Article

Not peer-reviewed version

Tree-Vertex Graph: New Hierarcal Graph Class

Takaaki Fujita *

Posted Date: 29 December 2025

doi: 10.20944/preprints202512.2469.v1

Keywords: SuperHyperGraph; HyperGraph; Tree-Vertex Graph

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 29 December 2025 doi:10.20944/preprints202512.2469.v1

Article

Tree-Vertex Graph: New Hierarcal Graph Class

Takaaki Fujita

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

Abstract

A hypergraph generalizes an ordinary graph by allowing an edge to connect any nonempty subset of the vertex set. By iterating the powerset operation one step further, one obtains nested (higher-order) vertex objects and, consequently, a finite SuperHyperGraph whose vertices and edges may themselves be set-valued at multiple levels. Thus, many hierarchical graph structures exist in the literature. Moreover, not only in graph theory but also in broader fields—such as through concepts like Decision Trees and Tree Soft Sets—it is well known that tree structures are effective tools for representing hierarchical concepts. In this paper, we define a new class of graphs called Tree-Vertex Graphs. In this framework, a tree structure is imposed on the vertex set, and the edge set is defined in a manner consistent with the tree-structured vertex set. The tree structure therefore serves as a key concept for representing hierarchical graphs.

Keywords: SuperHyperGraph; HyperGraph; Tree-Vertex Graph

---

1. Preliminaries

This section introduces the notation and basic terminology.

1.1. SuperHyperGraphs

Graph theory studies vertices and edges, analyzing connectivity, structure, and algorithms for modeling relationships in mathematics, computer science, and applications [1]. A *hypergraph* generalizes an ordinary graph by allowing an edge to connect any nonempty subset of the vertex set. Hence, hypergraphs provide a direct language for modeling multiway interactions [2,3]. In particular, such structures have played an important role in recent years in methods such as neural networks [3–7].

By iterating the powerset operation one step further, one obtains *nested* (higher-order) vertex objects and, consequently, a finite *SuperHyperGraph* whose vertices and edges may themselves be set-valued at multiple levels [8,9]. Such hierarchical representations are useful, for instance, in molecular design, complex-network analysis, neural networks, and related applications [10–16]. Moreover, as related concepts, research has also been progressing on notions such as Directed SuperHyperGraphs [17,18] and MetaSuperHyperGraphs [19].

Throughout, the index n in $\mathrm{PS}_n(\cdot)$ and in an n-SuperHyperGraph is always taken to be a nonnegative integer.

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

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

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

Definition 2 (Powerset)**.** *(see [20–22]) For a set S, the* powerset *of S is*

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

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

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 29 December 2025 doi:10.20944/preprints202512.2469.v1

Definition 3 (Hypergraph [23,24])**.** *A* hypergraph *is a pair $H = (V, E)$ such that:*

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

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

Definition 4 (Iterated powerset and flattening)**.** *Let V_0 be a finite nonempty set. Define $\mathrm{PS}^0(V_0) := V_0$ and $\mathrm{PS}^{k+1}(V_0) := \mathrm{PS}\big(\mathrm{PS}^k(V_0)\big)$ for $k \geq 0$. For each $k \geq 0$, define the flattening map*

$$\mathrm{Flat}_k : \mathrm{PS}^k(V_0) \setminus \{\varnothing\} \longrightarrow \mathrm{PS}(V_0) \setminus \{\varnothing\}$$

recursively by

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

Definition 5 (n-SuperHyperGraph)**.** *(see [25]) Let V_0 be a finite, nonempty base set. Define*

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

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

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

satisfying

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

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

As a reference, a high-level comparison of Graphs, Hypergraphs, and SuperHyperGraphs is presented in Table 1.

Table 1. High-level comparison of Graphs, Hypergraphs, and SuperHyperGraphs.

Object	Vertex objects	Edge objects (what an edge can connect)
Graph $G = (V, E)$	Atoms: V is a finite set of vertices.	Edges are pairs of vertices (undirected) or ordered pairs (directed): $E \subseteq \binom{V}{2}$ (undirected) or $E \subseteq V \times V$ (directed).
Hypergraph $H = (V, E)$	Atoms: V is a finite set of vertices.	Hyperedges are nonempty subsets of vertices: $E \subseteq \mathrm{PS}(V) \setminus \{\varnothing\}$. An edge may connect any number of vertices.
n-SuperHyperGraph $\mathrm{SHG}^{(n)} = (V, E)$	Nested objects: $V \subseteq \mathrm{PS}^n(V_0)$, i.e., vertices may be sets-of-sets (iterated powersets).	Superedges are nonempty subsets of the (possibly nested) vertex set: $E \subseteq \mathrm{PS}(V) \setminus \{\varnothing\}$. Thus edges connect collections of (nested) vertices.

1.2. Rooted Tree

A rooted tree is a connected acyclic graph with a distinguished root, where each non-root node has exactly one parent.

Definition 6 (Rooted tree)**.** *A* rooted tree *is a triple $T = (\mathcal{N}, \mathcal{F}, r)$ where $\mathcal{N}$ is a finite node set, $r \in \mathcal{N}$ is the root, and $\mathcal{F} \subseteq \mathcal{N} \times \mathcal{N}$ is a set of* parent–child *arcs such that every node $u \in \mathcal{N} \setminus \{r\}$ has a unique parent and the underlying undirected graph is connected and acyclic. For $u \in \mathcal{N}$, let $\mathrm{Ch}(u)$ denote its children and let $\mathrm{Leaf}(T) \subseteq \mathcal{N}$ denote the set of leaves.*

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 29 December 2025 doi:10.20944/preprints202512.2469.v1

2. Main Results

This section presents the main results of this paper.

2.1. Depth-n Tree-Vertex Graph

A depth-n Tree-Vertex Graph is a rooted tree of depth n whose nodes represent hierarchical units via nested iterated-powerset labels, with edges only within each level.

Definition 7 (Depth-n Tree-Vertex Graph with level edges). *Let V_0 be a finite, nonempty set of* base vertices *and let $n \in \mathbb{N}$. A* depth-n Tree-Vertex Graph (TVG) *on V_0 is a quadruple*

$$\mathrm{TVG}^{(n)} = \big(V_0,\ T,\ \eta,\ \{E^{(k)}\}_{k=0}^{n}\big),$$

where:

(i) $T = (\mathcal{N}, \mathcal{F}, r)$ *is a rooted tree whose leaves have depth* 0 *and whose root has depth n. For each $k \in \{0, \ldots, n\}$ define the level set*

$$\mathcal{N}_k := \{u \in \mathcal{N} \mid \mathrm{depth}(u) = k\}.$$

(ii) η *is a* nested labeling *(level-typed label map)*

$$\eta : \mathcal{N} \longrightarrow \bigcup_{k=0}^{n} \mathrm{PS}^k(V_0) \setminus \{\varnothing\}$$

satisfying:

(a) (Level-typing) *For every $u \in \mathcal{N}_k$, one has*

$$\eta(u) \in \mathrm{PS}^k(V_0) \setminus \{\varnothing\}.$$

(b) (Leaf grounding) *The restriction $\eta|_{\mathcal{N}_0} : \mathcal{N}_0 \to V_0$ is a bijection (so each leaf represents exactly one base vertex).*

(c) (Recursive nesting along the tree) *For every internal node $u \in \mathcal{N}_k$ with $k \geq 1$,*

$$\eta(u) = \{\eta(v) \mid v \in \mathrm{Ch}(u)\},$$

so the label of u is literally the set of its children's labels (hence lives in the next iterated powerset).

(iii) The support *(flattened base-vertex set) of a tree-vertex $u \in \mathcal{N}_k$ is defined by*

$$\lambda(u) := \mathrm{Flat}_k(\eta(u)) \subseteq V_0,$$

where Flat_k *is from Definition 4.*

(iv) For each level $k \in \{0, \ldots, n\}$, $E^{(k)}$ is a set of level-k graph edges*:*

$$E^{(k)} \subseteq \big\{\{u, v\} \subseteq \mathcal{N}_k \mid u \neq v\big\}.$$

An edge $\{u, v\} \in E^{(k)}$ encodes a binary relation within the same abstraction depth k *between the hierarchical units represented by u and v.*

Remark 1 (Why this matches the n-th powerset depth). *In Definition 7, the constraint*

$$\eta(u) \in \mathrm{PS}^{\mathrm{depth}(u)}(V_0)$$

is enforced by construction: leaves satisfy $\eta(\ell) \in V_0 = \mathrm{PS}^0(V_0)$, and each upward step applies one powerset operation via $\eta(u) = \{\eta(v) \mid v \in \mathrm{Ch}(u)\}$. Hence the tree depth is literally *the iterated-powerset depth. The flattened support $\lambda(u) = \mathrm{Flat}_{\mathrm{depth}(u)}(\eta(u))$ recovers the base vertices contained in u.*

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 29 December 2025 doi:10.20944/preprints202512.2469.v1

Remark 2 (Option: level-wise hyperedges instead of edges)**.** *If one wants* multiway relations *at each level, replace each* $E^{(k)}$ *by a hyperedge family*

$$\mathcal{E}^{(k)} \subseteq \mathrm{PS}(\mathcal{N}_k) \setminus \{\emptyset\},$$

so that each hyperedge $e \in \mathcal{E}^{(k)}$ *relates an arbitrary finite subset of nodes at level* k*. This yields a* level-hypergraph TVG *while retaining the same nested labeling* η*.*

A comparison of an n-SuperHyperGraph and a depth-n Tree-Vertex Graph is presented in Table 2.

Table 2. Concise comparison of an n-SuperHyperGraph and a depth-n Tree-Vertex Graph.

Aspect	**n-SuperHyperGraph $\mathrm{SHG}^{(n)} = (V, E)$**	**Depth-n Tree-Vertex Graph $\mathrm{TVG}^{(n)} = (V_0, T, \eta, \{E^{(k)}\}_{k=0}^{n})$**
Base objects	A finite base set V_0 is given, but vertices are *chosen* as $V \subseteq \mathrm{PS}^n(V_0)$.	A finite base set V_0 is given; leaves of T (level 0) are in bijection with V_0.
Vertex representation	Vertices are *uniform-depth* set-objects: $v \in \mathrm{PS}^n(V_0)$.	Vertices are *tree-vertices* $u \in \mathcal{N}$, each typed by its depth k and labeled by $\eta(u) \in \mathrm{PS}^k(V_0)$.
Hierarchy mechanism	Hierarchy is implicit in the iterated powerset level n; there is no canonical parent–child structure among vertices.	Hierarchy is explicit: $T = (\mathcal{N}, \mathcal{F}, r)$ provides parent–child arcs, and labels satisfy $\eta(u) = \{\eta(v) \mid v \in \mathrm{Ch}(u)\}$.
Depth / levels	A single global level n for vertices (although edges are subsets of V).	Multiple levels $k = 0, \ldots, n$ present simultaneously; $\mathcal{N}_k = \{u \mid \mathrm{depth}(u) = k\}$.
Flattened support in V_0	Not intrinsic; one may apply a flattening map on $\mathrm{PS}^n(V_0)$ if desired.	Built-in: each tree-vertex has support $\lambda(u) = \mathrm{Flat}_{\mathrm{depth}(u)}(\eta(u)) \subseteq V_0$.
Edges	Hyperedges are nonempty subsets of V: $E \subseteq \mathrm{PS}(V) \setminus \{\emptyset\}$ (multiway relations between supervertices).	Level edges are *binary* within each level: $E^{(k)} \subseteq \{\{u, v\} \subseteq \mathcal{N}_k \mid u \neq v\}$.
Granularity of relations	Relations are expressed as set-valued edges (hyperedges) among set-valued vertices.	Relations are expressed as graph edges among hierarchical units at the same abstraction level.
Typical modeling intent	Nested (higher-order) vertex objects with hyperedges capturing multiway interactions at the chosen level.	A hierarchical decomposition of V_0 via a tree, with additional same-level relational structure among the resulting clusters at each depth.

Specific examples are given below.

Example 1 (A depth-2 TVG on four base vertices)**.** *Let*

$$V_0 = \{a, b, c, d\}, \qquad n = 2.$$

We define $\mathrm{TVG}^{(2)} = (V_0, T, \eta, \{E^{(k)}\}_{k=0}^{2})$ *as follows.*

(i) ***Rooted tree.*** *Let the node set be*

$$\mathcal{N} = \{\ell_a, \ell_b, \ell_c, \ell_d,\ u_1, u_2,\ r\}.$$

Declare r *to be the root. Define the parent–child arcs*

$$\mathcal{F} = \{(r, u_1), (r, u_2),\ (u_1, \ell_a), (u_1, \ell_b),\ (u_2, \ell_c), (u_2, \ell_d)\}.$$

Then the leaves are $\mathcal{N}_0 = \{\ell_a, \ell_b, \ell_c, \ell_d\}$*, the level-*1 *nodes are* $\mathcal{N}_1 = \{u_1, u_2\}$*, and the unique level-*2 *node is* $\mathcal{N}_2 = \{r\}$*.*

(ii) ***Nested labeling*** η***.*** *We specify* $\eta : \mathcal{N} \to \bigcup_{k=0}^{2} \mathrm{PS}^k(V_0) \setminus \{\emptyset\}$ *by*

$$\eta(\ell_a) = a,\ \eta(\ell_b) = b,\ \eta(\ell_c) = c,\ \eta(\ell_d) = d,$$

$$\eta(u_1) = \{\eta(\ell_a), \eta(\ell_b)\} = \{a, b\}, \qquad \eta(u_2) = \{\eta(\ell_c), \eta(\ell_d)\} = \{c, d\},$$

$$\eta(r) = \{\eta(u_1), \eta(u_2)\} = \{\{a, b\}, \{c, d\}\}.$$

By construction:

- $\eta(\ell_x) \in \mathrm{PS}^0(V_0) = V_0$ *for each* $\ell_x \in \mathcal{N}_0$,
- $\eta(u_i) \in \mathrm{PS}^1(V_0)$ *for* $u_i \in \mathcal{N}_1$,
- $\eta(r) \in \mathrm{PS}^2(V_0)$ *for* $r \in \mathcal{N}_2$,

and the recursive nesting condition $\eta(u) = \{\eta(v) \mid v \in \mathrm{Ch}(u)\}$ *holds for the internal nodes.*

(iii) ***Supports (flattened base-vertex sets).*** *Using* $\lambda(u) = \mathrm{Flat}_{\mathrm{depth}(u)}(\eta(u))$, *we obtain*

$$\lambda(\ell_a) = \{a\},\ \lambda(\ell_b) = \{b\},\ \lambda(\ell_c) = \{c\},\ \lambda(\ell_d) = \{d\},$$

$$\lambda(u_1) = \{a,b\}, \qquad \lambda(u_2) = \{c,d\}, \qquad \lambda(r) = \{a,b,c,d\}.$$

(iv) ***Level edges*** $\{E^{(k)}\}_{k=0}^{2}$. *Define*

$$E^{(0)} = \{\{\ell_a,\ell_c\},\ \{\ell_b,\ell_d\}\}, \qquad E^{(1)} = \{\{u_1,u_2\}\}, \qquad E^{(2)} = \varnothing.$$

Example 2 (A depth-3 TVG on six base vertices with nontrivial middle levels). *Let*

$$V_0 = \{p_1,p_2,p_3,p_4,p_5,p_6\}, \qquad n = 3.$$

We define $\mathrm{TVG}^{(3)} = (V_0, T, \eta, \{E^{(k)}\}_{k=0}^{3})$ *as follows.*

(i) ***Rooted tree.*** *Let the leaves (depth* 0*) be*

$$\mathcal{N}_0 = \{\ell_1,\ell_2,\ell_3,\ell_4,\ell_5,\ell_6\}.$$

*Let the level-*1 *nodes be*

$$\mathcal{N}_1 = \{a_1,a_2,a_3\},$$

*the level-*2 *nodes be*

$$\mathcal{N}_2 = \{b_1,b_2\},$$

and the root (depth 3*) be*

$$\mathcal{N}_3 = \{r\}.$$

Set $\mathcal{N} = \mathcal{N}_0 \cup \mathcal{N}_1 \cup \mathcal{N}_2 \cup \mathcal{N}_3$ *and define arcs*

$$\mathcal{F} = \{(r,b_1),(r,b_2),\ (b_1,a_1),(b_1,a_2),\ (b_2,a_3),\ (a_1,\ell_1),(a_1,\ell_2),\ (a_2,\ell_3),(a_2,\ell_4),\ (a_3,\ell_5),(a_3,\ell_6)\}.$$

(ii) ***Nested labeling*** η. *Define* η *on leaves by the bijection*

$$\eta(\ell_i) = p_i \quad (i = 1,\ldots,6).$$

Then define η *on internal nodes by the recursive rule:*

$$\eta(a_1) = \{\eta(\ell_1),\eta(\ell_2)\} = \{p_1,p_2\},\ \eta(a_2) = \{\eta(\ell_3),\eta(\ell_4)\} = \{p_3,p_4\},\ \eta(a_3) = \{\eta(\ell_5),\eta(\ell_6)\} = \{p_5,p_6\},$$

$$\eta(b_1) = \{\eta(a_1),\eta(a_2)\} = \{\{p_1,p_2\},\{p_3,p_4\}\}, \qquad \eta(b_2) = \{\eta(a_3)\} = \{\{p_5,p_6\}\},$$
$$\eta(r) = \{\eta(b_1),\eta(b_2)\} = \Big\{\{\{p_1,p_2\},\{p_3,p_4\}\},\ \{\{p_5,p_6\}\}\Big\}.$$

(iii) ***Supports (flattened base-vertex sets).*** *Using* $\lambda(u) = \mathrm{Flat}_{\mathrm{depth}(u)}(\eta(u))$, *we obtain*

$$\lambda(\ell_i) = \{p_i\}\ (i = 1,\ldots,6),$$

$$\lambda(a_1) = \{p_1,p_2\}, \quad \lambda(a_2) = \{p_3,p_4\}, \quad \lambda(a_3) = \{p_5,p_6\},$$
$$\lambda(b_1) = \{p_1,p_2,p_3,p_4\}, \quad \lambda(b_2) = \{p_5,p_6\}, \qquad \lambda(r) = V_0.$$

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 29 December 2025 doi:10.20944/preprints202512.2469.v1

(iv) ***Level edges*** $\{E^{(k)}\}_{k=0}^{3}$. *Define*

$$E^{(0)} = \{\{\ell_2, \ell_3\}, \{\ell_4, \ell_6\}\}, \qquad E^{(1)} = \{\{a_1, a_2\}\}, \qquad E^{(2)} = \{\{b_1, b_2\}\}, \qquad E^{(3)} = \varnothing.$$

Notation 1 (Descendants, subtree leaves, and level-k container)**.** *Let $T = (\mathcal{N}, \mathcal{F}, r)$ be a rooted tree whose leaves have depth 0 and root has depth n. Write $v \preceq u$ if v is a descendant of u (possibly $v = u$), i.e. there is a directed path from u to v. For $u \in \mathcal{N}$ define the* subtree leaf set

$$\mathcal{L}(u) \;:=\; \{\ell \in \mathcal{N}_0 \mid \ell \preceq u\}.$$

For each level $k \in \{0, \dots, n\}$ and each base vertex $x \in V_0$, let $\ell_x \in \mathcal{N}_0$ denote the unique leaf with $\eta(\ell_x) = x$ (existence and uniqueness by leaf grounding). Define the level-k container of x *by*

$$\pi_k(x) \;:=\; \text{the unique node } u \in \mathcal{N}_k \text{ such that } \ell_x \in \mathcal{L}(u).$$

(The uniqueness of $\pi_k(x)$ will be proved in Theorem 2).

Theorem 1 (Support equals the set of leaves in the subtree)**.** *Let $\mathrm{TVG}^{(n)} = (V_0, T, \eta, \{E^{(k)}\}_{k=0}^{n})$ be a depth-n TVG and let $\lambda(u) = \mathrm{Flat}_{\mathrm{depth}(u)}(\eta(u)) \subseteq V_0$. Then for every $u \in \mathcal{N}$,*

$$\lambda(u) \;=\; \{\eta(\ell) \mid \ell \in \mathcal{L}(u)\} \;\subseteq\; V_0.$$

In particular, $|\lambda(u)| = |\mathcal{L}(u)|$.

Proof. We argue by induction on $k = \mathrm{depth}(u)$.

Base case $k = 0$. Then $u \in \mathcal{N}_0$ is a leaf, so $\mathcal{L}(u) = \{u\}$. Moreover $\eta(u) \in \mathrm{PS}^0(V_0) = V_0$ and by definition $\lambda(u) = \mathrm{Flat}_0(\eta(u)) = \{\eta(u)\}$. Hence

$$\lambda(u) = \{\eta(u)\} = \{\eta(\ell) \mid \ell \in \mathcal{L}(u)\}.$$

Inductive step. Assume the claim holds for all nodes of depth $< k$, and let $u \in \mathcal{N}_k$ with $k \geq 1$. By recursive nesting,

$$\eta(u) = \{\eta(v) \mid v \in \mathrm{Ch}(u)\}.$$

By the recursive definition of Flat_k (as in Definition 4),

$$\lambda(u) = \mathrm{Flat}_k(\eta(u)) = \bigcup_{v \in \mathrm{Ch}(u)} \mathrm{Flat}_{k-1}(\eta(v)) = \bigcup_{v \in \mathrm{Ch}(u)} \lambda(v).$$

By the induction hypothesis, for each child $v \in \mathrm{Ch}(u)$ we have $\lambda(v) = \{\eta(\ell) \mid \ell \in \mathcal{L}(v)\}$, hence

$$\lambda(u) = \bigcup_{v \in \mathrm{Ch}(u)} \{\eta(\ell) \mid \ell \in \mathcal{L}(v)\}.$$

Finally, the leaves under u are exactly the disjoint union of the leaves under its children:

$$\mathcal{L}(u) = \bigsqcup_{v \in \mathrm{Ch}(u)} \mathcal{L}(v),$$

so the right-hand side equals $\{\eta(\ell) \mid \ell \in \mathcal{L}(u)\}$. This proves the desired identity for depth k. The cardinality statement $|\lambda(u)| = |\mathcal{L}(u)|$ follows because the map $\eta|_{\mathcal{N}_0} : \mathcal{N}_0 \to V_0$ is a bijection, so distinct leaves have distinct base labels. $\square$

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 29 December 2025 doi:10.20944/preprints202512.2469.v1

Theorem 2 (Level supports form a partition of V_0)**.** *Fix $k \in \{0, \ldots, n\}$. Then the family of supports at level k,*

$$\mathcal{L}_k := \{\lambda(u) \mid u \in \mathcal{N}_k\},$$

is a partition of V_0, i.e.

$$V_0 = \bigcup_{u \in \mathcal{N}_k} \lambda(u) \qquad \text{and} \qquad u \neq v \Rightarrow \lambda(u) \cap \lambda(v) = \varnothing \ (u, v \in \mathcal{N}_k).$$

Equivalently, for every $x \in V_0$ there exists a unique $u \in \mathcal{N}_k$ with $x \in \lambda(u)$.

Proof. *Covering.* Let $x \in V_0$, and let $\ell_x \in \mathcal{N}_0$ be the unique leaf with $\eta(\ell_x) = x$. Consider the unique root-to-leaf path from r to ℓ_x. Since depths decrease by 1 along each parent–child arc and the root has depth n while ℓ_x has depth 0, this path contains exactly one node u at depth k. Then $\ell_x \in \mathcal{L}(u)$. By Theorem 1,

$$x = \eta(\ell_x) \in \{\eta(\ell) \mid \ell \in \mathcal{L}(u)\} = \lambda(u),$$

so x lies in the union of the level-k supports.

Disjointness and uniqueness. Let $u, v \in \mathcal{N}_k$ with $u \neq v$. If $\lambda(u) \cap \lambda(v) \neq \varnothing$, choose $x \in \lambda(u) \cap \lambda(v)$. By Theorem 1, there exist leaves $\ell, \ell' \in \mathcal{N}_0$ such that $\ell \in \mathcal{L}(u)$, $\ell' \in \mathcal{L}(v)$, and $\eta(\ell) = x = \eta(\ell')$. Since $\eta|_{\mathcal{N}_0}$ is injective, $\ell = \ell'$. Thus there is a leaf ℓ that lies in both subtrees rooted at u and v. In a tree, the path from the root to ℓ is unique, hence it cannot pass through two distinct nodes at the same depth. Therefore $u = v$, a contradiction. Hence $\lambda(u) \cap \lambda(v) = \varnothing$.

The equivalent uniqueness statement follows: if x were contained in $\lambda(u)$ and $\lambda(v)$ for two level-k nodes, then $\lambda(u) \cap \lambda(v) \neq \varnothing$, forcing $u = v$. □

Corollary 1 (Support recursion and additivity across children)**.** *Let $u \in \mathcal{N}_k$ with $k \geq 1$. Then*

$$\lambda(u) = \bigcup_{v \in \mathrm{Ch}(u)} \lambda(v), \qquad \text{and the union is disjoint, hence} \qquad |\lambda(u)| = \sum_{v \in \mathrm{Ch}(u)} |\lambda(v)|.$$

Proof. The identity $\lambda(u) = \bigcup_{v \in \mathrm{Ch}(u)} \lambda(v)$ was shown in the inductive step of Theorem 1. Children of u all lie at the same depth $k - 1$, so by Theorem 2 (applied at level $k - 1$), their supports are pairwise disjoint. Additivity of cardinalities follows. □

Theorem 3 (Laminarity of supports across all tree-vertices)**.** *For any $u, v \in \mathcal{N}$, exactly one of the following holds:*

$$\lambda(u) \cap \lambda(v) = \varnothing, \qquad \lambda(u) \subseteq \lambda(v), \qquad \lambda(v) \subseteq \lambda(u).$$

Moreover,

$$\lambda(u) \cap \lambda(v) \neq \varnothing \iff (u \preceq v) \text{ or } (v \preceq u).$$

Proof. If $u \preceq v$, then every leaf under u is also a leaf under v, i.e. $\mathcal{L}(u) \subseteq \mathcal{L}(v)$. Applying Theorem 1 gives

$$\lambda(u) = \{\eta(\ell) \mid \ell \in \mathcal{L}(u)\} \subseteq \{\eta(\ell) \mid \ell \in \mathcal{L}(v)\} = \lambda(v).$$

The same reasoning applies if $v \preceq u$.

Now assume neither $u \preceq v$ nor $v \preceq u$. Then the subtrees rooted at u and v are disjoint: if there were a leaf ℓ belonging to both $\mathcal{L}(u)$ and $\mathcal{L}(v)$, the unique root-to-ℓ path would contain both u and v, forcing one to be an ancestor of the other, contradicting the assumption. Hence $\mathcal{L}(u) \cap \mathcal{L}(v) = \varnothing$, and by Theorem 1,

$$\lambda(u) \cap \lambda(v) = \{\eta(\ell) \mid \ell \in \mathcal{L}(u)\} \cap \{\eta(\ell) \mid \ell \in \mathcal{L}(v)\} = \varnothing,$$

because $\eta|_{\mathcal{N}_0}$ is injective and the leaf sets are disjoint. This establishes the trichotomy and the equivalence. □

 doi:10.20944/preprints202512.2469.v1

2.2. n-SuperHyperGraph with Intra-Level and Inter-Level Edges

An n-SuperHyperGraph with intra-level and inter-level edges extends an n-SuperHyperGraph by adding edges among same-level supervertices and cross-level edges linking different nesting depths.

Definition 8 (n-SuperHyperGraph with intra-level and inter-level edges)**.** *Let V_0 be a finite, nonempty set and let $n \in \mathbb{N}$. An n-*SuperHyperGraph with intra-level and inter-level edges *is a tuple*

$$\mathrm{SHG}_{\times}^{(\leq n)} = \Big(V_0,\ V,\ \mathcal{E},\ \{E^{(k)}\}_{k=0}^{n},\ \{E^{(k\to\ell)}\}_{0\leq k<\ell\leq n}\Big),$$

where:

(i) (Graded supervertex set*)*

$$V \subseteq \bigcup_{k=0}^{n} \mathrm{PS}^k(V_0), \qquad V_k := V \cap \mathrm{PS}^k(V_0) \ \ (k = 0, 1, \ldots, n).$$

Optionally (and typically), one assumes grounding $V_0 \subseteq V$ *(equivalently $V_0 = V \cap \mathrm{PS}^0(V_0)$), and $V_k \subseteq \mathrm{PS}^k(V_0) \setminus \{\emptyset\}$ for $k \geq 1$.*

(ii) (Superhyperedges*)*

$$\mathcal{E} \subseteq \mathrm{PS}(V) \setminus \{\emptyset\}.$$

Each $e \in \mathcal{E}$ is a nonempty set of supervertices, possibly mixing levels.

(iii) (Support / flattening*) For $u \in V_k$ define its* support

$$\lambda(u) := \mathrm{Flat}_k(u) \subseteq V_0,$$

where Flat_k is the flattening map from Definition 4. (Thus $\lambda(u)$ is the set of base vertices "contained in" the nested object u.)

(iv) (Intra-level (graph) edges*) For each $k \in \{0, \ldots, n\}$,*

$$E^{(k)} \subseteq \big\{\{u, v\} \subseteq V_k \mid u \neq v\big\}.$$

An edge $\{u, v\} \in E^{(k)}$ encodes a binary relation between two supervertices at the same *level k.*

(v) (Inter-level (directed) edges*) For each pair $0 \leq k < \ell \leq n$,*

$$E^{(k\to\ell)} \subseteq V_k \times V_\ell.$$

An ordered pair $(u, v) \in E^{(k\to\ell)}$ is an inter-level edge *from level k to level ℓ.*

(vi) (Optional support-compatibility constraints*) Depending on semantics, one may require every $(u, v) \in E^{(k\to\ell)}$ to satisfy, for example,*

$$\lambda(u) \subseteq \lambda(v) \quad \textit{(upward/abstraction)}, \qquad \textit{or} \qquad \lambda(u) \cap \lambda(v) \neq \emptyset \quad \textit{(overlap)}.$$

Remark 3 (Relation to the classical n-SuperHyperGraph)**.** *If one restricts V to a single level $V = V_n \subseteq \mathrm{PS}^n(V_0)$ and discards all $\{E^{(k)}\}_{k<n}$ and $\{E^{(k\to\ell)}\}_{k<\ell}$, then $(V, \mathcal{E})$ is precisely an n-SuperHyperGraph in the sense of Definition 5. The present definition enriches the model by allowing* graded *supervertices (levels 0 through n) and by equipping them with both intra-level and inter-level* graph-type *edges in addition to hyperedges.*

As a reference, a comparison between an n-SuperHyperGraph and an n-SuperHyperGraph with intra-level and inter-level edges is presented in Table 3.

 doi:10.20944/preprints202512.2469.v1

Table 3. Comparison between an n-SuperHyperGraph and an n-SuperHyperGraph with intra-level and inter-level edges.

Aspect	Classical n-SuperHyperGraph $\mathrm{SHG}^{(n)} = (V, E)$	n-SHG with intra-/inter-level edges $\mathrm{SHG}^{(\le n)}_{\times}$ (Definition 8)
Base set	Finite nonempty V_0	Finite nonempty V_0
Vertex objects	Single-level supervertices: $V \subseteq \mathrm{PS}^n(V_0)$	Graded supervertices: $V \subseteq \bigcup_{k=0}^{n} \mathrm{PS}^k(V_0)$ with level slices $V_k = V \cap \mathrm{PS}^k(V_0)$
Grounding at level 0	Not required by definition (often implicit via choosing $V \subseteq \mathrm{PS}^n(V_0)$)	Optional but typical: $V_0 \subseteq V$ (i.e., base vertices appear as level-0 supervertices)
Primary relations	Hyperedges only: $E \subseteq \mathrm{PS}(V) \setminus \{\emptyset\}$	Hyperedges $\mathcal{E} \subseteq \mathrm{PS}(V) \setminus \{\emptyset\}$ *plus* binary edges within and across levels
Hyperedges mix levels?	Not applicable (only level n vertices exist)	Yes: $e \in \mathcal{E}$ may contain supervertices from different levels (unless restricted)
Intra-level edges	None	For each k, undirected graph edges $E^{(k)} \subseteq \{\{u,v\} \subseteq V_k \mid u \neq v\}$
Inter-level edges	None	For $0 \le k < \ell \le n$, directed cross edges $E^{(k\to\ell)} \subseteq V_k \times V_\ell$
Support / flattening to V_0	Not part of the minimal definition (can be added externally)	Built-in via $\lambda(u) = \mathrm{Flat}_k(u)$ for $u \in V_k$, enabling semantic constraints on cross edges
Expressivity gain	Encodes higher-order (set-valued) vertices and hyperedges at one fixed nesting depth n	Adds explicit *multi-resolution* (levels 0 to n) and explicit *binary* relations both within a level and between different depths
Recovery of classical model	By definition	If one restricts to $V = V_n$ and discards $\{E^{(k)}\}_{k=0}^{n}$ and $\{E^{(k\to\ell)}\}_{k<\ell}$, then $(V, \mathcal{E})$ reduces to a classical n-SuperHyperGraph (Remark 3).

Example 3 (A depth-2 instance with both hyperedges and cross-level edges). *Let $V_0 = \{a, b, c, d\}$ and $n = 2$. Define the graded supervertex sets*

$$V_0 = \{a, b, c, d\}, \qquad V_1 = \{A, B, C\}, \qquad V_2 = \{U, W\},$$

where

$$A = \{a, b\}, \qquad B = \{c, d\}, \qquad C = \{b, c\} \in \mathrm{PS}^1(V_0) = \mathrm{PS}(V_0),$$

and

$$U = \{A, B\}, \qquad W = \{A, C\} \in \mathrm{PS}^2(V_0) = \mathrm{PS}(\mathrm{PS}(V_0)).$$

Set $V := V_0 \cup V_1 \cup V_2 \subseteq \mathrm{PS}^0(V_0) \cup \mathrm{PS}^1(V_0) \cup \mathrm{PS}^2(V_0)$.

Supports (via flattening). *Using $\lambda(u) = \mathrm{Flat}_{\mathrm{level}(u)}(u)$,*

$$\lambda(a) = \{a\},\ \lambda(b) = \{b\},\ \lambda(c) = \{c\},\ \lambda(d) = \{d\},$$

$$\lambda(A) = \{a, b\}, \quad \lambda(B) = \{c, d\}, \quad \lambda(C) = \{b, c\},$$

$$\lambda(U) = \{a, b, c, d\}, \quad \lambda(W) = \{a, b, c\}.$$

Superhyperedges. *Let the hyperedge family be*

$$\mathcal{E} \;=\; \{\{a, A\},\ \{b, A, C\},\ \{c, C, B\},\ \{A, C, W\},\ \{U, W, B\}\} \;\subseteq\; \mathrm{PS}(V) \setminus \{\emptyset\}.$$

These hyperedges allow genuinely multiway relations, and they may mix levels (e.g., $\{U, W, B\}$ lives across levels 2 and 1).

Intra-level edges. *Define*

$$E^{(0)} = \{\{a, b\}, \{c, d\}\}, \qquad E^{(1)} = \{\{A, C\}\}, \qquad E^{(2)} = \{\{U, W\}\}.$$

For instance, $\{A, C\} \in E^{(1)}$ encodes a binary relation between two level-1 group-vertices (perhaps "overlapping communities" or "related clusters").

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 29 December 2025 doi:10.20944/preprints202512.2469.v1

Inter-level edges. *Let*

$$E^{(0\to1)} = \{(a,A),(b,A),(b,C),(c,C),(c,B),(d,B)\},$$

$$E^{(1\to2)} = \{(A,U),(B,U),(A,W),(C,W)\}, \qquad E^{(0\to2)} = \{(a,U),(d,U)\}.$$

If one enforces the optional upward constraint $\lambda(u) \subseteq \lambda(v)$, then it holds here: for example, $(b,C) \in E^{(0\to1)}$ satisfies $\lambda(b) = \{b\} \subseteq \lambda(C) = \{b,c\}$, and $(C,W) \in E^{(1\to2)}$ satisfies $\lambda(C) = \{b,c\} \subseteq \lambda(W) = \{a,b,c\}$. Thus the inter-level edges can be read as membership/abstraction links *across levels.*

Consequently,

$$\mathrm{SHG}_{\times}^{(\le 2)} = \Big(V_0,\ V,\ \mathcal{E},\ \{E^{(0)},E^{(1)},E^{(2)}\},\ \{E^{(0\to1)},E^{(0\to2)},E^{(1\to2)}\}\Big)$$

is a concrete 2*-SuperHyperGraph with both intra-level and inter-level edges in the sense of Definition 8.*

2.3. Depth-n TVG with Intra-Level and Inter-Level Edges

A depth-n TVG with intra-level and inter-level edges is a depth-n rooted tree whose nodes carry iterated-powerset labels; edges relate nodes within the same level and across levels.

Definition 9 (Depth-n TVG with intra-level and inter-level edges)**.** *Let V_0 be a finite, nonempty set and let $n \in \mathbb{N}$. A* depth-n Tree-Vertex Graph with cross-level edges *is a tuple*

$$\mathrm{TVG}_{\times}^{(n)} \;=\; \Big(V_0,\ T,\ \eta,\ \{E^{(k)}\}_{k=0}^{n},\ \{E^{(k\to\ell)}\}_{0\le k<\ell\le n}\Big),$$

where:

(i) $T = (\mathcal{N},\mathcal{F},r)$ *is a rooted tree whose leaves have depth* 0 *and whose root has depth n. Let $\mathcal{N}_k = \{u \in \mathcal{N} \mid \mathrm{depth}(u) = k\}$.*

(ii) $\eta : \mathcal{N} \to \bigcup_{k=0}^{n} \mathrm{PS}^k(V_0) \setminus \{\emptyset\}$ *is a nested labeling such that:*

- *(a)* (Level-typing) *If $u \in \mathcal{N}_k$, then $\eta(u) \in \mathrm{PS}^k(V_0) \setminus \{\emptyset\}$.*
- *(b)* (Leaf grounding) $\eta|_{\mathcal{N}_0} : \mathcal{N}_0 \to V_0$ *is a bijection.*
- *(c)* (Recursive nesting) *For every $u \in \mathcal{N}_k$ with $k \ge 1$,*

$$\eta(u) = \{\eta(v) \mid v \in \mathrm{Ch}(u)\}.$$

(iii) Define the flattened support $\lambda(u) \subseteq V_0$ by $\lambda(u) = \mathrm{Flat}_{\mathrm{depth}(u)}(\eta(u))$.

(iv) (Intra-level edges) *For each $k \in \{0,\dots,n\}$,*

$$E^{(k)} \subseteq \big\{\{u,v\} \subseteq \mathcal{N}_k \mid u \ne v\big\}.$$

(v) (Inter-level edges / cross edges) *For each pair $0 \le k < \ell \le n$,*

$$E^{(k\to\ell)} \subseteq \mathcal{N}_k \times \mathcal{N}_\ell.$$

An ordered pair $(u,v) \in E^{(k\to\ell)}$ is an inter-level edge *from level k to level ℓ.*

(vi) (Support-compatibility constraint, optional) *One may additionally require that every inter-level edge $(u,v) \in E^{(k\to\ell)}$ satisfies a support constraint such as either:*

$$\lambda(u) \subseteq \lambda(v) \quad \textit{(upward/abstraction edge)}, \qquad \textit{or} \qquad \lambda(v) \subseteq \lambda(u) \quad \textit{(downward/refinement edge)},$$

or the weaker overlap condition $\lambda(u) \cap \lambda(v) \ne \emptyset$, depending on the intended semantics.

 doi:10.20944/preprints202512.2469.v1

Remark 4 (Undirected cross-edges). *If one prefers undirected inter-level edges, replace $E^{(k\to\ell)} \subseteq \mathcal{N}_k \times \mathcal{N}_\ell$ by a symmetric set*

$$E^{(k,\ell)} \subseteq \{\{u,v\} \mid u \in \mathcal{N}_k,\ v \in \mathcal{N}_\ell\},$$

and interpret $\{u,v\}$ as an undirected relation between levels k and ℓ.

For reference, the comparison between a depth-n TVG and a depth-n TVG with intra-level and inter-level edges is presented in Table 4, and the comparison between a depth-n TVG with intra-level and inter-level edges and an n-SuperHyperGraph with intra-level and inter-level edges is presented in Table 5.

Table 4. Comparison between a depth-n TVG and a depth-n TVG with intra-level and inter-level edges.

Aspect	Depth-n TVG (level edges only)	Depth-n TVG with intra- and inter-level edges
Core structure	Rooted tree $T = (\mathcal{N}, \mathcal{F}, r)$ of fixed depth n with level sets $\mathcal{N}_k$	Same rooted tree backbone T with the same level decomposition $\mathcal{N} = \bigsqcup_{k=0}^{n} \mathcal{N}_k$
Objects (vertices)	Tree-vertices $\mathcal{N}$ (hierarchical units)	Same tree-vertices $\mathcal{N}$
Typing / hierarchy encoding	Nested labeling η is level-typed: $u \in \mathcal{N}_k \Rightarrow \eta(u) \in \mathrm{PS}^k(V_0)$, and $\eta(u) = \{\eta(v) \mid v \in \mathrm{Ch}(u)\}$ for $k \geq 1$	Same nested, level-typed labeling η (hierarchy still enforced by the tree and recursion)
Grounding at level 0	$\eta\|_{\mathcal{N}_0} : \mathcal{N}_0 \to V_0$ is a bijection	Same leaf grounding
Flattened support	$\lambda(u) = \mathrm{Flat}_{\mathrm{depth}(u)}(\eta(u)) \subseteq V_0$	Same support definition $\lambda(u)$
Intra-level edges	For each k, an undirected edge set $E^{(k)} \subseteq \{\{u,v\} \subseteq \mathcal{N}_k \mid u \neq v\}$	Same intra-level edge sets $E^{(k)}$
Inter-level (cross) edges	Not present	Present as directed relations across levels: $E^{(k\to\ell)} \subseteq \mathcal{N}_k \times \mathcal{N}_\ell$ for $0 \leq k < \ell \leq n$
Typical semantics of extra edges	Relations only among units at the same abstraction depth	Relations can connect units of different abstraction depths (e.g., "refines/abstracts/depends-on/explains")
Optional constraints	(Usually none beyond the tree/label recursion)	May impose support-compatibility on $(u,v) \in E^{(k\to\ell)}$ (e.g., $\lambda(u) \subseteq \lambda(v)$, or $\lambda(u) \cap \lambda(v) \neq \emptyset$)

Table 5. Comparison between a depth-n TVG with intra-level and inter-level edges and an n-SuperHyperGraph with intra-level and inter-level edges.

Aspect	Depth-n TVG with intra- and inter-level edges	n-SHG with intra- and inter-level edges
Underlying "hierarchy carrier"	A rooted tree T fixes the hierarchy: every non-leaf has children and the nesting is along parent–child arcs	No tree is required; hierarchy is encoded by membership in iterated powersets (i.e., by level k in $\mathrm{PS}^k(V_0)$) and by chosen supervertices
Vertex objects	Tree-vertices $\mathcal{N}$ (each represents a hierarchical unit)	Supervertices $V \subseteq \bigcup_{k=0}^{n} \mathrm{PS}^k(V_0)$ (chosen nested set-objects)
Level decomposition	Levels are $\mathcal{N}_k = \{u \in \mathcal{N} \mid \mathrm{depth}(u) = k\}$ with *fixed depth n*	Levels are $V_k = V \cap \mathrm{PS}^k(V_0)$ (graded by iterated powerset depth)
How higher-level objects are formed	Rigid recursion: $\eta(u) = \{\eta(v) \mid v \in \mathrm{Ch}(u)\}$ (every internal node is exactly the set of its children labels)	Flexible selection: a supervertex in V_k is any element of $\mathrm{PS}^k(V_0)$ included in V (no requirement that it equals the set of "children" of something)
Grounding at level 0	$\eta\|_{\mathcal{N}_0} : \mathcal{N}_0 \to V_0$ is a bijection (all base vertices appear as leaves)	Typically assumes grounding $V_0 \subseteq V$; otherwise optional in general formulations
Support map	$\lambda(u) = \mathrm{Flat}_{\mathrm{depth}(u)}(\eta(u)) \subseteq V_0$	$\lambda(u) = \mathrm{Flat}_k(u)$ for $u \in V_k$ (same flattening idea, but applied directly to nested set-objects)
Intra-level (graph) edges	$E^{(k)} \subseteq \{\{u,v\} \subseteq \mathcal{N}_k \mid u \neq v\}$	$E^{(k)} \subseteq \{\{u,v\} \subseteq V_k \mid u \neq v\}$
Inter-level (directed) edges	$E^{(k\to\ell)} \subseteq \mathcal{N}_k \times \mathcal{N}_\ell$	$E^{(k\to\ell)} \subseteq V_k \times V_\ell$
Higher-arity relations	Not part of the core (unless separately added); relations are primarily graph-type edges	Includes hyperedges $\mathcal{E} \subseteq \mathrm{PS}(V) \setminus \{\emptyset\}$ in addition to graph-type edges
Modeling emphasis	Tree-shaped, laminar, and recursively generated clusters; edges attach to explicitly organized hierarchical units	General nested set-valued vertices with both hyperedges (multiway) and graph-type edges (binary), without a required tree backbone

Specific examples are given below.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 29 December 2025 doi:10.20944/preprints202512.2469.v1

Example 4 (A depth-2 TVG with both intra-level and inter-level edges (overlap-compatible cross edges))**.** *Let*

$$V_0 = \{a, b, c, d\}, \qquad n = 2.$$

We define

$$\mathrm{TVG}_{\times}^{(2)} = \Big(V_0,\ T,\ \eta,\ \{E^{(k)}\}_{k=0}^{2},\ \{E^{(k\to\ell)}\}_{0\le k<\ell\le 2}\Big)$$

by specifying each component.

(i) ***Rooted tree*** T**.** *Let the node set be*

$$\mathcal{N} = \{\ell_a, \ell_b, \ell_c, \ell_d,\ u_1, u_2,\ r\}.$$

Take r as the root and define the parent–child arcs

$$\mathcal{F} = \{(r, u_1), (r, u_2),\ (u_1, \ell_a), (u_1, \ell_b),\ (u_2, \ell_c), (u_2, \ell_d)\}.$$

Thus

$$\mathcal{N}_0 = \{\ell_a, \ell_b, \ell_c, \ell_d\}, \quad \mathcal{N}_1 = \{u_1, u_2\}, \quad \mathcal{N}_2 = \{r\}.$$

(ii) ***Nested labeling*** η**.** *Define η on the leaves by*

$$\eta(\ell_a) = a,\ \eta(\ell_b) = b,\ \eta(\ell_c) = c,\ \eta(\ell_d) = d,$$

and on internal nodes by the recursive nesting rule:

$$\eta(u_1) = \{\eta(\ell_a), \eta(\ell_b)\} = \{a, b\}, \qquad \eta(u_2) = \{\eta(\ell_c), \eta(\ell_d)\} = \{c, d\},$$

$$\eta(r) = \{\eta(u_1), \eta(u_2)\} = \{\{a, b\}, \{c, d\}\}.$$

(iii) ***Flattened supports*** λ**.** *Using $\lambda(u) = \mathrm{Flat}_{\mathrm{depth}(u)}(\eta(u))$, we obtain*

$$\lambda(\ell_a) = \{a\},\ \lambda(\ell_b) = \{b\},\ \lambda(\ell_c) = \{c\},\ \lambda(\ell_d) = \{d\},$$

$$\lambda(u_1) = \{a, b\}, \qquad \lambda(u_2) = \{c, d\}, \qquad \lambda(r) = \{a, b, c, d\}.$$

(iv) ***Intra-level edges.*** *Set*

$$E^{(0)} = \{\{\ell_a, \ell_c\},\ \{\ell_b, \ell_d\}\}, \qquad E^{(1)} = \{\{u_1, u_2\}\}, \qquad E^{(2)} = \varnothing.$$

(v) ***Inter-level edges.*** *We give cross edges at both pairs of levels:*

$$E^{(0\to1)} = \{(\ell_a, u_1), (\ell_c, u_2)\}, \qquad E^{(0\to2)} = \{(\ell_b, r)\}, \qquad E^{(1\to2)} = \{(u_1, r), (u_2, r)\}.$$

(vi) ***Compatibility check (overlap condition).*** *Every listed inter-level edge (u, v) satisfies $\lambda(u) \cap \lambda(v) \neq \varnothing$: for instance, $(\ell_b, r) \in E^{(0\to2)}$ has $\lambda(\ell_b) = \{b\}$ and $\lambda(r) = \{a, b, c, d\}$, so the intersection is $\{b\}$; similarly (ℓ_a, u_1) and (ℓ_c, u_2) overlap by construction.*

Example 5 (A depth-3 TVG with both intra-level and inter-level edges (mix of upward and downward semantics))**.** *Let*

$$V_0 = \{p_1, p_2, p_3, p_4, p_5, p_6\}, \qquad n = 3.$$

We define

$$\mathrm{TVG}_{\times}^{(3)} = \Big(V_0,\ T,\ \eta,\ \{E^{(k)}\}_{k=0}^{3},\ \{E^{(k\to\ell)}\}_{0\le k<\ell\le 3}\Big).$$

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 29 December 2025 doi:10.20944/preprints202512.2469.v1

(i) ***Rooted tree*** T*. Let the levels be*

$$\mathcal{N}_0 = \{\ell_1, \ell_2, \ell_3, \ell_4, \ell_5, \ell_6\}, \quad \mathcal{N}_1 = \{a_1, a_2, a_3\}, \quad \mathcal{N}_2 = \{b_1, b_2\}, \quad \mathcal{N}_3 = \{r\}.$$

Let the arc set be

$$\mathcal{F} = \{(r, b_1), (r, b_2), \ (b_1, a_1), (b_1, a_2), \ (b_2, a_3), \ (a_1, \ell_1), (a_1, \ell_2), \ (a_2, \ell_3), (a_2, \ell_4), \ (a_3, \ell_5), (a_3, \ell_6)\}.$$

(ii) ***Nested labeling*** η. *Define* $\eta(\ell_i) = p_i$ *for* $i = 1, \ldots, 6$*. Then recursively:*

$$\eta(a_1) = \{\eta(\ell_1), \eta(\ell_2)\} = \{p_1, p_2\}, \quad \eta(a_2) = \{\eta(\ell_3), \eta(\ell_4)\} = \{p_3, p_4\}, \quad \eta(a_3) = \{\eta(\ell_5), \eta(\ell_6)\} = \{p_5, p_6\},$$

$$\eta(b_1) = \{\eta(a_1), \eta(a_2)\} = \{\{p_1, p_2\}, \{p_3, p_4\}\},$$
$$\eta(b_2) = \{\eta(a_3)\} = \{\{p_5, p_6\}\},$$
$$\eta(r) = \{\eta(b_1), \eta(b_2)\} = \Big\{\{\{p_1, p_2\}, \{p_3, p_4\}\}, \ \{\{p_5, p_6\}\}\Big\}.$$

(iii) ***Flattened supports*** λ.

$$\lambda(\ell_i) = \{p_i\} \ (i = 1, \ldots, 6), \qquad \lambda(a_1) = \{p_1, p_2\}, \ \lambda(a_2) = \{p_3, p_4\}, \ \lambda(a_3) = \{p_5, p_6\},$$

$$\lambda(b_1) = \{p_1, p_2, p_3, p_4\}, \quad \lambda(b_2) = \{p_5, p_6\}, \quad \lambda(r) = V_0.$$

(iv) ***Intra-level edges.*** *We specify relations within each abstraction depth:*

$$E^{(0)} = \{\{\ell_2, \ell_3\}, \ \{\ell_4, \ell_6\}\}, \qquad E^{(1)} = \{\{a_1, a_2\}\}, \qquad E^{(2)} = \{\{b_1, b_2\}\}, \qquad E^{(3)} = \varnothing.$$

(v) ***Inter-level edges (cross edges).*** *We include a mix of* upward *cross edges (support inclusion* $\lambda(u) \subseteq \lambda(v)$*) and* downward *cross edges (support inclusion* $\lambda(v) \subseteq \lambda(u)$*) by choosing:*

$$E^{(0\to1)} = \{(\ell_1, a_1), (\ell_4, a_2), (\ell_6, a_3)\}, \qquad E^{(1\to2)} = \{(a_2, b_1), (a_3, b_2)\}, \qquad E^{(2\to3)} = \{(b_1, r), (b_2, r)\},$$

together with two downward *edges from the top:*

$$E^{(3\to1)} = \{(r, a_1)\}, \qquad E^{(3\to0)} = \{(r, \ell_5)\}.$$

(Equivalently, one may encode these as $E^{(1\to3)}$ *and* $E^{(0\to3)}$ *with reversed orientation, depending on whether the intended semantics is "refines" or "abstracts.")*

(vi) ***Compatibility check (inclusion/overlap).*** *For each upward edge* $(u, v) \in E^{(0\to1)} \cup E^{(1\to2)} \cup E^{(2\to3)}$*, we have* $\lambda(u) \subseteq \lambda(v)$*, e.g.* $\lambda(\ell_4) = \{p_4\} \subseteq \lambda(a_2) = \{p_3, p_4\} \subseteq \lambda(b_1) = \{p_1, p_2, p_3, p_4\} \subseteq \lambda(r)$*. For each downward edge* (r, a_1) *and* (r, ℓ_5) *we have* $\lambda(a_1) \subseteq \lambda(r)$ *and* $\lambda(\ell_5) \subseteq \lambda(r)$*, so in particular* $\lambda(r) \cap \lambda(a_1) \neq \varnothing$ *and* $\lambda(r) \cap \lambda(\ell_5) \neq \varnothing$*.*

Notation 2 (Descendants, subtree leaves, and level-k container). *Let* $T = (\mathcal{N}, \mathcal{F}, r)$ *be a rooted tree whose leaves have depth* 0 *and whose root has depth* n*. Write* $v \preceq u$ *if* v *is a descendant of* u *(possibly* $v = u$*). For* $u \in \mathcal{N}$ *define the subtree leaf set*

$$\mathcal{L}(u) := \{\ell \in \mathcal{N}_0 \mid \ell \preceq u\}.$$

For each $x \in V_0$*, let* $\ell_x \in \mathcal{N}_0$ *denote the unique leaf with* $\eta(\ell_x) = x$*. For each* $k \in \{0, \ldots, n\}$ *define the level-*k *container map*

$$\pi_k : V_0 \to \mathcal{N}_k, \qquad \pi_k(x) := \text{the unique node } u \in \mathcal{N}_k \text{ such that } \ell_x \in \mathcal{L}(u).$$

(The existence and uniqueness of π_k *is established in Theorem 5.)*

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 29 December 2025 doi:10.20944/preprints202512.2469.v1

Theorem 4 (Support equals the base labels of subtree leaves)**.** *Let* $\mathrm{TVG}_{\times}^{(n)}$ *be as in Definition 9. Then for every* $u \in \mathcal{N}$*,*

$$\lambda(u) = \{\eta(\ell) \mid \ell \in \mathcal{L}(u)\} \subseteq V_0.$$

In particular, $|\lambda(u)| = |\mathcal{L}(u)|$*.*

Proof. Identical to the proof of the corresponding statement for level-only TVGs: induct on $k = \mathrm{depth}(u)$ using recursive nesting $\eta(u) = \{\eta(v) \mid v \in \mathrm{Ch}(u)\}$ and the recursive definition of Flat_k. The presence of additional edge sets $\{E^{(k \to \ell)}\}$ does not affect the labeling argument. □

Theorem 5 (Level supports partition V_0)**.** *Fix* $k \in \{0, \ldots, n\}$*. Then the family*

$$\mathcal{L}_k := \{\lambda(u) \mid u \in \mathcal{N}_k\}$$

is a partition of V_0*:*

$$V_0 = \bigcup_{u \in \mathcal{N}_k} \lambda(u), \qquad u \neq v \Rightarrow \lambda(u) \cap \lambda(v) = \varnothing \quad (u, v \in \mathcal{N}_k).$$

Equivalently, for every $x \in V_0$ *there exists a unique* $u \in \mathcal{N}_k$ *with* $x \in \lambda(u)$*, and thus* π_k *in Notation 2 is well-defined.*

Proof. Let $x \in V_0$ and let ℓ_x be its unique leaf. The unique root-to-ℓ_x path contains exactly one node u at depth k. Then $\ell_x \in \mathcal{L}(u)$, so by Theorem 4 we have $x = \eta(\ell_x) \in \lambda(u)$, proving the covering.

For disjointness, let $u, v \in \mathcal{N}_k$ with $u \neq v$ and suppose $\lambda(u) \cap \lambda(v) \neq \varnothing$. Choose x in the intersection. By Theorem 4 there exist leaves $\ell \in \mathcal{L}(u)$ and $\ell' \in \mathcal{L}(v)$ with $\eta(\ell) = x = \eta(\ell')$. Injectivity of $\eta|_{\mathcal{N}_0}$ implies $\ell = \ell'$. But a root-to-leaf path cannot pass through two distinct nodes at the same depth, contradiction. Hence the supports are disjoint. □

Theorem 6 (Laminarity of supports across all tree-vertices)**.** *For any* $u, v \in \mathcal{N}$*, exactly one of the following holds:*

$$\lambda(u) \cap \lambda(v) = \varnothing, \qquad \lambda(u) \subseteq \lambda(v), \qquad \lambda(v) \subseteq \lambda(u).$$

Moreover,

$$\lambda(u) \cap \lambda(v) \neq \varnothing \quad \Longleftrightarrow \quad (u \preceq v) \text{ or } (v \preceq u).$$

Proof. If $u \preceq v$, then $\mathcal{L}(u) \subseteq \mathcal{L}(v)$, hence

$$\lambda(u) = \{\eta(\ell) \mid \ell \in \mathcal{L}(u)\} \subseteq \{\eta(\ell) \mid \ell \in \mathcal{L}(v)\} = \lambda(v)$$

by Theorem 4. Symmetrically for $v \preceq u$.

If neither is an ancestor of the other, the subtrees are disjoint: $\mathcal{L}(u) \cap \mathcal{L}(v) = \varnothing$. Then, using injectivity of $\eta|_{\mathcal{N}_0}$,

$$\lambda(u) \cap \lambda(v) = \{\eta(\ell) \mid \ell \in \mathcal{L}(u)\} \cap \{\eta(\ell) \mid \ell \in \mathcal{L}(v)\} = \varnothing.$$

This yields the trichotomy and the equivalence. □

Definition 10 (Cross-edge expansion to a base relation)**.** *Fix* $0 \leq k < \ell \leq n$*. For* $(u, v) \in \mathcal{N}_k \times \mathcal{N}_\ell$ *define its base expansion as the set*

$$\mathrm{Exp}(u, v) := \lambda(u) \times \lambda(v) \ \subseteq \ V_0 \times V_0.$$

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 29 December 2025 doi:10.20944/preprints202512.2469.v1

Given a cross-edge set $E^{(k\to\ell)} \subseteq \mathcal{N}_k \times \mathcal{N}_\ell$, define the induced base relation

$$R_0^{(k\to\ell)} := \bigcup_{(u,v)\in E^{(k\to\ell)}} \operatorname{Exp}(u,v) \subseteq V_0 \times V_0.$$

Theorem 7 (Uniqueness of lifted endpoints for induced base pairs)**.** *Fix $0 \le k < \ell \le n$ and define $R_0^{(k\to\ell)}$ as in Definition 10. If $(x,y) \in R_0^{(k\to\ell)}$, then the nodes*

$$u = \pi_k(x) \in \mathcal{N}_k, \qquad v = \pi_\ell(y) \in \mathcal{N}_\ell$$

are uniquely determined and satisfy $(u,v) \in E^{(k\to\ell)}$. Conversely, if $(u,v) \in E^{(k\to\ell)}$ and $x \in \lambda(u)$, $y \in \lambda(v)$, then $(x,y) \in R_0^{(k\to\ell)}$.

Proof. Assume $(x,y) \in R_0^{(k\to\ell)}$. Then by definition there exists $(u',v') \in E^{(k\to\ell)}$ such that $x \in \lambda(u')$ and $y \in \lambda(v')$. By Theorem 5 applied at level k and at level ℓ, the level containers are unique:

$$u' = \pi_k(x), \qquad v' = \pi_\ell(y).$$

Hence $(\pi_k(x), \pi_\ell(y)) \in E^{(k\to\ell)}$, and uniqueness is immediate from the partition property.

Conversely, if $(u,v) \in E^{(k\to\ell)}$ and $x \in \lambda(u)$, $y \in \lambda(v)$, then $(x,y) \in \lambda(u) \times \lambda(v) = \operatorname{Exp}(u,v) \subseteq R_0^{(k\to\ell)}$. □

Theorem 8 (Support-compatibility implies monotone containment on base pairs)**.** *Fix $0 \le k < \ell \le n$ and suppose the following* upward support-compatibility *holds:*

$$\forall (u,v) \in E^{(k\to\ell)} : \ \lambda(u) \subseteq \lambda(v).$$

Then the induced base relation $R_0^{(k\to\ell)}$ satisfies:

$$(x,y) \in R_0^{(k\to\ell)} \implies x \in \lambda(\pi_\ell(y)).$$

Equivalently, every base pair (x,y) induced by a cross-edge points from a base vertex x that lies inside the (unique) level-ℓ cluster containing y.

Proof. Take $(x,y) \in R_0^{(k\to\ell)}$. By Theorem 7, $(\pi_k(x), \pi_\ell(y)) \in E^{(k\to\ell)}$. By the assumed compatibility,

$$\lambda(\pi_k(x)) \subseteq \lambda(\pi_\ell(y)).$$

Since $x \in \lambda(\pi_k(x))$ by definition of π_k, we conclude $x \in \lambda(\pi_\ell(y))$. □

Theorem 9 (No cross-edges between disjoint supports under overlap-compatibility)**.** *Fix $0 \le k < \ell \le n$ and assume the* overlap-compatibility *constraint:*

$$\forall (u,v) \in E^{(k\to\ell)} : \ \lambda(u) \cap \lambda(v) \neq \emptyset.$$

Then every cross-edge connects comparable nodes in the tree:

$$(u,v) \in E^{(k\to\ell)} \implies u \preceq v.$$

In particular, under overlap-compatibility, cross-edges cannot jump between disjoint branches.

 doi:10.20944/preprints202512.2469.v1

Proof. Let $(u, v) \in E^{(k \to \ell)}$. By assumption $\lambda(u) \cap \lambda(v) \neq \emptyset$. By Theorem 6, this implies $u \preceq v$ or $v \preceq u$. But $\mathrm{depth}(u) = k < \ell = \mathrm{depth}(v)$, so $v \preceq u$ is impossible (a descendant must have smaller depth). Hence $u \preceq v$. □

Corollary 2 (Under overlap-compatibility, cross-edges are determined by ancestor constraints)**.** *Assume overlap-compatibility as in Theorem 9. Then for each* $0 \leq k < \ell \leq n$,

$$E^{(k \to \ell)} \subseteq \{(u, v) \in \mathcal{N}_k \times \mathcal{N}_\ell \mid u \preceq v\}.$$

Moreover, for any $(u, v) \in E^{(k \to \ell)}$ *one has* $\lambda(u) \subseteq \lambda(v)$.

Proof. The containment is exactly Theorem 9. If $u \preceq v$, then $\mathcal{L}(u) \subseteq \mathcal{L}(v)$ and thus $\lambda(u) \subseteq \lambda(v)$ by Theorem 4. □

3. Conclusions

In this paper, we defined a new class of graphs called Tree-Vertex Graphs. We expect that future work will explore extensions based on Fuzzy Sets [26], Neutrosophic Sets [27,28], and Plithogenic Sets [29,30], as well as applications to methods such as neural networks.

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

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

Informed Consent Statement: Not applicable.

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

Public Involvement Statement: This study did not involve any clinical trials.

Use of Artificial Intelligence: No code or software was developed for this study. I use generative AI and AI-assisted tools for tasks such as English grammar checking, and I do not employ them in any way that violates ethical standards.

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

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

References

1. Diestel, R. *Graph theory*; Springer (print edition); Reinhard Diestel (eBooks), 2024.
2. Gao, Y.; Zhang, Z.; Lin, H.; Zhao, X.; Du, S.; Zou, C. Hypergraph learning: Methods and practices. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2020**, *44*, 2548–2566.
3. Feng, Y.; You, H.; Zhang, Z.; Ji, R.; Gao, Y. Hypergraph neural networks. In Proceedings of the Proceedings of the AAAI conference on artificial intelligence, 2019, pp. 3558–3565.
4. Chen, J.; Schwaller, P. Molecular hypergraph neural networks. *The Journal of Chemical Physics* **2024**, *160*.
5. Ji, S.; Feng, Y.; Di, D.; Ying, S.; Gao, Y. Mode Hypergraph Neural Network. *IEEE Transactions on Neural Networks and Learning Systems* **2025**.
6. Casetti, N.; Nevatia, P.; Chen, J.; Schwaller, P.; Coley, C.W. Comment on "Molecular hypergraph neural networks"[J. Chem. Phys. 160, 144307 (2024)]. *The Journal of Chemical Physics* **2024**, *161*.
7. Du, W.; Zhang, S.; Cai, Z.; Li, X.; Liu, Z.; Fang, J.; Wang, J.; Wang, X.; Wang, Y. Molecular Merged Hypergraph Neural Network for Explainable Solvation Gibbs Free Energy Prediction. *Research* **2025**, *8*, 0740.

doi:10.20944/preprints202512.2469.v1

8. Bravo, J.C.M.; Piedrahita, C.J.B.; Bravo, M.A.M.; Pilacuan-Bonete, L.M. Integrating SMED and Industry 4.0 to optimize processes with plithogenic n-SuperHyperGraphs. *Neutrosophic Sets and Systems* **2025**, *84*, 328–340.
9. Fujita, T. Multi-SuperHyperGraph Neural Networks: A Generalization of Multi-HyperGraph Neural Networks. *Neutrosophic Computing and Machine Learning* **2025**, *39*, 328–347.
10. Marcos, B.V.S.; Willner, M.F.; Rosa, B.V.C.; Yissel, F.F.R.M.; Roberto, E.R.; Puma, L.D.B.; Fernández, D.M.M. Using plithogenic n-SuperHyperGraphs to assess the degree of relationship between information skills and digital competencies. *Neutrosophic Sets and Systems* **2025**, *84*, 513–524.
11. Amable, N.H.; De Salazar, E.E.V.; Isaac, M.G.M.; Sánchez, O.C.O.; Palma, J.M.S. Representation of motivational dynamics in school environments through Plithogenic n-SuperHyperGraphs with family participation. *Neutrosophic Sets and Systems* **2025**, *92*, 570–583.
12. Berrocal Villegas, S.M.; Montalvo Fritas, W.; Berrocal Villegas, C.R.; Flores Fuentes Rivera, M.Y.; Espejo Rivera, R.; Bautista Puma, L.D.; Macazana Fernández, D.M. Using plithogenic n-SuperHyperGraphs to assess the degree of relationship between information skills and digital competencies. *Neutrosophic Sets and Systems* **2025**, *84*, 41.
13. Roshdy, E.; Khashaba, M.; Ali, M.E.A. Neutrosophic super-hypergraph fusion for proactive cyberattack countermeasures: A soft computing framework. *Neutrosophic Sets and Systems* **2025**, *94*, 232–252.
14. Fujita, T.; Mehmood, A. SuperHyperGraph Attention Networks. *Neutrosophic Computing and Machine Learning* **2025**, *40*, 10–27.
15. Fujita, T.; Smarandache, F. Superhypergraph Neural Networks and Plithogenic Graph Neural Networks: Theoretical Foundations. *Advancing Uncertain Combinatorics through Graphization, Hyperization, and Uncertainization: Fuzzy, Neutrosophic, Soft, Rough, and Beyond* **2025**, *5*, 577.
16. Hamidi, M.; Smarandache, F.; Davneshvar, E. Spectrum of superhypergraphs via flows. *Journal of Mathematics* **2022**, *2022*, 9158912.
17. Fujita, T.; Smarandache, F. Soft Directed n-SuperHyperGraphs with Some Real-World Applications. *European Journal of Pure and Applied Mathematics* **2025**, *18*, 6643–6643.
18. Fujita, T. Directed Acyclic SuperHypergraphs (DASH): A General Framework for Hierarchical Dependency Modeling. *Neutrosophic Knowledge* **2025**, *6*, 72–86.
19. Fujita, T. MetaHyperGraphs, MetaSuperHyperGraphs, and Iterated MetaGraphs: Modeling Graphs of Graphs, Hypergraphs of Hypergraphs, Superhypergraphs of Superhypergraphs, and Beyond. *Current Research in Interdisciplinary Studies* **2025**, *4*, 1–23.
20. Nacaroglu, Y.; Akgunes, N.; Pak, S.; Cangul, I.N. Some graph parameters of power set graphs. *Advances & Applications in Discrete Mathematics* **2021**, *26*.
21. Shalu, M.; Yamini, S.D. Counting maximal independent sets in power set graphs. *Indian Institute of Information Technology Design & Manufacturing (IIITD&M) Kancheepuram, India* **2014**.
22. Jech, T. *Set theory: The third millennium edition, revised and expanded*; Springer, 2003.
23. Bretto, A. Hypergraph theory. *An introduction. Mathematical Engineering. Cham: Springer* **2013**, *1*.
24. Berge, C. *Hypergraphs: combinatorics of finite sets*; Vol. 45, Elsevier, 1984.
25. Smarandache, F. *Extension of HyperGraph to n-SuperHyperGraph and to Plithogenic n-SuperHyperGraph, and Extension of HyperAlgebra to n-ary (Classical-/Neutro-/Anti-) HyperAlgebra*; Infinite Study, 2020.
26. Zadeh, L.A. Fuzzy sets. *Information and control* **1965**, *8*, 338–353.
27. Wang, H.; Smarandache, F.; Zhang, Y.; Sunderraman, R. *Single valued neutrosophic sets*; Infinite study, 2010.
28. Broumi, S.; Talea, M.; Bakali, A.; Smarandache, F. Single valued neutrosophic graphs. *Journal of New theory* **2016**, *10*, 86–101.
29. Smarandache, F. *Plithogenic set, an extension of crisp, fuzzy, intuitionistic fuzzy, and neutrosophic sets-revisited*; Infinite study, 2018.
30. Azeem, M.; Rashid, H.; Jamil, M.K.; Gütmen, S.; Tirkolaee, E.B. Plithogenic fuzzy graph: A study of fundamental properties and potential applications. *Journal of Dynamics and Games* **2024**, pp. 0–0.