Artificial Intelligence*, pages 549–564. Springer, 2002.

[94] Ralph Elliott, John RW Glauert, JR Kennaway, and Ian Marshall. The development of language processing support for the visicast project. In *Proceedings of the fourth international ACM conference on Assistive technologies*, pages 101–108, 2000.

[95] Pál Erdős, Miklós Simonovits, and Vera T Sós. Anti-ramsey theorems. 1975.

[96] Paul Erdös. Graph theory and probability. *Canadian Journal of Mathematics*, 11:34–38, 1959.

[97] Paul Erdos, Ralph J Faudree, Cecil C Rousseau, and Richard H Schelp. The size ramsey number. *Periodica Mathematica Hungarica*, 9(1-2):145–161, 1978.

[98] Friedrich Esser and Frank Harary. On the spectrum of a complete multipartite graph. *European Journal of Combinatorics*, 1(3):211–218, 1980.

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

[100] Yifan Feng, Haoxuan You, Zizhao Zhang, R. Ji, and Yue Gao. Hypergraph neural networks. In *AAAI Conference on Artificial Intelligence*, 2018.

[101] Scott Fortin. The graph isomorphism problem. 1996.

[102] Leslie R Foulds. *Graph theory applications*. Springer Science & Business Media, 1995.

[103] Ronald C. Freiwald. An introduction to set theory and topology. 2014.

[104] Hadar Frenkel and Sarai Sheinvald. Realizable and context-free hyperlanguages. *arXiv preprint arXiv:2209.10306*, 2022.

[105] Nir Friedman. Inferring cellular networks using probabilistic graphical models. *Science*, 303(5659):799–805, 2004.

[106] Takaaki Fujita. Bunch graph in fuzzy, neutrosophic, and plithogenic graphs, June 2024. License: CC BY 4.0.

[107] Takaaki Fujita. General plithogenic hypergraph and some uncertain hypergraph classes. 2024.

[108] Takaaki Fujita. Mixed graph in fuzzy, neutrosophic, and plithogenic graphs. June 2024.

[109] Takaaki Fujita. Note for hypersoft filter and fuzzy hypersoft filter. *Multicriteria Algorithms With Applications*, 5:32–51, 2024.

[110] Takaaki Fujita. Note for neutrosophic incidence and threshold graph. *SciNexuses*, 1:97–125, 2024.

[111] Takaaki Fujita. A review of the hierarchy of plithogenic, neutrosophic, and fuzzy graphs: Survey and applications. *ResearchGate(Preprint)*, 2024.

[112] Takaaki Fujita. Short note of supertree-width and n-superhypertree-width. *Neutrosophic Sets and Systems*, 77:54–78, 2024.

[113] Takaaki Fujita. Survey of trees, forests, and paths in fuzzy and neutrosophic graphs. July 2024.

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

[115] Takaaki Fujita. A comprehensive discussion on fuzzy hypersoft expert, superhypersoft, and indetermsoft graphs. *Neutrosophic Sets and Systems*, 77:241–263, 2025.

[116] Takaaki Fujita. Fundamental computational problems and algorithms for superhypergraphs. 2025.

[117] Takaaki Fujita. Review of some superhypergraph classes: Directed, bidirected, soft, and rough. 2025.

[118] Takaaki Fujita. Superhypergraph neural networks and plithogenic graph neural networks: Theoretical foundations. 2025.

[119] Takaaki Fujita and Florentin Smarandache. Antipodal turiyam neutrosophic graphs. *Neutrosophic Optimization and Intelligent Systems*, 5:1–13, 2024.

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

[121] Takaaki Fujita and Florentin Smarandache. Neutrosophic circular-arc graphs and proper circular-arc graphs. *Neutrosophic Sets and Systems*, 78:1–30, 2024.

[122] Takaaki Fujita and Florentin Smarandache. Uncertain automata and uncertain graph grammar. *Neutrosophic Sets and Systems*, 74:128–191, 2024.

[123] Giorgio Gallo, Giustino Longo, Stefano Pallottino, and Sang Nguyen. Directed hypergraphs and applications. *Discrete applied mathematics*, 42(2-3):177–201, 1993.

[124] Giorgio Galloy and Gabriella Ragoy. The satissability problem for the schh oennnkel-bernays fragment: Partial instantiation and hypergraph algorithms. 1994.

[125] A Nagoor Gani and K Radha. On regular fuzzy graphs. 2008.

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

[127] Ramón García-Domenech, Jorge Gálvez, Jesus V de Julián-Ortiz, and Lionello Pogliani. Some new trends in chemical graph theory. *Chemical Reviews*, 108(3):1127–1169, 2008.

[128] Johannes Gasteiger, Janek Groß, and Stephan Günnemann. Directional message passing for molecular graphs. *arXiv preprint arXiv:2003.03123*, 2020.

[129] Edward Gaughan. 1.1 sequences and convergence. *Introduction to Analysis. AMS*, 2009.

[130] Laura Gellert and Raman Sanyal. On degree sequences of undirected, directed, and bidirected graphs. *European Journal of Combinatorics*, 64:113–124, 2017.

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

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

[133] Saptarshi Ghosh, Pushkar Kane, and Niloy Ganguly. Identifying overlapping communities in folksonomies or tripartite hypergraphs. In *Proceedings of the 20th international conference companion on World wide web*, pages 39–40, 2011.

[134] Gourab Ghoshal, Vinko Zlatic, Guido Caldarelli, and Mark EJ Newman. Random hypergraphs and their applications. *Physical Review EStatistical, Nonlinear, and Soft Matter Physics*, 79(6):066118, 2009.

[135] Puspendu Giri, Somnath Paul, and Bijoy Krishna Debnath. A fuzzy graph theory and matrix approach (fuzzy gtma) to select the best renewable energy alternative in india. *Applied Energy*, 358:122582, 2024.

[136] Bornali Gogoi and Bichitra Kalita. Algorithm to color a circuit dual hypergraph for vlsi circuit. 2014.

[137] Yoav Goldberg. *Neural network methods in natural language processing*. Morgan & Claypool Publishers, 2017.

[138] Georg Gottlob, Gianluigi Greco, Nicola Leone, and Francesco Scarcello. Hypertree decompositions: Questions and answers. In *Proceedings of the 35th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, pages 57–74, 2016.

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

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

[141] Georg Gottlob and Reinhard Pichler. Hypergraphs in model checking: Acyclicity and hypertree-width versus clique-width. *SIAM Journal on Computing*, 33(2):351–378, 2004.

[142] Palash Goyal and Emilio Ferrara. Graph embedding techniques, applications, and performance: A survey. *Knowledge-Based Systems*, 151:78–94, 2018.

[143] Ronald L Graham, Vojtech Rodl, and Andrzej Rucinski. On graphs with linear ramsey numbers. *Journal of Graph Theory*, 35(3):176–192, 2000.

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

[145] Jean-Loup Guillaume and Matthieu Latapy. Bipartite graphs as models of complex networks. *Physica A: Statistical Mechanics and its Applications*, 371(2):795–813, 2006.

[146] Dimitrios Gunopulos, Heikki Mannila, Roni Khardon, and Hannu Toivonen. Data mining, hypergraph transversals, and machine learning. In *Proceedings of the sixteenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems*, pages 209–216, 1997.

[147] Venkatesan Guruswami and Euiwoong Lee. Inapproximability of h-transversal/packing. *SIAM Journal on Discrete Mathematics*, 31(3):1552–1571, 2017.

[148] Matthias Hagen. Lower bounds for three algorithms for transversal hypergraph generation. *Discrete Applied Mathematics*, 157(7):1460–1469, 2009.

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

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

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

[152] Xinchu Han, Ruijie Xu, Lijia Ma, Zengyang Shao, Yuan Bai, and Shanfeng Wang. Structural balance computation of signed hypergraphs via memetic algorithm. *2023 5th International Conference on Data-driven Optimization of Complex Systems (DOCS)*, pages 1–7, 2023.

[153] Robert Hancock, Katherine Staden, and Andrew Treglown. Independent sets in hypergraphs and ramsey properties of graphs and the integers. *SIAM Journal on Discrete Mathematics*, 33(1):153–188, 2019.

[154] Nasrin Hassanlou, Maryam Shoaran, and Alex Thomo. Probabilistic graph summarization. In *Web-Age Information Management: 14th International Conference, WAIM 2013, Beidaihe, China, June 14-16, 2013. Proceedings 14*, pages 545–556. Springer, 2013.

[155] Felix Hausdorff. *Set theory*, volume 119. American Mathematical Soc., 2021.

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

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

[158] Yixuan He, Xitong Zhang, Junjie Huang, Benedek Rozemberczki, Mihai Cucuringu, and Gesine Reinert. Pytorch geometric signed directed: a software package on graph neural networks for signed and directed graphs. In *Learning on Graphs Conference*, pages 12–1. PMLR, 2024.

[159] Michael A Henning and Anders Yeo. 2-colorings in k-regular k-uniform hypergraphs. *European Journal of Combinatorics*, 34(7):1192–1202, 2013.

[160] Ian Holyer. The np-completeness of edge-coloring. *SIAM Journal on computing*, 10(4):718–720, 1981.

[161] Karel Hrbacek and Thomas Jech. Introduction to set theory, revised and expanded. 2017.

[162] Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. Heterogeneous graph transformer. In *Proceedings of the web conference 2020*, pages 2704–2710, 2020.

[163] Jie Huang, Xin Liu, and Yangqiu Song. Hyper-path-based representation learning for hyper-networks. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 449–458, 2019.

[164] Junjie Huang, Huawei Shen, Liang Hou, and Xueqi Cheng. Signed graph attention networks. In *International Conference on Artificial Neural Networks*, 2019.

[165] Liangsong Huang, Yu Hu, Yuxia Li, PK Kishore Kumar, Dipak Koley, and Arindam Dey. A study of regular and irregular neutrosophic graphs with real life applications. *Mathematics*, 7(6):551, 2019.

[166] Yuchi Huang, Qingshan Liu, Shaoting Zhang, and Dimitris N Metaxas. Image retrieval via probabilistic hypergraph ranking. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 3376–3383. IEEE, 2010.

[167] S Satham Hussain, N Durga, Muhammad Aslam, G Muhiuddin, and Ganesh Ghorai. New concepts on quadripartitioned neutrosophic competition graph with application. *International Journal of Applied and Computational Mathematics*, 10(2):57, 2024.

[168] S Satham Hussain, Hossein Rashmonlou, R Jahir Hussain, Sankar Sahoo, Said Broumi, et al. Quadripartitioned neutrosophic graph structures. *Neutrosophic Sets and Systems*, 51(1):17, 2022.

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

[170] Dmitry I Ignatov. On closure operators related to maximal tricliques in tripartite hypergraphs. *Discrete Applied Mathematics*, 249:74–84, 2018.

[171] Borislav Iordanov. Hypergraphdb: a generalized graph database. In *Web-Age Information Management: WAIM 2010 International Workshops: IWGD 2010, XMLDM 2010, WCMT 2010, Jiuzhaigou Valley, China, July 15-17, 2010 Revised Selected Papers 11*, pages 25–36. Springer, 2010.

[172] Trevor Jack. On the complexity of properties of partial bijection semigroups. 2021.

[173] Chiranjibe Jana, Tapan Senapati, Monoranjan Bhowmik, and Madhumangal Pal. On intuitionistic fuzzy g-subalgebras of g-algebras. *Fuzzy Information and Engineering*, 7(2):195–209, 2015.

[174] Mohammad Ali Javidian. *Properties, Learning Algorithms, and Applications of Chain Graphs and Bayesian Hypergraphs*. PhD thesis, University of South Carolina, 2019.

[175] Mohammad Ali Javidian, Zhiyu Wang, Linyuan Lu, and Marco Valtorta. On a hypergraph probabilistic graphical model. *Annals of Mathematics and Artificial Intelligence*, 88:1003–1033, 2020.

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

[177] George A Jeffrey and Wolfram Saenger. *Hydrogen bonding in biological structures*. Springer Science & Business Media, 2012.

[178] Harold Jeffreys. *The theory of probability*. OuP Oxford, 1998.

[179] Ping Jian, Keming Chen, and Chenwei Zhang. A hypergraph-based context-sensitive representation technique for vhr remote-sensing image change detection. *International Journal of Remote Sensing*, 37:1814 – 1825, 2016.

[180] Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Hierarchical generation of molecular graphs using structural motifs. In *International conference on machine learning*, pages 4839–4848. PMLR, 2020.

[181] Bibin K Jose and Zsolt Tuza. Hypergraph domination and strong independence. *Applicable Analysis and Discrete Mathematics*, 3(2):347–358, 2009.

[182] Salim Jouili and Salvatore Tabbone. Hypergraph-based image retrieval for graph-based representation. *Pattern Recognit.*, 45:4054–4068, 2012.

[183] Young Bae Jun, Kul Hur, and Kyoung Ja Lee. Hyperfuzzy subalgebras of bck/bci-algebras. *Annals of Fuzzy Mathematics and Informatics*, 2017.

[184] Jinhong Jung, Jaemin Yoo, and U Kang. Signed graph diffusion network. *ArXiv*, abs/2012.14191, 2020.

[185] Hiroshi Kajino. Molecular hypergraph grammar with its application to molecular optimization. In *International Conference on Machine Learning*, pages 3183–3191. PMLR, 2019.

[186] Vasantha Kandasamy, K Ilanthenral, and Florentin Smarandache. *Neutrosophic graphs: a new dimension to graph theory*. Infinite Study, 2015.

[187] WB Vasantha Kandasamy, K Ilanthenral, and Florentin Smarandache. *Plithogenic Graphs*. Infinite Study, 2020.

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

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

[190] M. G. Karunambigai, R. Parvathi, and R. Buvaneswari. Arc in intuitionistic fuzzy graphs. *Notes on Intuitionistic Fuzzy Sets*, 17:37–47, 2011.

[191] Dimitris Kavvadias and Elias Stavropoulos. An efficient algorithm for the transversal hypergraph generation. *Journal of Graph Algorithms and Applications*, 9(2):239–264, 2005.

[192] Seyed Mehran Kazemi, Rishab Goel, Kshitij Jain, Ivan Kobyzev, Akshay Sethi, Peter Forsyth, and Pascal Poupart. Representation learning for dynamic graphs: A survey. *Journal of Machine Learning Research*, 21(70):1–73, 2020.

[193] Muhammad Hadir Khan, Bugra Onal, Eren Dogan, and Matthew R. Guthaus. Vlsi hypergraph partitioning with deep learning. *ArXiv*, abs/2409.01387, 2024.

[194] Lemont B Kier. A shape index from molecular graphs. *Quantitative Structure-Activity Relationships*, 4(3):109–116, 1985.

[195] Johannes Kobler, Uwe Schöning, and Jacobo Torán. *The graph isomorphism problem: its structural complexity*. Springer Science & Business Media, 2012.

[196] Daiki Koge, Naoaki Ono, Ming Huang, Md Altaf-Ul-Amin, and Shigehiko Kanaya. Embedding of molecular structure using molecular hypergraph variational autoencoder with metric learning. *Molecular informatics*, 40(2):2000203, 2021.

[197] Daphane Koller. Probabilistic graphical models: Principles and techniques, 2009.

[198] Elena V Konstantinova and Vladimir A Skorobogatov. Molecular hypergraphs: The new representation of nonclassical molecular structures with polycentric delocalized bonds. *Journal of chemical information and computer sciences*, 35(3):472–478, 1995.

[199] Elena V Konstantinova and Vladimir A Skorobogatov. Application of hypergraph theory in chemistry. *Discrete Mathematics*, 235(1-3):365–383, 2001.

[200] Alexandr Kostochka, Dhruv Mubayi, and Jacques Verstraete. Hypergraph ramsey numbers: triangles versus cliques. *Journal of Combinatorial Theory, Series A*, 120(7):1491–1507, 2013.

[201] Alexandr Kostochka, Dhruv Mubayi, and Jacques Verstraëte. On independent sets in hypergraphs. *Random Structures & Algorithms*, 44(2):224–239, 2014.

[202] Egor Kraev and Mark Harley. Probabilistic hypergraph grammars for efficient molecular optimization. *arXiv preprint arXiv:1906.01845*, 2019.

[203] Christian Krause and Holger Giese. Probabilistic graph transformation systems. In *International Conference on Graph Transformation*, pages 311–325. Springer, 2012.

[204] Suresh Kumar. Different types of graph used in network analysis. *International Journal of Mathematics Trends and Technology-IJMTT*, 66, 2020.

[205] Sergei K Lando. 7.4 multiplicative sequences. *Lectures on generating functions. AMS*.

[206] Geon Lee and Kijung Shin. Thyme+: Temporal hypergraph motifs and fast algorithms for exact counting. *2021 IEEE International Conference on Data Mining (ICDM)*, pages 310–319, 2021.

[207] Lung-Hao Lee and Yi Lu. Multiple embeddings enhanced multi-graph neural networks for chinese healthcare named entity recognition. *IEEE Journal of Biomedical and Health Informatics*, 25(7):2801–2810, 2021.

[208] Azriel Levy. *Basic set theory*. Courier Corporation, 2012.

[209] Gilbert N Lewis. The atom and the molecule. *Journal of the American Chemical Society*, 38(4):762–785, 1916.

[210] Kunhao Li, Zhenhua Huang, and Zhaohong Jia. Rahg: A role-aware hypergraph neural network for node classification in graphs. *IEEE Transactions on Network Science and Engineering*, 10:2098–2108, 2023.

[211] Shaobo Li, Matthew J Schneider, Yan Yu, and Sachin Gupta. Reidentification risk in panel data: Protecting for k-anonymity. *Information Systems Research*, 34(3):1066–1088, 2023.

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

[213] Liting Liu, Wenzheng Zhang, Jie Liu, Wenxuan Shi, and Yalou Huang. Learning multi-graph neural network for data-driven job skill prediction. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2021.

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

[215] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35, 2023.

[216] Xin Liu and Tsuyoshi Murata. Detecting communities in tripartite hypergraphs. *arXiv preprint arXiv:1011.1043*, 2010.

[217] Xueliang Liu, Meng Wang, Bao-Cai Yin, Benoit Huet, and Xuelong Li. Event-based media enrichment using an adaptive probabilistic hypergraph model. *IEEE Transactions on Cybernetics*, 45(11):2461–2471, 2014.

[218] Xuemei Liu, Jixiang Meng, and Yingzhi Tian. On forcibly k-edge-connected and forcibly super edge-connected uniform hypergraphic sequences. *The Journal of Supercomputing*, 79(14):15980–15996, 2023.

[219] Zijian Liu, Yang Luo, Xitong Pu, Geyong Min, and Chunbo Luo. A multi-modal hypergraph neural network via parametric filtering and feature sampling. *IEEE Transactions on Big Data*, 9:1365–1379, 2023.

[220] Anand Louis. Hypergraph markov operators, eigenvalues and approximation algorithms. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, pages 713–722, 2015.

[221] Caimei Lu, Xin Chen, and EK Park. Exploit the tripartite network of social tagging for web clustering. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 1545–1548, 2009.

[222] Ruitao Lu, Wanying Xu, Yongbin Zheng, and Xinsheng Huang. Visual tracking via probabilistic hypergraph ranking. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(4):866–879, 2015.

[223] Michael Luby. A simple parallel algorithm for the maximal independent set problem. In *Proceedings of the seventeenth annual ACM symposium on Theory of computing*, pages 1–10, 1985.

[224] Eugene M Luks. Hypergraph isomorphism and structural equivalence of boolean functions. In *Proceedings of the thirty-first annual ACM symposium on Theory of computing*, pages 652–658, 1999.

[225] Anam Luqman, Muhammad Akram, Ahmad N. Al-kenani, and José Carlos Rodriguez Alcantud. A study on hypergraph representations of complex fuzzy information. *Symmetry*, 11:1381, 2019.

[226] Anam Luqman, Muhammad Akram, and Florentin Smarandache. Complex neutrosophic hypergraphs: new social network models. *Algorithms*, 12(11):234, 2019.

[227] Edita Macajova, Andre Raspaud, and Martin Skoviera. The chromatic number of a signed graph. *Electron. J. Comb.*, 23:1, 2014.

[228] Pradip Kumar Maji, Ranjit Biswas, and A Ranjan Roy. Soft set theory. *Computers & mathematics with applications*, 45(4-5):555–562, 2003.

[229] Rama Mallick and Surapati Pramanik. Pentapartitioned neutrosophic set and its properties. *Neutrosophic Sets and Systems*, 35:49, 2020.

[230] Vipul Mann and Venkat Venkatasubramanian. Ai-driven hypergraph network of organic chemistry: network statistics and applications in reaction classification. *Reaction Chemistry & Engineering*, 2022.

[231] Christopher D Manning. *Foundations of statistical natural language processing*. The MIT Press, 1999.

[232] Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60, 2014.

[233] David E Manolopoulos and Patrick W Fowler. Molecular graphs, point groups, and fullerenes. *The Journal of chemical physics*, 96(10):7603–7614, 1992.

[234] J Abderramán Marrero, Juan Núñez Valdés, and María Trinidad Villar. Associating hub-directed multigraphs to arrowhead matrices. *Mathematical Methods in the Applied Sciences*, 41(6):2360–2369, 2018.

[235] Brendan D McKay et al. Practical graph isomorphism. 1981.

[236] Brendan D McKay and Adolfo Piperno. Practical graph isomorphism, ii. *Journal of symbolic computation*, 60:94–112, 2014.

[237] Brendan D McKay and Nicholas C Wormald. Asymptotic enumeration by degree sequence of graphs with degrees o (n 1/2). *Combinatorica*, 11(4):369–382, 1991.

[238] Brendan D McKay and Nicholas C Wormald. The degree sequence of a random graph. i. the models. *Random Structures & Algorithms*, 11(2):97–117, 1997.

[239] Syed M Meesum. An efficiently recognisable subset of hypergraphic sequences. In *International Computing and Combinatorics Conference*, pages 391–402. Springer, 2018.

[240] Martin Middendorf and Vadim G Timkovsky. Transversal graphs for partially ordered sets: Sequencing, merging and scheduling problems. *Journal of Combinatorial Optimization*, 3:417–435, 1999.

[241] Michael Molloy and Bruce Reed. A critical point for random graphs with a given degree sequence. *Random structures & algorithms*, 6(2-3):161–180, 1995.

[242] Dmitriy Molodtsov. Soft set theory-first results. *Computers & mathematics with applications*, 37(4-5):19–31, 1999.

[243] John N Mordeson and Sunil Mathew. *Advanced topics in fuzzy graph theory*, volume 375. Springer, 2019.

[244] Dhruv Mubayi and Andrew Suk. Off-diagonal hypergraph ramsey numbers. *Journal of Combinatorial Theory, Series B*, 125:168–177, 2017.

[245] Dhruv Mubayi and Andrew Suk. New lower bounds for hypergraph ramsey numbers. *Bulletin of the London Mathematical Society*, 50(2):189–201, 2018.

[246] M Myvizhi, Ahmed M Ali, Ahmed Abdelhafeez, and Haitham Rizk Fadlallah. *MADM Strategy Application of Bipolar Single Valued Heptapartitioned Neutrosophic Set*. Infinite Study, 2023.

[247] Brendan Nagle, Vojtech Rodl, and Mathias Schacht. The counting lemma for regular k-uniform hypergraphs. *Random Structures & Algorithms*, 28(2):113–179, 2006.

[248] Rajko Nenadov. Probabilistic hypergraph containers. *Israel Journal of Mathematics*, 261(2):879–897, 2024.

[249] Daniel Neuen. Hypergraph isomorphism for groups with restricted composition factors. *ACM Transactions on Algorithms (TALG)*, 18(3):1–50, 2022.

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

[251] Kung-Jui Pai, Shyue-Ming Tang, Jou-Ming Chang, and Jinn-Shyong Yang. Completely independent spanning trees on complete graphs, complete bipartite graphs and complete tripartite graphs. In *Advances in Intelligent Systems and Applications-Volume 1: Proceedings of the International Computer Symposium ICS 2012 Held at Hualien, Taiwan, December 12–14, 2012*, pages 107–113. Springer, 2013.

[252] Madhumangal Pal, Sovan Samanta, and Ganesh Ghorai. *Modern trends in fuzzy graph theory*. Springer, 2020.

[253] Junyoung Park, Fangying Chen, and Jinkyoo Park. A molecular hypergraph convolutional network with functional group information. 2021.

[254] Vasile Patrascu. Penta and hexa valued representation of neutrosophic information. *arXiv preprint arXiv:1603.03729*, 2016.

[255] Zdzis law Pawlak. Rough sets. *International journal of computer & information sciences*, 11:341–356, 1982.

[256] Zdzislaw Pawlak. Rough set theory and its applications to data analysis. *Cybernetics & Systems*, 29(7):661–688, 1998.

[257] Zdzislaw Pawlak, Lech Polkowski, and Andrzej Skowron. Rough set theory. *KI*, 15(3):38–39, 2001.

[258] Zdzislaw Pawlak, S. K. Michael Wong, Wojciech Ziarko, et al. Rough sets: probabilistic versus deterministic approach. *International Journal of Man-Machine Studies*, 29(1):81–95, 1988.

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

[260] Aymeric Perchant and Isabelle Bloch. Fuzzy morphisms between graphs. *Fuzzy Sets and Systems*, 128(2):149–168, 2002.

[261] JC Phillips. Ionicity of the chemical bond in crystals. *Reviews of modern physics*, 42(3):317, 1970.

[262] Demitri Joel Plessas. *The categories of graphs*. University of Montana, 2011.

[263] Mariya Popova, Mykhailo Shvets, Junier Oliva, and Olexandr Isayev. Molecularrnn: Generating realistic molecular graphs with optimized properties. *arXiv preprint arXiv:1905.13372*, 2019.

[264] Daniele Pretolani. Finding hypernetworks in directed hypergraphs. *European Journal of Operational Research*, 230(2):226–230, 2013.

[265] P.Subbaraj, S.Saravanasankar, and S.Anand. Combinatorial optimization in vlsi hypergraph partitioning using taguchi methods. 2017.

[266] Shio Gai Quek, Ganeshsree Selvachandran, D Ajay, P Chellamani, David Taniar, Hamido Fujita, Phet Duong, Le Hoang Son, and Nguyen Long Giang. New concepts of pentapartitioned neutrosophic graphs and applications for determining safest paths and towns in response to covid-19. *Computational and Applied Mathematics*, 41(4):151, 2022.

[267] Alain Quilliot. On the helly property working as a compactness criterion on graphs. *Journal of Combinatorial Theory, Series A*, 40(1):186–193, 1985.

[268] Stanislaw Radziszowski. Small ramsey numbers. *The electronic journal of combinatorics*, pages DS1–Jan, 2012.

[269] M Ramya, Sandesh Murali, and R.Radha. Bipolar quadripartitioned neutrosophic soft set. 2022.

[270] Milan Randić. On history of the randić index and emerging hostility toward chemical graph theory. *MATCH Commun. Math. Comput. Chem*, 59(1):5–124, 2008.

[271] Yongsheng Rao, Saeed Kosari, and Zehui Shao. Certain properties of vague graphs with a novel application. *Mathematics*, 8(10):1647, 2020.

[272] Hossein Rashmanlou and Rajab Ali Borzooei. Vague graphs with application. *Journal of Intelligent & Fuzzy Systems*, 30(6):3291–3299, 2016.

[273] Victor J. Rayward-Smith. Hypergrammars: an extension of macrogrammars. *Journal of Computer and System Sciences*, 14(1):130–149, 1977.

[274] Ronald C Read and Derek G Corneil. The graph isomorphism disease. *Journal of graph theory*, 1(4):339–363, 1977.

[275] Lisbeth J Reales-Chacón, María E Lucena de Ustáriz, Francisco J Ustáriz-Fajardo, Andrea C Peñafiel Luna, Gabriela J Bonilla-Ayala, Pablo Djabayan-Djibeyan, José L Erazo-Parra, and Mónica A Valdiviezo-Maygua. Study of the efficacy of neural mobilizations to improve sensory and functional responses of lower extremities in older adults with diabetic peripheral neuropathy using plithogenic n-superhypergraphs. *Neutrosophic Sets and Systems*, 74:1–12, 2024.

[276] Anna M. Ritz, Brendan Avent, Aditya Pratapa, and T. M. Murali. The hypergraph algorithms package. *F1000Research*, 7, 2018.

[277] David W Roberts. Analysis of forest succession with fuzzy graph theory. *Ecological Modelling*, 45(4):261–274, 1989.

[278] Vojtech Rodl and Jozef Skokan. Regularity lemma for k-uniform hypergraphs. *Random Structures & Algorithms*, 25(1):1–42, 2004.

[279] Judith Roitman. *Introduction to modern set theory*, volume 8. John Wiley & Sons, 1990.

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

[281] Sheldon M Ross. *Introduction to probability models*. Academic press, 2014.

[282] Kayvan Sadeghi. Stable mixed graphs. 2013.

[283] Hasmik Sahakyan. On the set of simple hypergraph degree sequences. *Applied Mathematical Sciences*, 9(5):243–253, 2015.

[284] Rıdvan Şahin. An approach to neutrosophic graph theory with applications. *Soft Computing*, 23(2):569–581, 2019.

[285] Abdullah Ali Salamai. A superhypersoft framework for comprehensive risk assessment in energy projects. *Neutrosophic Sets and Systems*, 77:614–624, 2025.

[286] Sovan Samanta, Madhumangal Pal, Hossein Rashmanlou, and Rajab Ali Borzooei. Vague graphs and strengths. *Journal of Intelligent & Fuzzy Systems*, 30(6):3675–3680, 2016.

[287] R Sanderson. *Chemical bonds and bonds energy*, volume 21. Elsevier, 2012.

[288] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE transactions on neural networks*, 20(1):61–80, 2008.

[289] Thomas Schank and Dorothea Wagner. Approximating clustering coefficient and transitivity. *Journal of Graph Algorithms and Applications*, 9(2):265–275, 2005.

[290] Christoph Schulz, Arlind Nocaj, Jochen Goertler, Oliver Deussen, Ulrik Brandes, and Daniel Weiskopf. Probabilistic graph layout for uncertain network visualization. *IEEE transactions on visualization and computer graphics*, 23(1):531–540, 2016.

[291] Ming-Sheng Shang, Zi-Ke Zhang, Tao Zhou, and Yi-Cheng Zhang. Collaborative filtering with diffusion-based similarity on tripartite graphs. *Physica A: Statistical Mechanics and its Applications*, 389(6):1259–1264, 2010.

[292] C.-J. Richard Shi and Janusz A. Brzozowski. A characterization of signed hypergraphs and its applications to vlsi via minimization and logic synthesis. *Discret. Appl. Math.*, 90:223–243, 1999.

[293] Chuan-Jin Shi. A signed hypergraph model of the constrained via minimization problem. *Microelectronics Journal*, 23:533–542, 1992.

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

[295] Xiaolong Shi, Saeed Kosari, Hossein Rashmanlou, Said Broumi, and S Satham Hussain. Properties of interval-valued quadripartitioned neutrosophic graphs with real-life application. *Journal of Intelligent & Fuzzy Systems*, 44(5):7683–7697, 2023.

[296] Julian Shun. Practical parallel hypergraph algorithms. *Proceedings of the 25th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, 2020.

[297] Uriel Singer, Ido Guy, and Kira Radinsky. Node embedding over temporal graphs. *arXiv preprint arXiv:1903.08889*, 2019.

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

[299] Prem Kumar Singh et al. Dark data analysis using intuitionistic plithogenic graphs. *International Journal of Neutrosophic Sciences*, 16(2):80–100, 2021.

[300] Mitali Sinha. Pair wise swapping based hypergraph partitioning algorithms for vlsi design. *International Journal of Recent Technology and Engineering*, 2019.

[301] F. Smarandache. Introduction to superhyperalgebra and neutrosophic superhyperalgebra. *Journal of Algebraic Hyperstructures and Logical Algebras*, 2022.

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

[303] Florentin Smarandache. *Definitions derived from neutrosophics*. Infinite Study, 2003.

[304] Florentin Smarandache. Neutrosophic set-a generalization of the intuitionistic fuzzy set. *International journal of pure and applied mathematics*, 24(3):287, 2005.

[305] Florentin Smarandache. Extension of soft set to hypersoft set, and then to plithogenic hypersoft set. *Neutrosophic sets and systems*, 22(1):168–170, 2018.

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

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

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

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

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

[311] Florentin Smarandache. *The SuperHyperFunction and the Neutrosophic SuperHyperFunction (revisited again)*, volume 3. Infinite Study, 2022.

[312] Florentin Smarandache. Decision making based on valued fuzzy superhypergraphs. 2023.

[313] Florentin Smarandache. Foundation of the superhypersoft set and the fuzzy extension superhypersoft set: A new vision. *Neutrosophic Systems with Applications*, 11:48–51, 2023.

[314] Florentin Smarandache. *New types of soft sets "hypersoft set, indetermsoft set, indetermhypersoft set, and treesoft set": an improved version*. Infinite Study, 2023.

[315] Florentin Smarandache. *SuperHyperFunction, SuperHyperStructure, Neutrosophic SuperHyperFunction and Neutrosophic SuperHyperStructure: Current understanding and future directions*. Infinite Study, 2023.

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

[317] Florentin Smarandache and NM Gallup. Generalization of the intuitionistic fuzzy set to the neutrosophic set. In *International Conference on Granular Computing*, pages 8–42. Citeseer, 2006.

[318] Florentin Smarandache and Nivetha Martin. *Plithogenic n-super hypergraph in novel multi-attribute decision making*. Infinite Study, 2020.

[319] Florentin Smarandache, A Saranya, A Kalavathi, and S Krishnaprakash. Neutrosophic superhypersoft sets. *Neutrosophic Sets and Systems*, 77:41–53, 2025.

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

[321] Yuzhi Song, Hailiang Ye, Ming Li, and Feilong Cao. Deep multi-graph neural networks with attention fusion for recommendation. *Expert Systems with Applications*, 191:116240, 2022.

[322] Andreas Spitz, Dennis Aumiller, Bálint Soproni, and Michael Gertz. A versatile hypergraph model for document collections. In *Proceedings of the 32nd International Conference on Scientific and Statistical Database Management*, pages 1–12, 2020.

[323] Elias C Stavropoulos, Vassilios S Verykios, and Vasileios Kagklis. A transversal hypergraph approach for the frequent itemset hiding problem. *Knowledge and information systems*, 47:625–645, 2016.

[324] A Sudha and P Sundararajan. Robust fuzzy graph. *Ratio Mathematica*, 46, 2023.

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

[326] Zoltán Szigeti. Packing mixed hyperarborescences. *Discret. Optim.*, 50:100811, 2023.

[327] Dimitrios Tasis, Nikos Tagmatarchis, Alberto Bianco, and Maurizio Prato. Chemistry of carbon nanotubes. *Chemical reviews*, 106(3):1105–1136, 2006.

[328] Jacobo Torán. On the hardness of graph isomorphism. *SIAM Journal on Computing*, 33(5):1093–1108, 2004.

[329] Nenad Trinajstic. *Chemical graph theory*. CRC press, 2018.

[330] Ke Tu, Peng Cui, Xiao Wang, Fei Wang, and Wenwu Zhu. Structural deep embedding for hyper-networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[331] Vakkas Ulucay and Memet Sahin. Intuitionistic fuzzy soft expert graphs with application. *Uncertainty discourse and applications*, 1(1):1–10, 2024.

[332] Paola Valdivia, Paolo Buono, Catherine Plaisant, Nicole Dufournaud, and Jean-Daniel Fekete. Analyzing dynamic hypergraphs with parallel aggregated ordered hypergraph visualization. *IEEE transactions on visualization and computer graphics*, 27(1):1–13, 2019.

[333] Hadrien Van Lierde and Tommy WS Chow. Query-oriented text summarization based on hypergraph transversals. *Information Processing & Management*, 56(4):1317–1338, 2019.

[334] B Vembu and S Loghambal. Pseudo-graph neural networks on ordinary differential equations. *Journal of Computational Mathematica*, 6(1):117–123, 2022.

[335] Stephan Wagner and Hua Wang. *Introduction to chemical graph theory*. Chapman and Hall/CRC, 2018.

[336] Aiqin Wang, Jun Li, and Tao Zhang. Heterogeneous single-atom catalysis. *Nature Reviews Chemistry*, 2(6):65–81, 2018.

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

[338] Shuai Wang, Jiayi Shen, Athanasios Efthymiou, Stevan Rudinac, Monika Kackovic, Nachoem Wijnberg, and Marcel Worring. Prototype-enhanced hypergraph learning for heterogeneous information networks. In *International Conference on Multimedia Modeling*, pages 462–476. Springer, 2024.

[339] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu. Heterogeneous graph attention network. In *The world wide web conference*, pages 2022–2032, 2019.

[340] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (tog)*, 38(5):1–12, 2019.

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

[342] Zhiyu Wang, Mohammad Ali Javidian, Linyuan Lu, and Marco Valtorta. The causal interpretations of bayesian hypergraphs. 2019.

[343] Zuhao Wang, Shaohui Chi, Ma Luo, Yuqin Yang, and Min Huang. Development of an instrument to evaluate high school students' chemical symbol representation abilities. *Chemistry Education Research and Practice*, 18(4):875–892, 2017.

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

[345] Zheng Wenping, Liu Meilin, and Liang Jiye. Hypergraphs: Concepts, applications and analysis. In *2022 IEEE 13th International Symposium on Parallel Architectures, Algorithms and Programming (PAAP)*, pages 1–6. IEEE, 2022.

[346] Douglas Brent West et al. *Introduction to graph theory*, volume 2. Prentice hall Upper Saddle River, 2001.

[347] Richard M. Wilson. Signed hypergraph designs and diagonal forms for some incidence matrices. *Designs, Codes and Cryptography*, 17:289–297, 1999.

[348] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45, 2020.

[349] Hongjie Wu, Weizhong Lu, Meiling Qian, Yu Zhang, Yijie Ding, Jiawei Shen, Xiaoyi Chen, Haiou Li, and Qiming Fu. Identification of membrane protein types based using hypergraph neural network. *Current Bioinformatics*, 2023.

[350] Jinming Wu, Qi Qi, Jingyu Wang, Haifeng Sun, Zhikang Wu, Zirui Zhuang, and Jianxin Liao. Not only pairwise relationships: Fine-grained relational modeling for multivariate time series forecasting. In *IJCAI*, pages 4416–4423, 2023.

[351] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1):4–24, 2020.

[352] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.

[353] Hongye Yang, Yuzhang Gu, Jianchao Zhu, Keli Hu, and Xiaolin Zhang. Pgcn-tca: Pseudo graph convolutional network with temporal and channel-wise attention for skeleton-based action recognition. *IEEE Access*, 8:10040–10047, 2020.

[354] Jian Yang and Joseph Y-T Leung. A generalization of the weighted set covering problem. *Naval Research Logistics (NRL)*, 52(2):142–149, 2005.

[355] Jeonghee Yi, Tetsuya Nasukawa, Razvan Bunescu, and Wayne Niblack. Sentiment analyzer: Extracting sentiments about a given topic using natural language processing techniques. In *Third IEEE international conference on data mining*, pages 427–434. IEEE, 2003.

[356] Nan Yin, Fuli Feng, Zhigang Luo, Xiang Zhang, Wenjie Wang, Xiao Luo, Chong Chen, and Xian-Sheng Hua. Dynamic hypergraph convolutional network. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pages 1621–1634. IEEE, 2022.

[357] Nan Yin, Li Shen, Huan Xiong, Bin Gu, Chong Chen, Xian-Sheng Hua, Siwei Liu, and Xiao Luo. Messages are never propagated alone: Collaborative hypergraph neural network for time-series forecasting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46:2333–2347, 2023.

[358] Jiaxuan You, Bowen Liu, Zhitao Ying, Vijay Pande, and Jure Leskovec. Graph convolutional policy network for goal-directed molecular graph generation. *Advances in neural information processing systems*, 31, 2018.

[359] Guihai Yu, Xiying Yuan, and Hui Qu. Signed k-uniform hypergraphs and tensors. *Linear Algebra and its Applications*, 2019.

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

[361] Lotfi A Zadeh. A fuzzy-set-theoretic interpretation of linguistic hedges. 1972.

[362] Lotfi A Zadeh. Fuzzy sets and their application to pattern classification and clustering analysis. In *Classification and clustering*, pages 251–299. Elsevier, 1977.

[363] Lotfi A Zadeh. Fuzzy sets versus probability. *Proceedings of the IEEE*, 68(3):421–421, 1980.

[364] Lotfi A Zadeh. Fuzzy sets and information granularity. In *Fuzzy sets, fuzzy logic, and fuzzy systems: selected papers by Lotfi A Zadeh*, pages 433–448. World Scientific, 1996.

[365] Lotfi A Zadeh. A note on prototype theory and fuzzy sets. In *Fuzzy sets, fuzzy logic, and fuzzy systems: Selected papers by Lotfi A Zadeh*, pages 587–593. World Scientific, 1996.

[366] Lotfi Asker Zadeh. Fuzzy sets as a basis for a theory of possibility. *Fuzzy sets and systems*, 1(1):3–28, 1978.

[367] Chengxi Zang and Fei Wang. Moflow: an invertible flow model for generating molecular graphs. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 617–626, 2020.

[368] Thomas Zaslavsky. How colorful the signed graph? *Discret. Math.*, 52:279–284, 1984.

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

[370] Lingling Zhang, Zhiwei Zhang, Guoren Wang, Ye Yuan, Shuai Zhao, and Jianliang Xu. Hyperiso: Efficiently searching subgraph containment in hypergraphs. *IEEE Transactions on Knowledge and Data Engineering*, 35(8):8112–8125, 2022.

[371] Zhongfu Zhang, Linzhong Liu, and Jianfang Wang. Adjacent strong edge coloring of graphs. *Applied mathematics letters*, 15(5):623–626, 2002.

[372] Zi-Ke Zhang and Chuang Liu. A hypergraph model of social tagging networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2010(10):P10005, 2010.

[373] Zi-Ke Zhang, Tao Zhou, and Yi-Cheng Zhang. Personalized recommendation via integrated diffusion on user–item–tag tripartite graphs. *Physica A: Statistical Mechanics and its Applications*, 389(1):179–186, 2010.

[374] Zitong Zhang, Ling-Feng Zhao, Junjie Wang, and Chunyu Wang. A hierarchical graph neural network framework for predicting protein-protein interaction modulators with functional group information and hypergraph structure. *IEEE Journal of Biomedical and Health Informatics*, 28:4295–4305, 2024.

[375] Zizhao Zhang, Haojie Lin, Yue Gao, and KLISS BNRist. Dynamic hypergraph structure learning. In *IJCAI*, pages 3162–3169, 2018.

[376] Hua Zhao, Zeshui Xu, Shousheng Liu, and Zhong Wang. Intuitionistic fuzzy mst clustering algorithms. *Computers & Industrial Engineering*, 62(4):1130–1140, 2012.

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

[378] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. *AI open*, 1:57–81, 2020.

[379] Peng Zhou, Zongqian Wu, Xiangxiang Zeng, Guoqiu Wen, Junbo Ma, and Xiaofeng Zhu. Totally dynamic hypergraph neural networks. In *International Joint Conference on Artificial Intelligence*, 2023.

[380] Linhong Zhu, Aram Galstyan, James Cheng, and Kristina Lerman. Tripartite graph clustering for dynamic sentiment analysis on social media. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*, pages 1531–1542, 2014.

[381] Wenwu Zhu, Xin Wang, and Peng Cui. Deep learning for learning graph representations. *Deep learning: concepts and architectures*, pages 169–210, 2020.