# On Edge-MetaGraph

Takaaki Fujita *

Posted Date: 10 March 2026

doi: 10.20944/preprints202603.0710.v1

Keywords: Edge-MetaGraph; MetaGraph; Iterated MetaGraph

 doi:10.20944/preprints202603.0710.v1



*Article*

# On Edge-MetaGraph

**Takaaki Fujita**

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

**Abstract**

This paper studies graph-based higher-order structures related to metagraphs and edge-labeled hierarchical networks. After reviewing MetaGraphs and Iterated MetaGraphs, we introduce the notion of an Edge-MetaGraph, in which each edge is labeled by a two-ported internal graph, allowing edge-substitution expansion through port gluing. We then define Iterated Edge-MetaGraphs recursively, so that edges may carry nested Edge-MetaGraph structures. Concrete examples from biomedical systems, software pipelines, and logistics are presented to illustrate the expressive power of the proposed framework.

**Keywords:** Edge-MetaGraph; MetaGraph; Iterated MetaGraph

---

## 1. Preliminaries

This section establishes notation and summarizes the set-theoretic and combinatorial notions needed in the sequel. Throughout, all sets that serve as vertex domains are assumed to be finite, and we write $\mathbb{N}_0 := \mathbb{N} \cup \{0\}$.

### *1.1. MetaGraph (Graph of Graphs)*

Graph theory investigates mathematical structures consisting of vertices and edges to model relationships and connectivity [1,2]. A MetaGraph is a graph whose vertices are themselves graphs, with edges representing specified relations between those graphs (cf.[3–6]).

**Definition 1.1** (Metagraph (graph of graphs))**.** *[7] Fix a nonempty universe $\mathfrak{G}$ of finite graphs (undirected, loopless by default) and a nonempty family of binary relations*

$$\mathcal{R} \subseteq \mathcal{P}(\mathfrak{G} \times \mathfrak{G}).$$

*A* metagraph over $(\mathfrak{G}, \mathcal{R})$ *is a directed, labelled multigraph*

$$M = (V, E, s, t, \lambda)$$

*with*

$$V \subseteq \mathfrak{G}, \qquad s, t : E \to V, \qquad \lambda : E \to \mathcal{R},$$

*satisfying the incidence constraint*

$$\forall e \in E : \ \big(s(e),\, t(e)\big) \in \lambda(e).$$

*Elements of $V$ are* meta-vertices *(each is a graph $G \in \mathfrak{G}$). For $e \in E$ with $\lambda(e) = R$, we write $s(e) \xrightarrow{R} t(e)$ and call $e$ a* meta-edge*. If $\mathcal{R} = \{R\}$ is a singleton, labels may be omitted. If every $R \in \mathcal{R}$ is symmetric, $M$ can be viewed as an undirected labelled multigraph.*

**Remark 1.1.** *We fix a common base universe $\mathfrak{G}$ of finite, loopless graphs that encode biochemical modules. Each $G \in \mathfrak{G}$ comes with a vertex-labeling $\ell_V : V(G) \to \Sigma_V$ (e.g. metabolite/protein/gene names) and, when needed,*

 doi:10.20944/preprints202603.0710.v1



*an edge-labeling $\ell_E : E(G) \to \Sigma_E$ (e.g. reaction/interaction types). We use the following chemically meaningful binary relations on $\mathfrak{G}$:*

$$\begin{aligned} R_{\text{shareM}}(G,H) &:\iff \{\textit{metabolite names in } G\} \cap \{\textit{metabolite names in } H\} \neq \emptyset, \\ R_{\text{shareP}}(G,H) &:\iff \{\textit{protein names in } G\} \cap \{\textit{protein names in } H\} \neq \emptyset, \\ R_{\text{GRN}\Delta}(G,H) &:\iff E(G) \triangle E(H) \neq \emptyset \quad (\textit{directed GRNs; activation/inhibition labels allowed}). \end{aligned}$$

*Set $\mathcal{R} := \{R_{\text{shareM}}, R_{\text{shareP}}, R_{\text{GRN}\Delta}\}$.*

*1.2. Iterated MetaGraph (Graph of Graphs of ... of Graphs)*

An Iterated MetaGraph is a graph whose vertices are metagraphs, recursively extending graph-of-graphs structure to multiple hierarchical levels [7–9]. Related concepts, such as MetaStructures [10,11], are also known.

**Definition 1.2** (Unit metagraph embedding). *[7] For $X \in \mathfrak{G}$ define the* unit metagraph

$$\mathsf{U}(X) := (\{X\}, \emptyset, \_, \_, \_).$$

*This gives an injective map $\mathsf{U} : \mathfrak{G} \hookrightarrow \mathrm{Obj}\big(\mathrm{Meta}(\mathfrak{G}, \mathcal{R})\big)$.*

**Definition 1.3** (Relation lifting). *Given $\mathcal{R}$ on $\mathfrak{G}$, define its* lift $\mathcal{R}^{\uparrow}$ *on finite metagraphs over $(\mathfrak{G}, \mathcal{R})$ by*

$$\forall R \in \mathcal{R}, \quad (M_1, M_2) \in R^{\uparrow} \iff \exists\, x \in V(M_1),\, y \in V(M_2) :\ (x,y) \in R.$$

*Set $\mathcal{R}^{\uparrow} := \{R^{\uparrow} : R \in \mathcal{R}\}$.*

**Definition 1.4** (Iterated object and relation universes). *Define recursively for $t \in \mathbb{N}_0$:*

$$\mathfrak{G}^{(0)} := \mathfrak{G}, \qquad \mathcal{R}^{(0)} := \mathcal{R},$$

$$\mathfrak{G}^{(t+1)} := \Big\{ \textit{finite metagraphs over } \big(\mathfrak{G}^{(t)}, \mathcal{R}^{(t)}\big) \Big\}, \qquad \mathcal{R}^{(t+1)} := \big(\mathcal{R}^{(t)}\big)^{\uparrow}.$$

**Definition 1.5** (Iterated MetaGraph of depth $t$). *For $t \in \mathbb{N}_0$, an* iterated metagraph of depth $t$ *is a metagraph*

$$M^{(t)} = (V^{(t)}, E^{(t)}, s^{(t)}, t^{(t)}, \lambda^{(t)})$$

*over $\big(\mathfrak{G}^{(t)}, \mathcal{R}^{(t)}\big)$, i.e., $V^{(t)} \subseteq \mathfrak{G}^{(t)}$, $\lambda^{(t)} : E^{(t)} \to \mathcal{R}^{(t)}$ and*

$$\forall e \in E^{(t)} :\ \big(s^{(t)}(e),\, t^{(t)}(e)\big) \in \lambda^{(t)}(e).$$

**Remark 1.2.** *We use the relation lifting of the preliminaries: for $R \in \mathcal{R}$ and metagraphs $\mathsf{M}_1, \mathsf{M}_2$ over $(\mathfrak{G}, \mathcal{R})$,*

$$(\mathsf{M}_1, \mathsf{M}_2) \in R^{\uparrow} \iff \exists X \in V(\mathsf{M}_1),\, \exists Y \in V(\mathsf{M}_2) \textit{ such that } (X,Y) \in R.$$

**Example 1.1** (A concrete iterated metagraph (depth $t = 1$) with a biomedical interpretation). *We give an explicit example of an iterated metagraph of depth $t = 1$ (i.e., a graph whose vertices are metagraphs of base graphs).*

***Base level ($\mathfrak{G}^{(0)} = \mathfrak{G}$).*** *Let $\mathfrak{G}$ contain three pathway graphs*

$$G_1 = \textit{Glycolysis}, \qquad G_2 = \textit{Pyruvate Oxidation}, \qquad G_3 = \textit{TCA Cycle}.$$

 doi:10.20944/preprints202603.0710.v1



*Let* $\mathcal{R}^{(0)} = \{R_{\text{shareM}}\}$*, where*

$$(G, H) \in R_{\text{shareM}} \iff \textit{the pathways } G \textit{ and } H \textit{ share at least one metabolite.}$$

*Assume* $(G_1, G_2) \in R_{\text{shareM}}$ *(shared metabolite: pyruvate) and* $(G_2, G_3) \in R_{\text{shareM}}$ *(shared metabolite: acetyl-CoA).*

***Metagraphs of graphs (objects in*** $\mathfrak{G}^{(1)}$***).*** *Define two metagraphs over* $(\mathfrak{G}^{(0)}, \mathcal{R}^{(0)})$*:*

$$M_A^{(0)} : V(M_A^{(0)}) = \{G_1, G_2\}, \textit{ one meta-edge } G_1 \xrightarrow{R_{\text{shareM}}} G_2,$$

$$M_B^{(0)} : V(M_B^{(0)}) = \{G_2, G_3\}, \textit{ one meta-edge } G_2 \xrightarrow{R_{\text{shareM}}} G_3.$$

*By construction,* $M_A^{(0)}, M_B^{(0)} \in \mathfrak{G}^{(1)}$*.*

***Lifted relation and depth-1 iterated metagraph.*** *Since* $(G_2, G_3) \in R_{\text{shareM}}$ *with* $G_2 \in V(M_A^{(0)})$ *and* $G_3 \in V(M_B^{(0)})$*, the lifted relation satisfies*

$$(M_A^{(0)}, M_B^{(0)}) \in R^{\uparrow}_{\text{shareM}} \in \mathcal{R}^{(1)}.$$

*Hence the depth-1 iterated metagraph*

$$M^{(1)} = (V^{(1)}, E^{(1)}, s^{(1)}, t^{(1)}, \lambda^{(1)})$$

*can be defined by*

$$V^{(1)} = \{M_A^{(0)}, M_B^{(0)}\}, \quad E^{(1)} = \{e\}, \quad s^{(1)}(e) = M_A^{(0)}, \quad t^{(1)}(e) = M_B^{(0)}, \quad \lambda^{(1)}(e) = R^{\uparrow}_{\text{shareM}}.$$

*Figure 1.1 depicts this construction.*

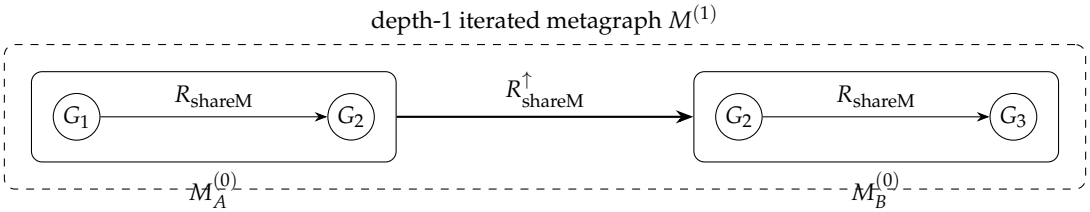


**Figure 1.1.** A depth-1 iterated metagraph: vertices are metagraphs $M_A^{(0)}, M_B^{(0)} \in \mathfrak{G}^{(1)}$, and the outer edge is labeled by the lifted relation $R^{\uparrow}_{\text{shareM}} \in \mathcal{R}^{(1)}$.

## 2. Results

The results of this paper are presented in this section.

### *2.1. Edge-MetaGraph (Graph-Labeled Edges)*

Edge-MetaGraph is a graph whose edges carry two-ported internal graphs, enabling edge-substitution expansion by gluing ports to endpoints.

**Definition 2.1** (Two-ported graphs)**.** *Fix a nonempty universe* $\mathfrak{G}$ *of finite (undirected, loopless by default) graphs. A* two-ported graph *is a triple*

$$H^{\partial} = (H, p^-, p^+),$$

 doi:10.20944/preprints202603.0710.v1



*where $H \in \mathfrak{G}$ and $p^-, p^+ \in V(H)$ are distinguished vertices called the* tail-port *and* head-port*, respectively. We write* $\mathrm{Tail}(H^\partial) := p^-$ *and* $\mathrm{Head}(H^\partial) := p^+$*, and denote by*

$$\mathfrak{G}^{\partial 2} := \{(H, p^-, p^+) : H \in \mathfrak{G},\ p^-, p^+ \in V(H)\}$$

*the class of all two-ported graphs over $\mathfrak{G}$.*

**Definition 2.2** (Edge-MetaGraph)**.** *Let $\mathfrak{G}^{\partial 2}$ be as in Definition 2.1. An* Edge-MetaGraph over $\mathfrak{G}$ *is a directed multigraph equipped with a two-ported-graph label on each edge, i.e., a tuple*

$$\mathbb{E} \;=\; (V, E, s, t, \ell),$$

*where $(V, E, s, t)$ is a directed multigraph (loops and parallel edges allowed) and*

$$\ell : E \to \mathfrak{G}^{\partial 2}$$

*is a labeling map. For $e \in E$, writing $\ell(e) = (H_e, p_e^-, p_e^+)$, the intended meaning is: the edge $e$ from $s(e)$ to $t(e)$ carries the internal graph $H_e$, whose ports $p_e^-, p_e^+$ will be attached to the endpoints $s(e), t(e)$, respectively.*

**Definition 2.3** (Expansion / edge-substitution)**.** *Let $\mathbb{E} = (V, E, s, t, \ell)$ be an Edge-MetaGraph and write $\ell(e) = (H_e, p_e^-, p_e^+)$. Define a graph* $\mathrm{Expand}(\mathbb{E})$ *(the* expansion *of $\mathbb{E}$) by gluing each labeled graph $H_e$ along its ports to the endpoints of $e$ as follows.*

*Take the disjoint union of vertex sets*

$$W \;:=\; V \;\sqcup\; \bigsqcup_{e \in E} V(H_e),$$

*and let $\sim$ be the smallest equivalence relation on $W$ such that for every $e \in E$,*

$$p_e^- \sim s(e), \qquad p_e^+ \sim t(e).$$

*Set $V(\mathrm{Expand}(\mathbb{E})) := W/\!\sim$. For edges, take the disjoint union*

$$F \;:=\; \bigsqcup_{e \in E} E(H_e),$$

*and for $f = \{x, y\} \in E(H_e)$ define its endpoints in $V(\mathrm{Expand}(\mathbb{E}))$ to be*

$$\big([x]_\sim, [y]_\sim\big).$$

*Then* $\mathrm{Expand}(\mathbb{E})$ *is the (multi)graph with vertex set $W/\!\sim$ and edge multiset induced from $F$.*

*(If $\mathfrak{G}$ is a universe of directed graphs, replace $\{x, y\}$ by ordered pairs $(x, y)$ throughout.)*

**Example 2.1** (Real-world example: software deployment pipeline with edge-level internal dependencies)**.** *Consider a company operating a microservice platform. At a coarse operational level, we describe the* deployment pipeline *as a directed multigraph*

$$(V, E, s, t),$$

*whose vertices are macro-stages:*

$$V = \{\mathit{Code},\ \mathit{Build},\ \mathit{Test},\ \mathit{Deploy},\ \mathit{Monitor}\}.$$

 doi:10.20944/preprints202603.0710.v1



*A directed edge $e : \mathit{Build} \to \mathit{Test}$ represents the handoff "send build artifacts to the test stage". However, this handoff is not atomic: it involves multiple tools and checks (artifact registry, signature verification, test orchestration, reporting).*

*To model such hidden internal structure, define an Edge–MetaGraph*

$$\mathbb{E} = (V, E, s, t, \ell), \qquad \ell : E \to \mathfrak{G}^{\partial 2},$$

*where each edge $e \in E$ is labeled by a two-ported internal graph*

$$\ell(e) = (H_e, p_e^-, p_e^+).$$

*For example, for the edge $e : \mathit{Build} \to \mathit{Test}$, take $H_e$ to be the graph whose vertices are*

$$V(H_e) = \{\mathit{ArtifactRepo},\ \mathit{SignatureCheck},\ \mathit{TestRunner},\ \mathit{ReportStore}\},$$

*and whose edges represent required interactions (e.g. artifact fetch, signature verification, executing tests, storing reports). Choose ports*

$$p_e^- = \mathit{ArtifactRepo} \in V(H_e), \qquad p_e^+ = \mathit{ReportStore} \in V(H_e),$$

*so that $p_e^-$ is the entry interface to the internal process (receiving artifacts) and $p_e^+$ is the exit interface (publishing test reports).*

*Similarly, for the edge $\mathit{Deploy} \to \mathit{Monitor}$, the internal graph $H_{e'}$ may encode*

$$\mathit{DeployController} \to \mathit{MetricsAgent} \to \mathit{AlertManager},$$

*with ports identifying the start/end of the deployment-to-observability integration.*

*Figure 2.1 visualizes the Edge–MetaGraph viewpoint: the outer pipeline is shown together with two illustrative internal graphs attached to edges. By Definition 2.3, the expanded graph $\mathrm{Expand}(\mathbb{E})$ is obtained by gluing each port $p_e^-$ to $s(e)$ and each port $p_e^+$ to $t(e)$.*

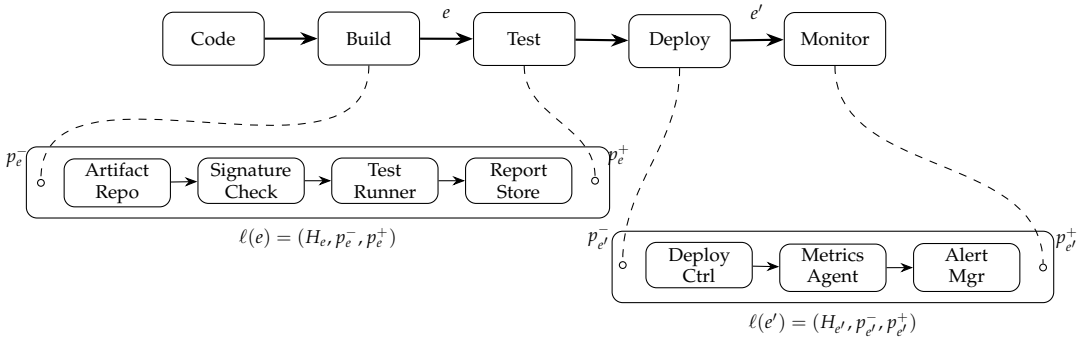


**Figure 2.1.** An Edge–MetaGraph for a CI/CD pipeline. The outer directed multigraph encodes macro-stages. Edges $e$ and $e'$ are labeled by two-ported internal graphs; dashed curves indicate the port-gluing used in $\mathrm{Expand}(\mathbb{E})$.

### *2.2. Iterated Edge-MetaGraphs*

Iterated Edge-MetaGraphs recursively label each edge by a two-ported Edge-MetaGraph, creating multi-level edge-substitution and expansion hierarchy.

**Definition 2.4** (Two-ported iterated Edge-MetaGraphs)**.** *Let $t \in \mathbb{N}_0$ and let $\mathfrak{E}^{(t)}$ be a class of finite objects each equipped with a finite vertex set $V(X)$. A* two-ported iterated Edge-MetaGraph of depth $t$ *is a triple*

$$X^{\partial} = (X, p^-, p^+), \qquad X \in \mathfrak{E}^{(t)},\ \ p^-, p^+ \in V(X),$$

 doi:10.20944/preprints202603.0710.v1



*where $p^-$ and $p^+$ are distinguished vertices called the* tail-port *and* head-port. *We write* $\mathrm{Tail}(X^\partial) := p^-$ *and* $\mathrm{Head}(X^\partial) := p^+$, *and set*

$$\left(\mathfrak{E}^{(t)}\right)^{\partial 2} := \{(X, p^-, p^+) : X \in \mathfrak{E}^{(t)},\ p^-, p^+ \in V(X)\}.$$

**Definition 2.5** (Iterated Edge-MetaGraph universes)**.** *Fix a nonempty universe $\mathfrak{G}$ of finite graphs (as in the preliminaries). Define recursively a hierarchy $\left(\mathfrak{E}^{(t)}\right)_{t\in\mathbb{N}_0}$ by*

$$\mathfrak{E}^{(0)} := \mathfrak{G},$$

*and for each $t \geq 0$,*

$$\mathfrak{E}^{(t+1)} := \left\{ (V, E, s, t, \ell) \;\middle|\; (V, E, s, t) \text{ is a finite directed multigraph and } \ell : E \to \left(\mathfrak{E}^{(t)}\right)^{\partial 2} \right\}.$$

*Thus, an object of $\mathfrak{E}^{(t+1)}$ is an Edge-MetaGraph whose edge labels are two-ported objects from the previous level $\mathfrak{E}^{(t)}$.*

**Definition 2.6** (Iterated Edge-MetaGraph of depth $t$)**.** *Let $t \in \mathbb{N}_0$. An* iterated Edge-MetaGraph of depth $t$ *is an element of $\mathfrak{E}^{(t)}$ from Definition 2.5.*

*Equivalently, for $t \geq 1$ it is a tuple*

$$\mathbb{E}^{(t)} = (V^{(t)}, E^{(t)}, s^{(t)}, t^{(t)}, \ell^{(t)})$$

*such that $(V^{(t)}, E^{(t)}, s^{(t)}, t^{(t)})$ is a finite directed multigraph and*

$$\ell^{(t)} : E^{(t)} \longrightarrow \left(\mathfrak{E}^{(t-1)}\right)^{\partial 2}.$$

*In particular, depth 1 coincides with the (non-iterated) Edge-MetaGraph concept where edges are labeled by two-ported base graphs.*

**Remark 2.1** (Recursive expansion (optional notation))**.** *Using the expansion operator* Expand *already defined for Edge-MetaGraphs, one may define a depth-t flattening* $\mathrm{Expand}^{(t)}$ *recursively by*

$$\mathrm{Expand}^{(0)}(G) := G, \qquad \mathrm{Expand}^{(t+1)}(\mathbb{E}) := \mathrm{Expand}\Big(V, E, s, t,\ e \mapsto \big(\mathrm{Expand}^{(t)}(X_e), p_e^-, p_e^+\big)\Big),$$

*where $\ell(e) = (X_e, p_e^-, p_e^+)$. This yields an ordinary graph obtained by expanding all nested edge labels down to depth 0.*

**Example 2.2** (Real-world example: multi-layer logistics & service dependencies)**.** *Consider an international e-commerce company that ships products from a factory to a customer. At the* top level *(depth 2), we model the end-to-end process as a directed multigraph*

$$(V^{(2)}, E^{(2)}, s^{(2)}, t^{(2)}),$$

*whose vertices are major stages:*

$$V^{(2)} = \{\mathit{Factory},\ \mathit{ExportHub},\ \mathit{ImportHub},\ \mathit{LastMile},\ \mathit{Customer}\}.$$

*An edge $e \in E^{(2)}$ represents a contractual handoff such as $\mathit{ExportHub} \to \mathit{ImportHub}$ or $\mathit{LastMile} \to \mathit{Customer}$.*

*Each such handoff is* not *atomic in practice: it is implemented by a nested workflow involving multiple carriers, checkpoints, and IT services. Hence we label every top-level edge e by a two-ported object*

$$\ell^{(2)}(e) = (X_e, p_e^-, p_e^+) \in \left(\mathfrak{E}^{(1)}\right)^{\partial 2},$$

 doi:10.20944/preprints202603.0710.v1



*where $X_e \in \mathfrak{E}^{(1)}$ is a depth-1 Edge–MetaGraph describing the internal sub-process realizing the handoff, and the ports $p_e^-, p_e^+$ indicate the entry/exit interface of that sub-process.*

*For instance, the edge $e$ : ExportHub $\to$ ImportHub may be labeled by a depth-1 Edge–MetaGraph $X_e$ whose vertices represent operational checkpoints*

$$V(X_e) = \{\mathit{PickUp},\ \mathit{OriginCustoms},\ \mathit{Airline},\ \mathit{DestinationCustoms},\ \mathit{Unload}\},$$

*and whose edges correspond to concrete tasks (document verification, container scanning, flight booking, etc.). Crucially,* each task edge $f \in E(X_e)$ *is again labeled by a two-ported* base graph

$$\ell_{X_e}(f) = (H_f, q_f^-, q_f^+) \in \left(\mathfrak{E}^{(0)}\right)^{\partial 2} = \mathfrak{G}^{\partial 2},$$

*where $H_f$ is a small graph encoding the micro-structure of the task, e.g. a dependency graph between IT services:*

$$\mathit{OrderDB} \to \mathit{RiskScoring} \to \mathit{CustomsAPI} \to \mathit{LabelPrinter}.$$

*Thus, the overall system forms an* iterated Edge–MetaGraph of depth 2*: top-level shipment handoffs are edges labeled by depth-1 Edge–MetaGraphs, and inside those, task edges are labeled by two-ported base graphs capturing service-level dependencies. Expanding the structure (recursively) yields a single flattened dependency graph for the entire factory-to-customer delivery pipeline.*

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

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

**Use of Artificial Intelligence:** LLM-based software was used for language polishing (e.g., grammar and clarity checks), formatting verification, and assistance in diagnosing potential LaTeX compilation errors. It was not used to produce results or in any manner that would compromise research integrity.

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

**Conflicts of Interest:** The author confirms that there is no conflict of interest related to the research or its publication.

**Ethical Approval:** As this research is entirely theoretical in nature and does not involve human participants or animal subjects, no ethical approval is required. I have not engaged in any unethical conduct.

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

## References

1. Diestel, R. *Graph theory*; Springer (print edition); Reinhard Diestel (eBooks), 2024.
2. Gross, J.L.; Yellen, J.; Anderson, M. *Graph theory and its applications*; Chapman and Hall/CRC, 2018.
3. Lima Filho, R.B.; Sabino-Silva, R.; Carneiro, M.G. High-level classification based on meta-graph of visibility graphs for Autism detection. In Proceedings of the 2025 International Joint Conference on Neural Networks (IJCNN). IEEE, 2025, pp. 1–8.
4. Azevedo, R.; Lohaus, R.; Paixao, T. Networking networks. *Evol Dev* **2008**, *10*, 882.

 doi:10.20944/preprints202603.0710.v1




5. Velazquez-Garcia, E.; Lopez-Arevalo, I.; Sosa-Sosa, V.J. Representing document semantics by means of graphs. In Proceedings of the 2011 8th International Conference on Electrical Engineering, Computing Science and Automatic Control. IEEE, 2011, pp. 1–6.
6. Donnat, C.; Holmes, S. Tracking network dynamics: A survey using graph distances. *The Annals of Applied Statistics* **2018**, *12*, 971–1012.
7. 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.
8. Fujita, T.; Smarandache, F. *Representing Higher-Order Networks: A Survey of Graph-Based Frameworks*; Neutrosophic Science International Association (NSIA) Publishing House, 2026.
9. Fujita, T.; Smarandache, F. *HyperGraph and SuperHyperGraph Theory with Applications*; Neutrosophic Science International Association (NSIA) Publishing House, 2026.
10. Fujita, T. Metastructure, meta-hyperstructure, and meta-superhyper structure. *Journal of Computers and Applications* **2025**, *1*, 1–22.
11. Fujita, T. Molecular MetaGraph and Molecular Iterated MetaGraph in Chemistryand BioChemistry. *Journal of Applied Mathematics and Symbolic Science* **2025**, *1*, 43–57.