HIGHER-ORDER GRAPH REPRESENTATION LEARNING NEEDS A ROSETTA STONE FOR PORTABILITY

Anonymous authors
Paper under double-blind review

ABSTRACT

The last decade has seen a rapid proliferation of non-Euclidean neural network architectures to address so-called higher-order network data exhibiting multiway interactions. These architectures run the gamut, differing in their choice of higher-order representations (e.g., hypergraphs and simplicial complexes), their mathematical vernacular (algebraic, topological, geometric), and their intended task (e.g., link prediction, node classification, or segmentation). To help navigate this diverse landscape, several detailed surveys on higher-order representations have recently appeared, attempting to categorize and taxonomize these architectures. Yet multiple blindspots remain, obstructing both the utility of higher-order network representation learning and the development of its associated theoretical guarantees. In this position paper, we argue that portability between higher-order architectures is one such critical, largely under-explored area, vital to the future of graph deep learning. Through the lens of this meta-analysis, we present a "Rosetta stone" for higher-order graph representation learning, highlighting the recent progress, connecting the mathematical dots among multiple fragmented frameworks, and offering a synergistic perspective on the open problems and future directions of non-Euclidean abstract representation learning.

1 INTRODUCTION

There has been a rapid proliferation of neural network architectures for learning non-Euclidean data such as graphs and manifolds, with the most recent focus on capturing the higher-order interactions inherent in these structures (Hajij et al., 2025; Abiad et al., 2026; Su et al., 2026; Peixoto et al., 2026; Petruso et al., 2026). The first major thrust to advance deep learning (DL) machinery to non-Euclidean objects in a systematic manner traces back to Bronstein et al. (2017) and has been coined *geometric deep learning* (GDL). However, it has been quickly recognized that non-Euclidean data often exhibit highly non-trivial interdependencies that go well beyond the conventional pairwise relationships typically addressed by GDL (Benson et al., 2018; Torres et al., 2021; Battiston et al., 2021). For example, in biomolecular graphs, we may observe multiway interactions between proteins that can influence phenotype and cellular functions. These can in turn be important to consider when identifying disease mechanisms and developing therapeutic strategies (Murgas et al., 2022). Similarly, money laundering and other fraudulent schemes in criminal and illicit networks can involve complex multiway interactions involving groups of transactions and addresses, and the functioning of this inherently hierarchical network is governed by these higher-order, multi-actor interactions (Ranshous et al., 2017; Ye et al., 2026; van der Zwet et al., 2025). In neuroscience, one often needs to account for various types of multiway interactions such as the co-firing of multiple neurons, or cognitive functions that are influenced by distributed sets of brain regions (Giusti et al., 2016; Li et al., 2026).

Given these and other examples of complex datasets characterized by multiway interactions, there has been growing interest in DL architectures that accommodate higher-order graph generalizations (Besta et al., 2026; Tian & Zafarani, 2024; Pellegrin et al., 2025). This includes, but is not limited to, modeling complex phenomena using hypergraphs, simplicial complexes, cell complexes, and differential forms (Su et al., 2026). Different representations are in turn more amenable to different mathematical toolkits, from tensor algebra to algebraic topology to discrete exterior calculus and more. To navigate this landscape of higher-order generalizations, many surveys have recently appeared that attempt to taxonomize different corners of higher-order representation learning (Schaub et al., 2021; Hajij et al., 2022; Papillon et al., 2023; Antelmi et al., 2023; Papamarkou et al., 2024; Tian & Zafarani, 2024;

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Kim et al., 2024; Telyatnikov et al., 2025; Besta et al., 2026; Su et al., 2026). While each offers a unique and helpful perspective, certain aspects inevitably remain siloed, fragmented, or have received little to no coverage across this body of work. Such knowledge gaps are further widened by the rapidly evolving nature of graph representation learning, obstructing both the utility of higher-order network representation learning and the development of its associated theoretical guarantees.

In light of the advent of world models and, in particular, Joint Embedding Predictive Architectures (JEPA) (LeCun, 2022) along with the associated emerging focus on generalization and abstract latent representation learning, we ask: *what do we actually know about the transferability (or portability) of higher-order graph representations?* JEPA's central departure from generative self-supervision is that it operates in the *latent representation space* rather than input space. That is, a context embedding predicts a target embedding within a shared latent space, decoupled from the raw encoding utilized to reach it. Structurally, this is the same question we ask on higher-order graph representations: *can a model trained on one algebraic instantiation, for example, boundary operators on simplicial complexes, be ported to another, for example, incidence structures on a hypergraph, via a shared latent space, rather than being restricted to the representation it has been trained on?* Remarkably, despite the deep interconnections among the mathematical tools used to characterize higher-order graph generalizations, transfer learning and portability between higher-order architectures remain yet largely unaddressed. Through the lens of a meta-analysis of the knowledge accumulated across surveys over the last five years, we aim to fill these knowledge gaps and connect the mathematical dots among multiple fragmented representation frameworks, highlighting the bridge between the unified perspective on higher-order graph representations and the emerging need for portability across them. Specifically, we focus on 10 concepts, namely, 4 types of data representations (hypergraphs, simplicial complexes, cell complexes, and differential forms), 3 characteristics (expressivity, spectral properties, and computational cost), and 3 notions of portability between representations (translation, pretraining, and transfer across representations). We motivate and accompany this meta-analysis by a controlled numerical experiment that demonstrates various nuances in higher-order representational learning, which arise even in the context of a relatively simple task. We conclude with a roadmap of open problems and future directions for abstract non-Euclidean representation learning, with particular focus on generalization and portability.

2 Taxonomy of the Higher-Order Graph Representations

We start from an overview of the mathematical abstractions for higher-order representations, their algebraic instantiations for DL, and the associated DL architectures.

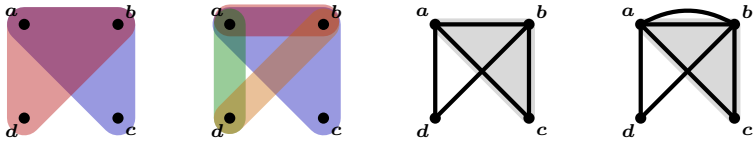
2.1 Higher-order networks abstractions

Higher-order networks are those that exhibit *multiway*, as opposed to strictly pairwise, relations. Though often discussed as a single entity, there is a rich diversity of mathematical approaches for describing them. Below, we emphasize this breadth and briefly overview the key representations for higher-order networks that are combinatorial (hypergraphs), topological (simplicial and cell complexes), and algebraic (cochain complexes).

- Perhaps the simplest combinatorial abstraction for higher-order networks is a **hypergraph** $H = (V, E)$ where V is a set of elements called vertices, and E is a collection of subsets of V. Hypergraphs in which every edge has exactly k elements are called k-uniform.
- A **simplicial complex** K on V is a hypergraph with downward closure: if $\sigma \in E(K)$, then for all nonempty $\tau \subseteq \sigma$, we have $\tau \in E(K)$. Elements of K are referred to as *simplices*, so that a hyperedge with $k+1$ vertices is called a k-dimensional simplex or k-simplex.
- Whereas in simplicial complexes a k-simplex must have exactly $k+1$ vertices and contain all sub-simplices as faces, a **cell complex** allows a k-cell to be bounded by an arbitrary collection of lower-dimensional cells. More formally, a *regular cell complex* X is a topological space of dimension n constructed inductively. That is, starting from a discrete set of 0-cells, each k-skeleton $X^{(k)}$ is formed by attaching k-dimensional cells (spaces homeomorphic to open k-balls) to the $(k-1)$-skeleton $X^{(k-1)}$ along their boundary, for $k = 1, \ldots, n$.
- A **cochain complex** (C, d) is an algebraic structure associated with a topological space, consisting of a sequence of vector spaces C^k connected by linear maps $d_k : C^k \to C^{k+1}$. Each vector space

Paper	Year	Authors
P1	2015	$\{a, b, c\}$
P2	2016	$\{a, b\}$
P3	2017	$\{a, d\}$
P4	2018	$\{b, d\}$

Raw co-authorships



(a) Triadic motif (b) Paper-as-hyperedge (c) Simplicial complex (d) Cell complex

Figure 1: Translation of raw dataset (left) into four distinct higher-order encodings (right).

represents the space of discrete differential k-forms, so that, for example, C^0 represents node features, C^1 represents edge flows, and C^2 are multiway fluxes. The operator d_k is the discrete exterior derivative, which maps k-forms to $(k+1)$-forms.

This list is incomplete and there are also directed, temporal, and multi-layer versions within each category. There are also proposed generalizations, like *combinatorial complexes* Hajij et al. (2022), for which hypergraphs, simplicial and cell complexes are special cases.

Example: Which Abstraction to Choose? An important consequence of this diversity is that there is no canonical choice for abstracting higher-order network data. In practice, a single raw dataset often affords representation via multiple of the aforementioned structures. The "correct" encoding is context-dependent, driven by the intended task, available computational resources, and geometric priors required by the neural architecture. To make this representational ambiguity concrete, consider a toy co-authorship dataset. Suppose we have the four authors, a, b, c, d, and four papers published as shown in the left side of Figure 1. Here, ignoring higher-order interactions would yield the complete graph on 4 vertices minus the $\{c, d\}$ edge. However, if we need to capture the multiway nature of collaborations, there are multiple abstractions we could use:

- **Triadic motif (3-uniform hypergraph):** A common practice is to lift motifs, like triangles in the projected graph, to a 3-uniform hypergraph. This captures triadic closure but loses fidelity, failing to distinguish a single 3-author paper from three disjoint 2-author publications.
- **Paper-as-hyperedge (non-uniform hypergraph):** Mapping the data directly treats each paper as a set of authors, which is a hyperedge.
- **Simplicial complex:** Forming a simplicial complex enforces downward closure, geometrically rendering 3-author papers as filled 2-dimensional faces. However, this mathematical rigidity forces the creation of unobserved pairwise ties (like $\{a, c\}$) and absorbs independent 2-author papers into the boundaries of larger simplices. At the same time, the additional structure provided by the simplicial complex makes it possible to leverage algebraic tools from topological data analysis that do not directly apply to hypergraphs.
- **Regular cell complex:** Building a cell complex, the 3-author paper $P_1 = \{a, b, c\}$ is a 2-cell, which must still be attached along the 1-cells $\{a, b\}$, $\{b, c\}$, $\{a, c\}$. Unlike a simplicial complex, a cell complex allows distinct 1-cells with the same endpoints, so P_2 remains its own 1-cell.
- **Cochain complex:** To process directional flows, we can overlay an algebraic orientation (such as author contribution orderings) onto the topological space. The discrete exterior derivative d_k then maps lower-dimensional pairwise signal flows into higher-dimensional fluxes, while preserving the fundamental chain complex property $d \circ d = 0$.

As this toy example illustrates, the boundaries between these representations are not inherent to the raw data, but are instead structural priors imposed by the modeler. Next, we discuss how these abstractions can be instantiated so they can be consumed by neural networks.

2.2 From Abstractions to Algebraic Instantiations for Deep Learning

While the choice of higher-order abstraction (e.g., hypergraph versus simplicial complex) determines the geometric boundaries of the data, practitioners face a subsequent and equally consequential choice: *how to mathematically encode this structure for a neural network?* A single conceptual representation can be instantiated via multiple distinct algebraic objects, i.e. matrices, tensors, or

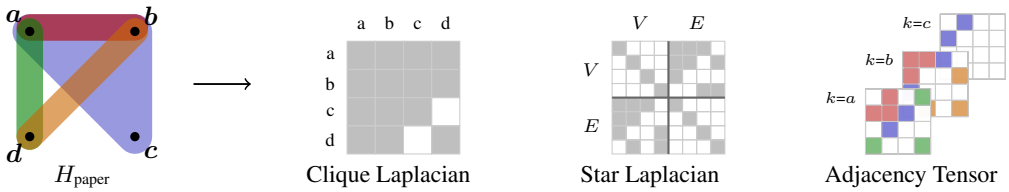


Figure 2: Three types of algebraic instantiations of the same paper-as-hyperedge abstraction H_{paper}.

boundary operators, each carrying fundamentally different structural priors, expressivity ceilings, and spectral properties.

Continuing our co-authorship example, suppose we have settled on the paper-as-hyperedge abstraction H_{paper}. To feed this into a DL architecture, we must translate the sets into an algebraic object. Figure 2 illustrates several categories of choices here. The most ubiquitous approach is to construct a hypergraph Laplacian derived from a *clique-expansion* graph, which is obtained by replacing hyperedges with (potentially edge-weighted) cliques, yielding the leftmost sparsity pattern (see Figure 2. While computationally convenient and directly amenable to standard graph convolutional layers, these matrix representations inherently project multiway interactions down to pairwise summaries, introducing documented structural ambiguity (Kim & Kirkland, 2022). Furthermore, though many such Laplacian variants have been defined, much of this apparent diversity is superficial: at the spectral level, many of these descriptors are equivalent. Indeed, as shown by Agarwal et al. (2006), the hypergraph Laplacians of Zhou et al. (2006), Bolla (1993), and Rodríguez (2002) are equivalent up to shifts and scalings. We emphasize that this imposes *an architectural ceiling for DL*: **any hypergraph neural network whose forward pass factors through one of these descriptors inherits the collapse and cannot distinguish hypergraphs that the expansion graph cannot**.

A second approach is to construct a Laplacian associated with the bipartite graph obtained by interpreting vertices as one color class, hyperedges, as the other, and vertex-hyperedge memberships as the edge criterion. The middle panel of Figure 2 shows the sparsity pattern for these matrices of dimension $|V| + |E|$ for the toy example. Sometimes called the "star expansion" approach, this has been utilized in contexts where one wishes to jointly analyze both vertices and hyperedges without distinguishing them, such as in co-clustering (Zhu et al., 2021). This approach effectively models incidences between vertices and hyperedges using pairwise relations, but still does not address *how to model multiway adjacency between groups of vertices*.

For that purpose, tensor representations appear to be a natural choice for representing hypergraphs. Rather than decomposing a hyperedge into pairwise edges, a tensor encoding records the hyperedge directly, as nonzero entries indexed by orderings of its vertices. This route, however, poses conceptual and computational obstacles. First, nonuniform hypergraphs admit no canonical tensor representation. One approach is to unbiasedly inflate hyperedges up to the size of the maximum hyperedge size (i.e. rank), as in the Banerjee-Char-Mondal adjacency tensor (Banerjee et al., 2017). For a hyperedge e in a rank r hypergraph, its *blowup* is the set of r-tuples of vertices of e in which every vertex of e appears at least once. The Banerjee tensor $\mathcal{A}$ is then an order r tensor which places nonzeros in all indices corresponding to hyperedge blowups. Figure 2 (right panel) illustrates this pattern on the tensor slices for the example. However, this definition is difficult to work with in practice, since $\mathcal{A}$ has $O(n^r)$ entries. To address these computational bottlenecks, Aksoy et al. (2024) develop implicit approaches for primitives like tensor-times-same-vector algorithms on the Banerjee tensor that run in low-degree polynomial time in r without ever forming the order-r array. In turn, T-HyperGNNs (Wang et al., 2025) port a similar principle to the learning setting with compressed adjacency tensors and a tensor message-passing layer. We provide a more detailed discussion on tensors, Laplacians, and other associated algebraic operators for higher-order networks in Appendix A.

2.3 Higher-order Graph Learning Model Architectures

Neural message passing on hypergraphs Messages are first transmitted from nodes to incident hyperedges, where each hyperedge aggregates incoming node messages to update its own representation $\mathbf{x}_e^{\text{new}}$ using the aggregation operator $\odot$. The updated hyperedge representations are then used to construct hyperedge messages, i.e., propagating back to all incident nodes and are aggregated

using the permutation-invariant operator $\oplus$ to update the node representations. Different hypergraph neural architectures adopt different message functions and aggregation operators, but they generally follow the same node–hyperedge–node message passing paradigm. Formally, for a hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E}, \mathbf{x})$, during one message passing step, node and hyperedge features are updated as follows:

$$\mathbf{x}_e^{\text{new}} = \phi_{\mathcal{E}}\Big(\mathbf{x}_e, \bigodot_{u \in \mathcal{V}:\, u \subseteq e} \psi_{\mathcal{V}}(\mathbf{x}_u)\Big), \quad \mathbf{x}_v^{\text{new}} = \phi_{\mathcal{V}}\Big(\mathbf{x}_v, \bigoplus_{e \in \mathcal{E}:\, v \subseteq e} \psi_{\mathcal{E}}(\mathbf{x}_e^{\text{new}})\Big). \tag{1}$$

where $v \in \mathcal{V}$, $e \in \mathcal{E}$, $\phi_{\mathcal{V}}, \phi_{\mathcal{E}}$ denote learnable functions, and $\bigodot$ and $\bigoplus$ denote permutation-invariant aggregation operators that aggregate messages from neighboring nodes and incident hyperedges, respectively. The above presents a generic message-passing framework for hypergraph neural networks, which serves as a unified abstraction for a broad family of architectures.

Neural message passing on simplicial complexes The convolution operations in this model can be understood as follows: at layer l, a simplicial convolutional filter (SCF) $\mathbf{H}_k^l$ updates features on k-dimensional simplices by taking a learned linear combination of their previous-layer features and signals propagated through lower and upper adjacency relations over up to T_d and T_u hops, respectively; $k-1$ and $k+1$-simplicial $\mathbf{x}_{k-1}^{l-1}$ and $\mathbf{x}_{k+1}^{l-1}$, are first projected onto the k-simplicial domain before being processed by the lower and upper simplicial convolution filters, respectively. In an SC, we take $\mathbf{x}_{k-1}^{l-1}$, $\mathbf{x}_k^{l-1}$, and $\mathbf{x}_{k+1}^{l-1}$ as inputs, the MPSC at layer $l = 1, \ldots, L$ computes the k-simplicial output $\mathbf{x}_k^l$ via the map:

$$\mathbf{x}_{k,d}^{l-1} = \mathbf{B}_k^\top \mathbf{x}_{k-1}^{l-1}, \quad \mathbf{x}_{k,u}^{l-1} = \mathbf{B}_{k+1}\mathbf{x}_{k+1}^{l-1}, \quad \mathbf{x}_k^l = \sigma\Big(\mathbf{H}_{k,d}^l \mathbf{x}_{k,d}^{l-1} + \mathbf{H}_k^l \mathbf{x}_k^{l-1} + \mathbf{H}_{k,u}^l \mathbf{x}_{k,u}^{l-1}\Big), \text{ with}$$

$$\mathbf{H}_k = \sum_{t=0}^{T_d} w_{k,d,t} \mathbf{L}_{k,d}^t + \sum_{t=0}^{T_u} w_{k,u,t} \mathbf{L}_{k,u}^t, \quad \mathbf{H}_{k,d} = \sum_{t=0}^{T_d} w'_{k,d,t} \mathbf{L}_{k,d}^t, \quad \mathbf{H}_{k,u} = \sum_{t=0}^{T_u} w'_{k,u,t} \mathbf{L}_{k,u}^t. \tag{2}$$

Based on Figure 1, hypergraph message passing constructs separate representations for the papers $P_1 = \{a, b, c\}$, $P_2 = \{a, b\}$, and $P_3 = \{a, d\}$, then combines their messages at author (a). This preserves the distinction between a joint three-author collaboration and an independent two-author publication during message construction. Whether the update exploits this distinction depends on the aggregation functions. For example, nonlinear transformations applied within each hyperedge can make the output sensitive to paper-specific groupings, whereas purely linear sum aggregation without additional hyperedge features can reduce to weighted pairwise propagation.

Simplicial message passing organizes these interactions through shared faces and cofaces. For edge features, i.e., $k = 1$, edges $\{a, b\}$ and $\{a, c\}$ are both lower- and upper-adjacent: they share vertex a and belong to the filled triangle $\{a, b, c\}$. By contrast, $\{a, d\}$ and $\{b, d\}$ are lower-adjacent through d, but have no shared triangular coface. The separate filters in Eq. 2 can therefore assign different transformations to these relationships. However, the independent publication P_2 and the boundary edge $\{a, b\}$ occupy the same simplex unless additional provenance features distinguish their roles. Hence, hypergraph updates aggregate information through individual publications, while simplicial filters propagate it through boundary and coface relations, and each architecture can exploit only the distinctions retained by its encoding.

Numerical experiment: When does message passing use higher-order structure? We now perform a controlled experiment to examine whether a simple message-passing rule can distinguish two subtly-different hypergraphs with the same pairwise summaries. More formally, we consider *Gram mates* (Kim & Kirkland, 2022), pairs of hypergraphs with incidence matrices S and R that satisfy $SS^\top = RR^\top$ and $S^\top S = R^\top R$. Combinatorially, this means the hypergraphs have equal vertex co-occurences and hyperedge-overlaps and thus identical "underlying graphs". We consider the smallest nonisomorphic Gram mates from Aksoy et al. (2024) with hyperedges:

$$E_S = \{\, \{1,2,3\},\ \{1,2,4\},\ \{2,3,4,5\},\ \{2,3,4,6\},\ \{3,4,5,6\},\ \{5,6\} \,\},$$
$$E_R = \{\, \{3,5,6\},\ \{4,5,6\},\ \{2,3,4,5\},\ \{2,3,4,6\},\ \{1,2,3,4\},\ \{1,2\} \,\}.$$

To test whether a propagation rule responds to this difference, we supply both hypergraphs with the same vertex features $x \in \mathbb{R}^6$ and compute

$$z_S = S\,\sigma(S^\top x), \qquad z_R = R\,\sigma(R^\top x),$$

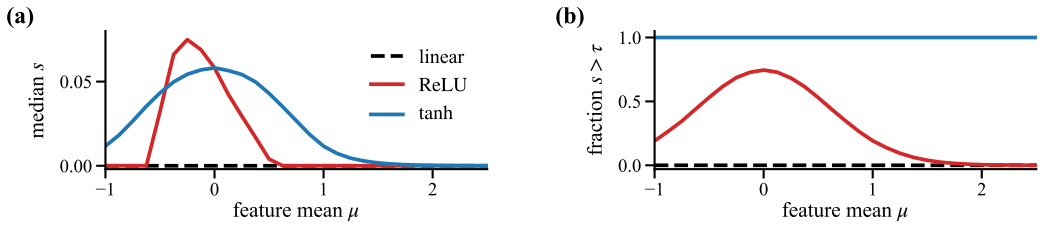


Figure 3: **Distinguishing Gram-mate hypergraphs depends on the activation and input features.** As the feature mean μ varies, (a) median normalized output distance s and (b) fraction of draws with $s > 10^{-12}$, for identity, ReLU, and $\tanh$ activations.

so that this update first sums vertex features within each hyperedge, applies an activation σ to each aggregate, and then sums the resulting hyperedge contributions at each vertex. By definition, under the identity activation this reduces to $z_S = SS^\top x = RR^\top x = z_R$ for every input, meaning that only passing through hyperedges is thus insufficient. Instead, we consider a nonlinear activation to prevent this collapse by transforming each hyperedge's aggregate before contributions from different hyperedges are summed at each vertex. To test when this nonlinearity yields different outputs for the two hypergraphs, we compare identity, ReLU, and $\tanh$ activations while varying the input features. For each mean, we draw 60,000 feature vectors $x \in \mathbb{R}^6$ with independent entries $x_i \sim \mathcal{N}(\mu, 1)$, supplying the same vector to both hypergraphs. We measure their output difference by $s = \frac{\|z_S - z_R\|_2}{\|z_S\|_2 + \|z_R\|_2}$, setting $s = 0$ when both outputs vanish and retaining these draws. Figure 3 reports both the magnitude of the difference and the fraction of draws on which it is detected, using the threshold $s > \tau$ for $\tau = 10^{-12}$. Outputs are compared under the fixed vertex correspondence.[1]

As expected since $SS^\top = RR^\top$, unnormalized linear propagation gives identical outputs throughout. While the nonlinear updates can distinguish the two hypergraphs, their responses depend on the features. ReLU selectively suppresses negative hyperedge aggregates, allowing different group memberships to produce different outputs. As μ increases, however, the aggregates increasingly become nonnegative. On draws where all aggregates in both hypergraphs are nonnegative, ReLU acts as the identity and the outputs agree. Its median separation therefore reaches zero even while some draws still produce differences. Lastly, $\tanh$ produces detectable differences on essentially every sampled input in the tested range, although their magnitude decreases at larger positive feature means, reflecting the near-constant behavior of $\tanh$ away from zero.

This experiment shows **why the use of higher-order structure cannot be judged from representations alone**. Here, the same incidence matrix encoding (that faithfully encodes hypergraphs) supports both an update that discards or exposes the Gram-mate hypergraph distinction. This is not solely an issue of matrices-vs-tensors, since the distinction can become visible without changing the encoding. The relevant unit of analysis is the combination of representation choices, propagation rules, and input features. This motivates the meta-analysis in the next section: interpreting performance and progress in higher-order neural networks requires examining which of these choices change together and how.

3 Meta-analysis: What We Know and What We Do Not

As the preceding discussion has exemplified, higher-order DL faces a wilderness of modeling choices. From selecting a mathematical abstraction, to instantiating it algebraically so that it can be consumed by an architecture, to determining a message passing scheme, each of these decision points presents many options which are sometimes subtly different yet consequential in ways that are not well-understood. Accordingly, the last five years has produced a steady stream of detailed papers that aim to provide guidance by surveying, taxonomizing, or otherwise unifying higher-order neural networks.

Here, we conduct a meta-analysis of these surveys in an attempt to take stock of current progress and to highlight persistent and new blindspots. We consider papers that meet three criteria:

[1] On the sampled inputs, sum pooling and comparison of the sorted vertex outputs detected differences on the same draws as the vertexwise comparison.

(1) they explicitly define the underlying higher-order structure considered (e.g., hypergraphs, simplicial complexes, etc.); (2) they primarily aim to survey, categorize, and/or generalize the field; and (3) they focus on neural networks, rather than solely on the mathematical object itself (e.g., while there are many surveys on hypergraphs, we only consider those concerning hypergraph neural nets). Table 1 lists the 10 papers found that fit these criteria, which we critically evaluate across three axes:

For the **representation** axis, we ask *what higher-order abstractions is the neural network based on*. We organize into the four categories discussed in Section 2, as hypergraphs, simplicial complexes, cell/combinatorial complexes, and differential forms/co-chain complexes. Though, for example, all simplicial complexes are also hypergraphs, a survey on simplicial neural networks would need to utilize the formalism, machinery, or combinatorial perspective of hypergraphs to be labeled as such.

For the **consequences** axis, we ask if *consequences of the design choices made are discussed or analyzed*. We break this down into three subareas. *Expressivity* is an account of which structures a model can and cannot tell apart, like isomorphism based expressivity tests (e.g. the Weisfeiler-Leman hierarchy) and motif count based expressiveness. We are interested expressivity notions that extend beyond, for example, the observation that the hypergraph abstraction can directly capture multiway interactions in a way that a graph cannot. By *spectral*, we mean an analysis of the eigenvalues or eigenvectors of the algebraic instantiation utilized, i.e. the operators a model propagates through. Third, by *cost* we mean any computational complexity or implementation analysis, whether focused on memory efficiency, runtime analysis, scaling studies, distributed implementations, and so on.

Finally, our last axis of **portability** concerns discussion of *how to translate, reuse, or port over parts of the pipeline*. We look for whether there is a discussion of *translation* between higher-order structures, or down to lower ones (e.g., converting a hypergraph to a simplicial complex or down to a clique expansion). Finally, we also look for the ability to *pretrain on the same representation*, or to conduct *transfer learning* across different higher-order representations. Table 1 presents the results for each of the above areas, scored using three levels: substantive (meaning dedication section or sustained treatment of the topic), partial (meaning mentioned briefly or in passing) and absent.

Table 1: Coverage of the works whose purpose is to organize higher-order deep learning. ● substantive, meaning a dedicated section or sustained treatment; ○ partial; – absent.

	Representations				Consequences			Portability		
	Hyper-graphs	Simpl. cplx.	Cell / comb.	Diff. forms	Expres-sivity	Spec-tral	Cost	Trans-lation	Pre-train	Trans-fer
Schaub (2021)	●	●	–	○	–	●	–	○	–	–
Hajij (2022)	○	●	●	○	–	–	–	●	–	–
Papillon (2023)	●	●	●	–	–	–	–	○	–	–
Antelmi (2023)	●	–	–	–	–	●	○	●	–	–
Papamarkou (2024)	○	●	●	–	●	–	●	○	○	–
Tian (2024)	●	●	–	○	–	○	○	●	–	–
Kim (2024)	●	–	–	–	–	–	–	●	●	–
Telyatnikov (2025)	●	●	●	–	–	–	●	●	–	–
Besta (2026)	●	●	●	–	●	–	●	●	–	–
Su (2026)	○	●	–	●	–	●	–	○	–	–

Takeaways from the meta-analysis Read together, the ten surveys in Table 1 reveal three striking gaps. First, while the union of all the surveys covers an expansive range of topics, individually these works are more siloed. For example, Kim et al. (2024) catalogue more than thirty hypergraph architectures without once mentioning simplicial complexes, while conversely, Su et al. (2026) develop differential forms and simplicial complexes at length and treat hypergraphs only in passing. This suggests that while research on higher-order learning has considerable depth, this depth still falls within the confines of one's preferred higher-order vernacular. Here, hypergraphs and simplicial complexes are the most popular structures, followed by cell and combinatorial complexes and finally differential forms which are almost always either absent or discussed only in passing.

Second, while *all* surveyed papers deal with the topic of translating between higher-order networks and other structures, only two analyze the expressivity of models by characterizing what structures they can tell apart. This disparity is notable in that expressivity and translation are interdependent

concerns: **the decision to translate a hypergraph to either a simplicial complex or down to clique expansion may depend on how well models can discern structural differences between members of these classes**. Despite their value in choosing among higher-order abstractions, expressivity results are hard to establish. Existing results often take the form of giving specific pairs of example structures that cannot be discerned by a model or approach. However, for higher-order neural networks, more general and analytic statements about expressivity are substantially rarer in the literature.

Third, while the surveys consider the portability of the *data* via translation between structures before training begins, they do not consider the portability of the *models*. This model portability can take several forms. First, one can change the task while treating the representation as fixed, pretraining a model on a large unlabeled corpus and fine-tuning it downstream. Only Kim et al. (2024) treats this substantively and Papamarkou et al. (2024) in passing. Second, none of the surveys handle transfer learning, which would change the representation itself. For example, a hypergraph neural network propagates signal through incidences, while one treated on simplicial complexes propagates through boundary operators, so these two models have no shared parameters. A theory of transfer learning and portability that sorts out how to move between these models is absent from the literature.

4 Why Portability Matters for Higher-Order Graph Representation Learning: Motivation and Open Problems

Despite its omission from the current literature, transfer learning across higher-order representations matters to the field for several reasons. The most immediate and obvious is economic: higher-order models are expensive to train. One who commits to one representation but later finds another better suited to their task presently has no recourse but to rebuild and train again from scratch. Second, transfer learning could help bridge the ecosystems surrounding higher-order abstractions, which often differ. The software, feature constructions, and downstream analyses available for hypergraphs are not those available for simplicial complexes. A practitioner may need to move a model into a given representation simply because tooling for the next stage of their analysis lives there. Third, transfer learning could help us understand, both experimentally and analytically, how well certain higher-order structures are suited for particular learning tasks. For example, consider the intuition that the downward closure of a simplicial complex is well-suited for link prediction, making simplicial complexes a de facto choice for this task (Benson et al., 2018). This is difficult to rigorously test today, because changing the representation also implies training a new model, confounding the representation with the training run. By allowing what a model has learned to be fixed, while we vary the representation, transfer learning could help turn such claims into empirical questions.

However, realizing these benefits for higher-order graph representations entails a series of open questions related to roadblocks on the path toward portability and generalization across representations:

Open Problem 1. Toward a Higher-Order Graph JEPA: Representation-Agnostic Latent Spaces. Can a shared latent space be learned across higher-order graph representations that disagree on the ground truth itself, rather than merely offering partial views of a shared ground truth? In contrast to current JEPA frameworks with the proven success for audio and video, which rely on context and target being reconcilable partial views of the same underlying sample (e.g., masked patches of one image), a hypergraph and a simplicial complex built from the same raw data are not necessarily views of one another. That is, they can encode genuinely different, sometimes irreconcilable structural commitments. (Indeed, recall the toy example in Fig. 1, where the simplicial complex hallucinates pairwise ties absent from the hypergraph.) Is it possible to define a shared latent space that both are "views of," despite this disagreement? And if so, can such distinct representations be made to complement one another, rather than simply be reconciled?

Open Problem 2. Random Walks and Efficient Exploration of Higher-Order Graph Substructures. Are random walks on simplicial, tensor, and hypergraph representations equivalent in any formal sense? If so, under what conditions? Can such equivalencies (or the lack thereof) be exploited for more efficient graph exploration, inform the design of corresponding DL architectures, or serve as a foundation for shared latent space representations?

Open Problem 3. Network Dimension and Generalization Bounds. What are the ways to characterize network dimension across different higher-order representations, and how consistent are these characterizations with one another? How do these various notions of dimension relate to

generalization bounds for DL models trained on them? And how might they inform the appropriate dimension of a shared latent space?

Open Problem 4. Spectral Properties. How do spectral properties of one higher-order representation relate to those of another? For example, how do Cheeger-type inequalities on hypergraphs and simplicial complexes relate to tensor eigenvalues, particularly, the Z-eigenvalues of the associated adjacency tensor? Can such results inform the design of more efficient, data- and task-specific DL architectures, and could spectral correspondences across higher-order graph representations serve as an analytically grounded target for the shared latent space posed in Open Problem 1?

Open Problem 5: Understanding expressivity in the context of transfer learning. What effect does transfer learning have on the expressivity of the models involved? As discussed earlier (see Subsection 2.2), certain higher-order architectures are unable to distinguish, for example, between hypergraph structures that share the same graph projections. In particular, since the predictor only operates on the encoder's output, it cannot recover information the encoder has already discarded. Consequently, even transferring a pretrained model to a target representation that supports finer distinctions, such as a hypergraph adjacency tensor, does not ameliorate this ceiling if the source-side encoder has collapsed the relevant structure before alignment. This raises two open questions: does joint-embedding training across representations simply inherit the ceiling of whichever encoder is least expressive? Or can training a shared predictor across genuinely complementary representations (Open Problem 1) raise the effective expressivity beyond what either representation supports alone?

5 Conclusion, Limitations, and Future Prospects

More than a growing catalogue of models, *higher-order learning needs principles for choosing among them*. We advance this goal by distinguishing three decisions that are often conflated: *how to represent multiway interactions abstractly*, *how to instantiate that representation algebraically*, and *how to learn from it*. We argue these choices must be evaluated together. Indeed, as Section 2 illustrates, they determine not only what structure is retained or imposed, but also which mathematical tools and learning strategies become available, with consequences for expressivity, spectral properties, computational cost, and transfer. One of our main contributions is to establish portability across higher-order representations as a research priority, to connect it with the emerging JEPA paradigm and the associated shared latent space, and to identify the obstacles that any principled account of portability must address. Our open problems turn this observation into specific research directions: identifying what translations preserve, investigating spectral correspondences as targets for cross-representation alignment, and determining when shared-embedding approaches can exploit complementary structure rather than reproduce an encoder's limitations. Together, these directions move the discussion beyond cataloguing architectures toward understanding *when*, *how*, and *at what cost* they can be used together.

Since our analysis is bounded by the literature and representations considered, it offers a research agenda, not a universal ranking of models or a validated procedure for selecting and transferring them. The proposed connections require task-specific theoretical and empirical testing. Neither successful alignment nor a shared latent space alone establishes that task-relevant structure has been preserved. We also need more progress on controlled benchmarks that separate the effects of representation, algebraic instantiation, and learning architecture; explicit accounts of what translations preserve or discard; and negative results that establish when higher-order structure or cross-representation transfer offers no benefit. Going beyond deciding whether a problem is "really" a hypergraph or a graph, we need to establish which choices help, under what conditions, and at what cost.

Nowadays, these fundamental questions gain urgency as scientific workflows increasingly rely on agentic AI rather than on humans to help formulate hypotheses, select experiments, and revise models as evidence accumulates. In an automated laboratory, new measurements or changing scientific questions may call for a different representation of the same underlying system. An agent must then decide not only which model to use, but whether existing representations and pretrained components remain useful, what information a translation would lose, and when retraining is necessary. Principled portability could make these decisions more systematic and auditable, allowing scientific knowledge encoded in models to be repurposed without assuming that different mathematical descriptions are interchangeable. Hence, the ultimate goal to which this paper aims to contribute is to replace today's human intuition-driven choices with automated ones, making model selection and adaptation responsive to evidence.

AI USE STATEMENT

The authors retain full responsibility for the paper's content and scientific integrity. LLM assistance was used to write Python scripts to execute the experiment in Section 2.3, to assist in drawing TikZ figures and formatting tables, and to conduct editorial revisions of author-written text, including grammar correction and improvements to clarity, conciseness, and presentation. All scientific contributions including the proposed taxonomy, evaluation of existing literature, proposed open problems, experimental design, and conclusions were developed and verified by the authors.

REPRODUCIBILITY STATEMENT

This paper provides mathematical definitions, toy constructions, and references to the underlying methods. The illustrative experiments are accompanied by descriptions of their structural inputs and evaluation procedures. These examples are intended to demonstrate specific representational distinctions rather than establish broad empirical performance claims.

ETHICS STATEMENT

This paper promotes informed and transparent use of higher-order graph learning by clarifying how representation choices affect the information preserved and transferred across models. These insights may support applications in biomedical research, complex system, and social network analysis. Because each representation emphasizes different structural properties and modeling assumptions, its suitability should be assessed in the context of the underlying data and intended application, with attention to potential biases and the interpretation of downstream results.

REFERENCES

Aida Abiad, Alex Arenas, Ágnes Backhausz, József Balogh, Christopher R. S. Banerji, Sergio Barbarossa, Ginestra Bianconi, Christian Bick, Magnus Botnan, Timoteo Carletti, Lucia Cavallaro, Andrea Civilini, Tina Eliassi-Rad, Xue Gong, Krystal Guo, Heather Harrington, Jürgen Jost, P. L. Krapivsky, Pietro Liò, Ben D. MacArthur, Carolina Mattsson, Pedro A. M. Mediano, Ana P. Millan, Raffaella Mulas, Alice Patania, Giovanni Petri, Cerene Rathilal, Rubén J. Sánchez-García, Martina Scolamiero, Michael T. Schaub, Hanlin Sun, Yu Tian, Francesco Vaccarino, and Kelin Xia. Hypergraphs and simplicial complexes in focus: a roadmap for future research in higher-order interactions. *Journal of Physics: Complexity*, 7(2):022501, 2026. doi: 10.1088/2632-072X/ae3c4e.

Sameer Agarwal, Kristin Branson, and Serge Belongie. Higher order learning with graphs. In *Proceedings of the 23rd International Conference on Machine Learning (ICML)*, pp. 17–24, 2006.

Sinan G. Aksoy, Ilya Amburg, and Stephen J. Young. Scalable tensor methods for nonuniform hypergraphs. *SIAM Journal on Mathematics of Data Science*, 6(2):481–503, 2024. doi: 10.1137/23M1584472.

Alessia Antelmi, Gennaro Cordasco, Mirko Polato, Vittorio Scarano, Carmine Spagnuolo, and Dingqi Yang. A survey on hypergraph representation learning. *ACM Comput. Surv.*, 56(1), August 2023. ISSN 0360-0300. doi: 10.1145/3605776. URL https://doi.org/10.1145/3605776.

Anirban Banerjee, Arnab Char, and Bibhash Mondal. Spectra of general hypergraphs. *Linear Algebra and its Applications*, 518:14–30, 2017.

Federico Battiston, Enrico Amico, Alain Barrat, Ginestra Bianconi, Guilherme Ferraz de Arruda, Benedetta Franceschiello, Iacopo Iacopini, Sonia Kéfi, Vito Latora, Yamir Moreno, Micah M. Murray, Tiago P. Peixoto, Francesco Vaccarino, and Giovanni Petri. The physics of higher-order interactions in complex systems. *Nature Physics*, 17(10):1093–1098, 2021. doi: 10.1038/s41567-021-01371-4.

Austin R. Benson. Three hypergraph eigenvector centralities. *SIAM Journal on Mathematics of Data Science*, 1(2):293–312, 2019.

Austin R. Benson, Rediet Abebe, Michael T. Schaub, Ali Jadbabaie, and Jon Kleinberg. Simplicial closure and higher-order link prediction. *Proceedings of the National Academy of Sciences*, 115 (48):E11221–E11230, 2018. doi: 10.1073/pnas.1800683115.

Maciej Besta, Florian Scheidl, Lukas Gianinazzi, Grzegorz Kwasniewski, Shachar Klaiman, Jürgen Müller, and Torsten Hoefler. Demystifying higher-order graph neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 48(3):2544–2565, 2026. doi: 10.1109/TPAMI.2025.3637114.

Cristian Bodnar, Fabrizio Frasca, Nina Otter, Yuguang Wang, Pietro Liò, Guido F. Montúfar, and Michael Bronstein. Weisfeiler and Lehman go cellular: CW networks. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

Marianna Bolla. Spectra, Euclidean representations and clusterings of hypergraphs. *Discrete Mathematics*, 117(1–3):19–39, 1993.

Michael M. Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: Going beyond Euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, 2017. doi: 10.1109/MSP.2017.2693418.

T.-H. Hubert Chan, Anand Louis, Zhihao Gavin Tang, and Chenzi Zhang. Spectral properties of hypergraph Laplacian and approximation algorithms. *Journal of the ACM*, 65(3), 2018. doi: 10.1145/3178123.

Uthsav Chitra and Benjamin Raphael. Random walks on hypergraphs with edge-dependent vertex weights. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 1172–1181, 2019.

Fan Chung. Laplacians and the Cheeger inequality for directed graphs. *Annals of Combinatorics*, 9 (1):1–19, 2005. doi: 10.1007/s00026-005-0237-z.

Gonzalo Contreras-Aso, Cristian Pérez-Corral, and Miguel Romance. Uplifting edges in higher-order networks: Spectral centralities for non-uniform hypergraphs. *AIMS Mathematics*, 9(11): 32045–32075, 2024. doi: 10.3934/math.20241539.

Joshua Cooper and Aaron Dutle. Spectra of uniform hypergraphs. *Linear Algebra and its Applications*, 436(9):3268–3292, 2012. doi: 10.1016/j.laa.2011.11.018.

Mathieu Desbrun, Anil N Hirani, Melvin Leok, and Jerrold E Marsden. Discrete exterior calculus. *arXiv preprint math/0508341*, 2005.

Chad Giusti, Robert Ghrist, and Danielle S. Bassett. Two's company, three (or more) is a simplex: Algebraic-topological tools for understanding higher-order structure in neural data. *Journal of Computational Neuroscience*, 41(1):1–14, 2016. doi: 10.1007/s10827-016-0608-6.

Mustafa Hajij, Ghada Zamzmi, Theodore Papamarkou, Nina Miolane, Aldo Guzmán-Sáenz, Karthikeyan Natesan Ramamurthy, Tolga Birdal, Tamal K. Dey, Soham Mukherjee, Shreyas N. Samaga, Neal Livesay, Robin Walters, Paul Rosen, and Michael T. Schaub. Topological deep learning: Going beyond graph data. *arXiv preprint arXiv:2206.00606*, 2022.

Mustafa Hajij, Lennart Bastian, Sarah Osentoski, Hardik Kabaria, John Davenport, Sheik Dawood, Balaji Cherukuri, Joseph Kocheemoolayil, Nastaran Shahmansouri, Adrian Lew, Theodore Papamarkou, and Tolga Birdal. Copresheaf topological neural networks: A generalized deep learning framework. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-5009.

Jakob Hansen and Robert Ghrist. Toward a spectral theory of cellular sheaves. *Journal of Applied and Computational Topology*, 3(4):315–358, 2019.

Koby Hayashi, Sinan G. Aksoy, Cheong Hee Park, and Haesun Park. Hypergraph random walks, Laplacians, and clustering. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM)*, pp. 495–504, 2020. doi: 10.1145/3340531.3412034.

Danijela Horak and Jürgen Jost. Spectra of combinatorial Laplace operators on simplicial complexes. *Advances in Mathematics*, 244:303–336, 2013. doi: 10.1016/j.aim.2013.05.007.

Elvin Isufi, Geert Leus, Baltasar Beferull-Lozano, Sergio Barbarossa, and Paolo Di Lorenzo. Topological signal processing and learning: Recent advances and future challenges. *Signal Processing*, 233:109930, 2025. doi: 10.1016/j.sigpro.2025.109930.

Sooyeong Kim and Steve Kirkland. Gram mates, sign changes in singular values, and isomorphism. *Linear Algebra and its Applications*, 644:108–148, 2022.

Sunwoo Kim, Soo Yong Lee, Yue Gao, Alessia Antelmi, Mirko Polato, and Kijung Shin. A survey on hypergraph neural networks: An in-depth and step-by-step guide. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 6534–6544. ACM, 2024. doi: 10.1145/3637528.3671457.

Yann LeCun. A path towards autonomous machine intelligence. OpenReview, 2022. URL https://openreview.net/forum?id=BZ5a1r-kVsf.

Qiang Li, Zening Fu, Hasse Walum, Masoud Seraji, Prerana Bajracharya, Vince D. Calhoun, Sarah Shultz, and Armin Iraji. Deciphering multiway multiscale brain network connectivity from birth to 6 months. *Communications Biology*, 9(1):271, 2026. doi: 10.1038/s42003-026-09549-3.

Lek-Heng Lim. Hodge laplacians on graphs. *SIAM Review*, 62(3):685–715, 2020.

Sayan Mukherjee and John Steenbergen. Random walks on simplicial complexes and harmonics. *Random Structures & Algorithms*, 49(2):379–405, 2016. doi: 10.1002/rsa.20645.

Kevin A. Murgas, Emil Saucan, and Romeil Sandhu. Hypergraph geometry reflects higher-order dynamics in protein interaction networks. *Scientific Reports*, 12(1):20879, 2022. doi: 10.1038/s41598-022-24584-w.

Xavier Ouvrard, Jean-Marie Le Goff, and Stéphane Marchand-Maillet. Adjacency and tensor representation in general hypergraphs part 1: e-adjacency tensor uniformisation using homogeneous polynomials. *arxXiv preprint arXiv:1712.08189*, 2017.

Theodore Papamarkou, Tolga Birdal, Michael M. Bronstein, Gunnar E. Carlsson, Justin Curry, Yue Gao, Mustafa Hajij, Roland Kwitt, Pietro Liò, Paolo Di Lorenzo, Vasileios Maroulas, Nina Miolane, Farzana Nasrin, Karthikeyan Natesan Ramamurthy, Bastian Rieck, Simone Scardapane, Michael T. Schaub, Petar Veličković, Bei Wang, Yusu Wang, Guo-Wei Wei, and Ghada Zamzmi. Position: Topological deep learning is the new frontier for relational learning. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 39529–39555. PMLR, 2024.

Mathilde Papillon, Sophia Sanborn, Mustafa Hajij, and Nina Miolane. Architectures of topological deep learning: A survey of message-passing topological neural networks. *arXiv preprint arXiv:2304.10031*, 2023.

Tiago P. Peixoto, Leto Peel, Thilo Gross, and Manlio De Domenico. Graphs are maximally expressive for higher-order interactions. *arXiv preprint arXiv:2602.16937*, 2026.

Raphaël Pellegrin, Lukas Fesser, and Melanie Weber. Higher-order learning with graph neural networks via hypergraph encodings. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-5782.

Flavia Petruso, Maria Giulia Preti, and Dimitri Van De Ville. Topological signal processing: An application-oriented tutorial. *arXiv preprint arXiv:2605.22853*, 2026.

Stephen Ranshous, Cliff A. Joslyn, Sean Kreyling, Kathleen Nowak, Nagiza F. Samatova, Curtis L. West, and Samuel Winters. Exchange pattern mining in the Bitcoin transaction directed hypergraph. In *Financial Cryptography and Data Security (FC 2017)*, pp. 248–263, Cham, 2017. Springer International Publishing. doi: 10.1007/978-3-319-70278-0_16.

T. Mitchell Roddenberry, Michael T. Schaub, and Mustafa Hajij. Signal processing on cell complexes. In *ICASSP 2022 – IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 8852–8856, 2022. doi: 10.1109/ICASSP43922.2022.9747233.

Juan A. Rodríguez. On the Laplacian eigenvalues and metric parameters of hypergraphs. *Linear and Multilinear Algebra*, 50(1):1–14, 2002.

Stefania Sardellitti, Sergio Barbarossa, and Lucia Testa. Topological signal processing over cell complexes. In *2021 55th Asilomar Conference on Signals, Systems, and Computers*, pp. 1558–1562, 2021. doi: 10.1109/IEEECONF53345.2021.9723256.

Michael T. Schaub, Austin R. Benson, Paul Horn, Gabor Lippner, and Ali Jadbabaie. Random walks on simplicial complexes and the normalized Hodge 1-Laplacian. *SIAM Review*, 62(2):353–391, 2020. doi: 10.1137/18M1201019.

Michael T. Schaub, Yu Zhu, Jean-Baptiste Seby, T. Mitchell Roddenberry, and Santiago Segarra. Signal processing on higher-order networks: Livin' on the edge... and beyond. *Signal Processing*, 187:108149, 2021.

Zhe Su, Xiang Liu, Layal Bou Hamdan, Vasileios Maroulas, Jie Wu, Gunnar Carlsson, and Guo-Wei Wei. Topological data analysis and topological deep learning beyond persistent homology: a review. *Artificial Intelligence Review*, 59(2):58, 2026. doi: 10.1007/s10462-025-11462-w.

Lev Telyatnikov, Guillermo Bernárdez, Marco Montagna, Mustafa Hajij, Martin Carrasco, Pavlo Vasylenko, Mathilde Papillon, Ghada Zamzmi, Michael T. Schaub, Jonas Verhellen, Pavel Snopov, Bertran Miquel-Oliver, Manel Gil-Sorribes, Alexis Molina, Victor Guallar, Theodore Long, Julian Suk, Patryk Rygiel, Alexander Nikitin, Giordan Escalona, Michael Banf, Dominik Filipiak, Max Schattauer, Liliya Imasheva, Alvaro Martinez, Halley Fritze, Marissa Masden, Valentina Sánchez, Manuel Lecha, Andrea Cavallo, Claudio Battiloro, Matt Piekenbrock, Mauricio Tec, George Dasoulas, Nina Miolane, Simone Scardapane, and Theodore Papamarkou. TopoBench: A framework for benchmarking topological deep learning. *Journal of Data-centric Machine Learning Research*, 2(20):1–39, 2025. URL https://data.mlr.press/assets/pdf/v02-20.pdf.

Hao Tian and Reza Zafarani. Higher-order networks representation and learning: A survey. *ACM SIGKDD Explorations Newsletter*, 26(1):1–18, 2024. doi: 10.1145/3682112.3682114.

Leo Torres, Ann S. Blevins, Danielle Bassett, and Tina Eliassi-Rad. The why, how, and when of representations for complex systems. *SIAM Review*, 63(3):435–485, 2021. doi: 10.1137/20M1355896.

Koen van der Zwet, Ana I. Barros, Tom M. van Engers, and Peter M. A. Sloot. Opportunistic organization of illicit supply chains. *Journal of Quantitative Criminology*, 41(4):623–646, 2025. doi: 10.1007/s10940-025-09613-x.

Fuli Wang, Karelia Peña-Peña, Wei Qian, and Gonzalo R. Arce. T-HyperGNNs: Hypergraph neural networks via tensor representations. *IEEE Transactions on Neural Networks and Learning Systems*, 36(3):5044–5058, 2025. doi: 10.1109/TNNLS.2024.3371382.

Naganand Yadati, Madhav Nimishakavi, Prateek Yadav, Vikram Nitin, Anand Louis, and Partha Talukdar. HyperGCN: A new method for training graph convolutional networks on hypergraphs. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Mingshun Ye, Dezhi Han, Mingdong Tang, and Weili Chen. HyperSeq-LLM: Evolving hypergraph sequence learning with LLMs for phishing scam detection on Ethereum. *Expert Systems with Applications*, 308:131101, 2026. doi: 10.1016/j.eswa.2026.131101.

Dengyong Zhou, Jiayuan Huang, and Bernhard Schölkopf. Learning with hypergraphs: Clustering, classification, and embedding. In *Advances in Neural Information Processing Systems*, volume 19, pp. 1601–1608, 2006.

Yu Zhu, Boning Li, and Santiago Segarra. Co-clustering vertices and hyperedges via spectral hypergraph partitioning. In *2021 29th European Signal Processing Conference (EUSIPCO)*, pp. 1416–1420. IEEE, 2021. doi: 10.23919/EUSIPCO54536.2021.9616223.

A Algebraic Instantiations for Higher-Order Networks

As Section 2 argued, choosing an algebraic instantiation is an important (and separate) decision from choosing a higher-order abstraction. Here, we provide more detail in organizing the classes of algebraic instations available for hypergraphs, simplicial complexes, cell and combintorial complexes, and cochain complexes and cellular sheaves. Where appropriate, we also include a discussion of related random walks, as many are either implicitly or explicitly related to stochastic processes on higher-order networks.

Hypergraphs *Incidence matrices and graph expansions.* Nearly every matrix operator on a hypergraph $H = (V, E)$ starts from its incidence matrix $S \in \{0,1\}^{|V|\times|E|}$, with $S_{ve} = 1$ when $v \in e$. In particular, three graphs associated with the hypergraph can be constructed from it: (1) the weighted clique expansion is given by $SS^\top$, whose (u, v) entry counts the hyperedges containing both u and v; (2) the weighted line graph $S^\top S$ relates hyperedges by the size of their overlap; and (3) the so-called star expansion is the bipartite graph on $V \cup E$ with adjacency $\left(\begin{smallmatrix} 0 & S \\ S^\top & 0 \end{smallmatrix}\right)$. These graph structures, in turn, determine the nonzero pattern of classical hypergraph Laplacians. For example, Zhou et al. (2006), Bolla (1993), and Rodríguez (2002) each define a hypergraph Laplacian as

$$L_{\mathrm{Z}} = I - D_v^{-1/2} SW D_e^{-1} S^\top D_v^{-1/2}, \tag{3}$$

$$L_{\mathrm{B}} = D_v - SD_e^{-1} S^\top, \tag{4}$$

$$L_{\mathrm{R}} = D_{\mathrm{R}} - A_{\mathrm{R}}, \qquad A_{\mathrm{R}} = SS^\top - \mathrm{diag}(SS^\top), \tag{5}$$

respectively. Here W is the diagonal matrix of hyperedge weights $w(e)$, and D_v and D_e are the diagonal matrices of vertex degrees $d(v) = \sum_{e \ni v} w(e)$ and hyperedge degrees $\delta(e) = |e|$. Bolla's and Rodríguez's Laplacians are defined for unweighted hypergraphs ($W = I$); the latter is the graph Laplacian of the clique expansion whose edge $\{u, v\}$ has weight equal to the number of hyperedges containing both u and v, with D_{R} the diagonal matrix of row sums of A_{R} (Agarwal et al., 2006). However, Agarwal et al. (2006) showed that they are equivalent, up to shifts and scalings, to Laplacians of one of the aforementioned weighted expansion graphs. Section 2 discusses the consequences of this collapse for deep learning, together with Gram mates (Kim & Kirkland, 2022), whose incidence matrices agree in both $SS^\top$ and $S^\top S$. As we discuss next, there are several approaches to avoiding this collapse.

Nonlinear Laplacians. One approach for constructing hypergraph Laplacians that avoid collapsing to graph Laplacians is to use nonlinearity. For example, Chan et al. (2018) define a hypergraph Laplacian through a diffusion process in which, within each hyperedge, measure flows from the vertices of largest weighted measure to those of smallest. Analysis of these Laplacians is of course more difficult due to the loss of linear algebraic tools; for instance, Chan et al. recover a spectral property of a second eigenvalue by other means, and show that higher-order spectral properties cannot hold in general within their framework. HyperGCN (Yadati et al., 2019) is built on this operator: each hyperedge is represented by a single pairwise edge joining the two of its vertices whose current features differ most, or, in a mediator variant, by edges from these two vertices to the remaining vertices of the hyperedge, and the resulting graph is re-estimated at every epoch.

Adjacency tensors. Another approach is to use tensors, under the intuition that multi-linear algebra is most appropriate for defining multiway relations. For a k-uniform hypergraph, Cooper & Dutle (2012) define the adjacency hypermatrix as the order-k array of dimension $|V|$ with entry $1/(k-1)!$ at every index tuple whose underlying set is a hyperedge, define its eigenvalues both through a characteristic polynomial and variationally, and prove analogues of basic results of spectral graph theory. However, there is no canonical choice for how to define the tensor of a nonuniform hypergraph. The blowup construction of Banerjee et al. (2017) described in Section 2 is one such proposal, padding each hyperedge with repeated copies of its own vertices. Two other constructions pad with added auxiliary vertices instead: Ouvrard et al. (2017) build an e-adjacency tensor by uniformizing the hypergraph through homogeneous polynomials, so that hyperedge cardinalities are retained, and Contreras-Aso et al. (2024) introduce an *uplift* that adds an auxiliary node so that H- and Z-eigenvector centralities extend to nonuniform hypergraphs. Several features separating tensor spectra from their matrix counterparts are worth mentioning. The eigenproblem is a system of polynomial equations, so several notions of eigenvector coexist; for example, Benson (2019) compare centralities built from Z- and H-eigenvectors of the adjacency tensor with one built from the clique expansion. And these tensors

typically have exponentially-many entries, which is why algorithms that never form them (Aksoy et al., 2024) and compressed tensor layers (Wang et al., 2025) make these objects usable in practice.

Random walks. Random walks are intimately related to hypergraph operators. As discussed in (Hayashi et al., 2020), the standard walk is a two-step process: from the current vertex, choose an incident hyperedge with probability proportional to its weight $\omega(e)$, then choose a vertex of that hyperedge. If the second step is uniform, or more generally uses vertex weights that do not depend on the hyperedge, the walk is equivalent to a random walk on a weighted clique expansion, so Laplacians derived from it use only pairwise information (Chitra & Raphael, 2019). Accordingly, Chitra & Raphael (2019) utilize so-called edge-dependent vertex weights $\gamma_e(v)$, which arise naturally in data such as tf-idf in term-document data (Hayashi et al., 2020). Storing them in a weighted incidence matrix $R \in \mathbb{R}_{\geq 0}^{|V|\times|E|}$ with $R_{ve} = \gamma_e(v)$, and letting $W \in \mathbb{R}_{\geq 0}^{|V|\times|E|}$ have $W_{ve} = \omega(e)$ when $v \in e$ and 0 otherwise, the transition matrix is

$$P \;=\; D_V^{-1}\, W\, D_E^{-1} R^\top, \qquad D_V = \operatorname{diag}(W\mathbf{1}), \quad D_E = \operatorname{diag}(R^\top \mathbf{1}). \tag{6}$$

Such walks can be non-reversible, and hence inexpressible as a random walk on any undirected graph. Hayashi et al. (2020) explain how to use this walk to construct a suite of hypergraph Laplacians. For example, the asymmetric Laplacians $I - P$ and $\Phi^{1/2}(I - P)\Phi^{-1/2}$, with Φ the diagonal matrix of the stationary distribution, cannot be written as Laplacians of any graph when the walk is non-reversible. On the other hand, Chung's symmetrized directed Laplacian (Chung, 2005), applied to the same walk, turns out to be exactly the Laplacian of an edge-weighted clique expansion. Here, the collapse-to-graphs reappears at the symmetrization step, a phenomenon which is relevant to higher-order architectures that symmetrize its propagation operator.

Simplicial complexes *Boundary matrices and Hodge Laplacians.* The operators defined for simplicial complex (in contrast to the aforementioned ones for hypergraphs) rely on the downward closure property that makes the faces of every simplex available. More formally, fix a reference orientation for each simplex, write n_k for the number of k-simplices, and let $B_k \in \{0, \pm 1\}^{n_{k-1}\times n_k}$ represent the boundary map from k-simplices to $(k-1)$-simplices. The k-th combinatorial Hodge Laplacian (Lim, 2020; Schaub et al., 2020) is

$$L_k \;=\; B_k^\top B_k + B_{k+1} B_{k+1}^\top. \tag{7}$$

The two summands, the lower and upper Laplacians, couple k-simplices through shared faces and shared cofaces respectively, and the simplicial convolutional filters of Section 2.3 are polynomials in these two matrices. The Hodge Laplacian detects key topological invariants. For example, the dimension of $\ker L_k$ equals the k-th Betti number, and $\mathbb{R}^{n_k} = \operatorname{im} B_{k+1} \oplus \operatorname{im} B_k^\top \oplus \ker L_k$ orthogonally (Schaub et al., 2020). For edge signals these three components are the curl, gradient, and harmonic parts used throughout topological signal processing (Schaub et al., 2021). Returning to the simplicial complex in Figure 1, for example, $\ker L_1$ is one-dimensional, reflecting the single unfilled triangle on $\{a, b, d\}$.

Random walks. Schaub et al. (2020) define a random walk whose states are the oriented edges of a simplicial complex, so that each edge appears twice, once per orientation. At each step the walker moves to an edge sharing either a vertex or a triangle with its current edge, and the net flow f on each edge (the probability on one orientation minus that on the other) evolves as

$$f \;\mapsto\; -\tfrac{1}{2}\,\mathcal{L}_1 f, \tag{8}$$

where $\mathcal{L}_1$ is a degree-normalized variant of the Hodge 1-Laplacian L_1. The walk thus relates to L_1 as the random walk on a graph relates to the normalized graph Laplacian; for the precise transition rules and normalization, the reader is referred to Schaub et al. (2020). Mukherjee & Steenbergen (2016) take a related approach based on random walks with absorbing states.

Cell, combinatorial, and cochain complexes We also acknowledge an active literature on defining algebraic operators for other higher-order network structures, and briefly mention a few key results. For regular cell complexes, Roddenberry et al. (2022) define boundary matrices and Hodge Laplacians of the same form $L_k = B_k^\top B_k + B_{k+1} B_{k+1}^\top$, now allowing 2-cells to be polygons of any length, and Sardellitti et al. (2021) show that cell complexes inferred from edge signals represent those signals more sparsely than simplicial complexes inferred from the same data. On the learning side, CW Networks (Bodnar et al., 2021) lift graphs to cell complexes and are strictly more powerful than the

Weisfeiler–Leman test, while combinatorial complexes (Hajij et al., 2022), which include hypergraphs, simplicial complexes, and cell complexes as special cases, are handled through neighborhood matrices (incidence, adjacency, and coadjacency matrices). In the cochain view, the coboundary $B_{k+1}^{\top}$ acts as the discrete exterior derivative (Desbrun et al., 2005), and the Hodge Laplacian becomes $d_k^* d_k + d_{k-1} d_{k-1}^*$ for a chosen inner product on cochains, with the standard inner product recovering L_k and weighted ones giving the operators of Horak & Jost (2013). Cellular sheaves (Hansen & Ghrist, 2019) go one step further, replacing the ± 1 entries of the coboundary with linear restriction maps between vector spaces attached to cells. For more on Hodge Laplacians on simplicial and cell complexes, and their use in signal processing and learning, the reader is referred to Isufi et al. (2025).