applied sciences

MDPI

Article

Learning Directed Knowledge Using Higher-Ordered Neural Networks: Building a Predictive Framework

Yousra Moh Ousellam [1], Bikram Pratim Bhuyan [2,3,*], Rachida Fissoune [1], Galina Ivanova [4] and Amar Ramdane-Cherif [2]

[1] Systems and Data Engineering Team (IDS), National School of Applied Sciences, Abdelmalek Essaadi University, Tangier 93000, Morocco; yousra.mh.slm@gmail.com (Y.M.O.); rfissoune@uae.ac.ma (R.F.)
[2] LISV Laboratory, University of Paris-Saclay, 10–12 Avenue of Europe, 78140 Velizy, France
[3] École Centrale d'Électronique (ECE Engineering School), 10 Rue Sextius Michel, 75015 Paris, France
[4] Faculty of Electrical Engineering, Electronics and Automation, University of Ruse "Angel Kanchev", 8 Studentska Street, 7017 Ruse, Bulgaria
* Correspondence: bikram23bhuyan@gmail.com

Abstract

Most graph learning methods remain limited to undirected, pairwise interactions, restricting their ability to capture the multi-entity and directional relationships common in real-world systems. We propose the Directed Higher-Ordered Neural Network (HONN) framework that introduces directionality into hypergraph learning through flexible spectral Laplacian formulations. Unlike fixed-Laplacian methods such as the Generalized Directed Hypergraph Neural Network (GeDi-HNN), a tunable q-parameter in our framework balances local identity preservation with global diffusion, enabling robust and generalizable feature propagation. Experiments on five benchmark datasets show that HONN consistently matches or outperforms state-of-the-art baselines, achieving 84% on NTU-2012, 87.4% on WebKB Texas, and 86.2% on Cornell, while maintaining computational efficiency. Ablation studies confirm the crucial role of Laplacian selection, activation functions, and q-tuning in shaping model performance. By unifying directionality and higher-order reasoning, HONN provides a scalable foundation for predictive modeling in domains such as knowledge graphs, spatio-temporal networks, and recommendation systems.

Keywords: Directed Hypergraph Neural Networks; higher-order graph learning; spectral Laplacian operators; knowledge graphs; spatio-temporal networks

Academic Editors: João M. F. Rodrigues and Jose María Alvarez Rodríguez

Received: 5 September 2025
Revised: 9 October 2025
Accepted: 13 October 2025
Published: 16 October 2025

Citation: Moh Ousellam, Y.; Bhuyan, B.P.; Fissoune, R.; Ivanova, G.; Ramdane-Cherif, A. Learning Directed Knowledge Using Higher-Ordered Neural Networks: Building a Predictive Framework. *Appl. Sci.* **2025**, *15*, 11085. https://doi.org/10.3390/app152011085

1. Introduction

Knowledge representation has traditionally relied on graph-based structures, where entities can be modeled as vertices and pairwise interactions are captured through directed or undirected edges. Graph-based analysis plays a crucial role in understanding complex networks across domains such as bioinformatics, traffic systems, recommender systems, and social networks [1,2] (to name a few). While effective for simple relational learning [3,4], such representations fail to capture the complexity of higher-order interactions that naturally occur in domains such biological systems and social networks [5]. These interactions often involve multiple entities simultaneously and exhibit asymmetric dependencies, necessitating a shift from graph-based models to directed hypergraphs [6,7]. Directed hypergraphs provide a more expressive alternative, capturing higher-order relationships with explicit directionality [6,7].

Higher-Ordered Neural Networks (HONNs) operating on directed hypergraphs provide a principled framework to learn from such multi-way, asymmetric relationships [8]. Unlike standard Graph Neural Networks (GNNs) [3], HONNs can leverage incidence-based formulations, spectral convolutions, and Laplacian operators tailored for higher-order structures. The integration of directionality is particularly critical, as it enables the modeling of causal flows, temporal dependencies, and knowledge hierarchies that cannot be expressed in undirected or pairwise settings.

Recent advances in hypergraph learning have introduced several spectral Laplacians; such as the normalized Laplacian, Hermitian Laplacian, and their complex-valued variants; that generalize classical graph theory to directed higher-order domains. However, existing spectral approaches are based on a single Laplacian formulation [9], yet its impact on directed HONNs performance remains under-explored. Most studies either restrict themselves to undirected settings or fail to provide a systematic framework that unifies directionality, higher-order representations, and predictive modeling. As a result, there is a pressing need to establish a coherent mathematical and algorithmic foundation for learning directed knowledge using higher-ordered neural networks.

This work lies in unifying directionality, higher-order representations, and predictive learning into a single principled framework. Unlike prior approaches that either focus exclusively on undirected hypergraphs or adopt ad hoc definitions of directionality, we provide the first systematic study of spectral Laplacians tailored to directed hypergraphs and their integration into Higher-Ordered Neural Networks (HONNs). Our specific contributions are as follows:

1. We extend classical Laplacian constructions by formulating directed variants such as the Hermitian Laplacian with tunable parameter q, and its complex-valued extension. These operators explicitly balance local versus global structural information and capture phase-dependent directionality, offering unprecedented flexibility for predictive learning.
2. We develop a Higher-Ordered Neural Network (HONN) architecture grounded in spectral convolution, designed to exploit these new Laplacians. To the best of our knowledge, this is the first framework that systematically links Laplacian choice and parameterization with predictive performance in directed knowledge settings.
3. Through extensive experiments on benchmark datasets, we reveal how Laplacian formulations and q-values govern stability, convergence, and generalization. These findings provide practical guidelines for deploying HONNs in real-world scenarios where higher-order, directed dependencies are critical.

Together, these contributions establish a novel direction in neuro-symbolic [10,11] and hypergraph learning. The remainder of this paper is organized as follows. Section 2 reviews related work on graph and hypergraph neural networks, highlighting the limitations of existing approaches in handling directionality and higher-order relationships. Section 3 introduces the theoretical foundations of directed hypergraphs and defines the spectral Laplacian operators used in HONN. Section 4 presents the proposed methodology, including dataset descriptions, hypergraph construction, and model design. Section 5 reports experimental results and ablation studies, followed by a detailed discussion in Section 6 on general performance, stability, and comparisons with baselines. Section 7 discusses the results with key insights and directions for future research. Finally, Section 8 concludes this paper.

2. State of the Art and Research Gaps

Graph Neural Networks (GNNs) and Hypergraph Neural Networks (HGNNs) have become central to learning complex relationships in graph-structured data. GNNs, pio-

neered by Scarselli et al. [3], leveraged graph topology for node embeddings, enabling nodes to aggregate information from neighboring nodes. This innovation laid the foundation for modern GNNs. Over time, methods such as MMagNet [12] and SigMaNet [13] extended GNNs to incorporate magnetic and sign-magnetic Laplacians, improving edge directionality and weight representation. These advancements have been impactful in tasks like spectral partitioning and link prediction. However, GNNs face challenges in scaling to dynamic, large-scale graphs and struggle to capture higher-order interactions effectively. It is important to note that SigMaNet was designed to handle directed and signed relationships in traditional Graph Neural Networks (GNNs), which are limited to pairwise connections.

Building on GNN advancements, HGNNs model hypergraphs, where hyperedges connect multiple nodes simultaneously, facilitating the modeling of higher-order relationships. This approach is valuable in domains like multimodal data integration and citation classification. Models by Bai et al. [14] and Feng et al. [15] used convolutional and attention mechanisms to improve hypergraph representation learning. Yet, HGNNs still struggle with scalability, dynamic environments, and the integration of directional dependencies within hypergraphs.

While HGNNs have advanced the field of higher-order learning, recent models like Hypergraph Convolutional Networks (HGCNs) and Hypergraph Convolutional Networks with Hyperedge Attention (HCHAs) [14] still have limitations. HGCN [15] is a strong baseline that effectively extends GCNs to hypergraphs, but it operates on undirected hypergraph structures, thus failing to capture the directional information critical in many real-world systems. Similarly, HCHA introduces an attention mechanism [16] to weigh the importance of hyperedges, but it also does not inherently model the direction of information flow. Both HGCN and HCHA show the importance of spectral and attention-based methods for hypergraphs, but they leave a significant gap in systematically handling directed higher-order relationships, which our framework is specifically designed to address.

Directed Hypergraph Neural Networks (DHGNNs) [6] have addressed some of these challenges by integrating directionality into hypergraphs, improving the modeling of source–target relationships in domains like traffic systems and citation analysis. Tran et al. [17] used normalized Laplacians to model dynamic interactions, and Pretolani [6] explored directed hypergraphs for routing and connectivity. Despite this, existing DHGNNs often lack scalability, efficiency, and generalization for large, diverse datasets.

Recent work on the Generalized Directed Hypergraph Neural Network (GeDi-HNN) [18] has made strides in overcoming the limitations of both traditional GNNs and HGNNs. GeDi-HNN introduces a novel complex-valued Hermitian matrix (the Generalized Directed Laplacian), which allows for seamless integration of directed and undirected hyperedges while generalizing existing Laplacian formulations. This development is crucial for effectively modeling asymmetric interactions and higher-order relationships. However, a significant limitation is that the model relies on a single, fixed Laplacian formulations. The broader landscape for DHGNNs still presents gaps, particularly regarding the systematic evaluation of different Laplacian formulations and their impact on model performance [19,20]. This indicates a need for a more flexible framework that can explore and adapt to various spectral properties beyond a singular, fixed approach. The limitations of traditional GNNs and HGNNs, particularly their difficulty in capturing directionality and complex interactions, create a gap for more expressive models. Domains such as recommendation systems, traffic networks, and academic citation analysis demand models that can represent both higher-order interactions and directional dependencies. Current methods are often inadequate, leading to suboptimal predictions and limited interpretability.

While previous works have introduced Directed Hypergraph Neural Networks (DHGNNs), they have relied on fixed Laplacian formulations without systematically evaluating their impact. The effect of different Laplacian matrices and q-values on DHGNN performance remains unexplored.

This study fills this gap by systematically comparing multiple Laplacian formulations and varying q to control the balance between local node properties and global hypergraph structure. Our goal is to provide a deeper understanding of how spectral properties influence learning stability, accuracy, and generalization in directed hypergraphs.

Recent advances in graph learning have emphasized spectral adaptability, uncertainty estimation, and structural robustness. Lin et al. [21] proposed Graph Neural Stochastic Diffusion (GNSD), which integrates stochastic diffusion into GNNs to enhance both predictive accuracy and uncertainty estimation, particularly in noisy or out-of-distribution settings. In the hypergraph domain, Yang and Xu [22] provided a comprehensive survey of recent HGNN models, identifying critical challenges related to expressiveness, directionality, and spectral design—challenges directly addressed by our HONN framework. Ko et al. [23] introduced a Laplacian perturbation approach for universal graph contrastive learning, demonstrating the benefits of carefully designed spectral modifications. These works reinforce the importance of flexible Laplacian modeling and motivate our investigation of multiple Laplacian variants and adaptive spectral tuning through the q-parameter.

Table 1 provides a summary of different graph-based neural network models, highlighting their key characteristics and spectral foundation. GNNs form the foundation, relying on the Standard Graph Laplacian for simple, undirected pairwise relationships, ensuring stability but lacking the ability to model flow or group dynamics. HGNNs extend this by using the Undirected Hypergraph Laplacians to capture higher-order relationships. Concurrently, DGNNs and DHGNNs incorporate directionality into their analysis, utilizing Directed Laplacians to model flow asymmetry. DHGNNs, which are the focus of this research, offer the most comprehensive representation by combining higher-order interactions and directionality, typically via a Generalized Directed Laplacian. However, as summarized in the table, previous DHGNNs suffer from a lack of spectral flexibility due to reliance on a single, fixed Laplacian formulation. Our framework addresses this critical gap by systematically comparing Multiple Laplacian Variants and introducing the adaptive q-parameter.

However, several key challenges remain in the field of HGNNs and directed hypergraphs:

- **Scalability and Efficiency**: Handling large hypergraphs remains a critical issue [15,24]. Developing methods that maintain efficiency and computational speed for massive datasets is essential [12].
- **Handling Edge Weights and Directions**: While progress has been made with methods like the sign-magnetic Laplacian [13], fully integrating edge weights and directionality into HGNNs remains a significant challenge [6,17].
- **Combining Different Data Types**: Incorporating heterogeneous data types into a unified hypergraph framework is difficult [24,25]. Future work should explore more effective ways to merge diverse data types (e.g., categorical, numerical, temporal) into hypergraph representations [9].
- **Generalization Across Different Uses**: Many HGNN models are domain-specific, limiting their applicability [3]. There is a need for models that can generalize to various domains, such as social networks, recommendation systems, and biology [1].
- **Improving Learning Techniques**: Enhancing learning techniques, especially for sparse data, is critical for the practical application of HGNNs [9,12]. Improving the efficiency and accuracy of these methods will increase their effectiveness in real-world scenarios [15].

Addressing these research gaps is crucial to enhancing the scalability, versatility, and real-world applicability of hypergraph neural networks.

Table 1. Summary of state-of-the-art techniques and spectral analysis. For the computational complexity, $n = |\mathcal{V}|$ (vertices), $e = |E|$ (graph edges), $m = |\mathcal{E}|$ (hyperedges), $r = \mathrm{nnz}(H) = \sum_{e\in\mathcal{E}} |e|$ (non-zeros in the incidence matrix H), d is the feature width/hidden size per layer, B is the number of parallel branches, and K is the number of Laplacian variants evaluated. Costs are per layer; diagonal scalings (e.g., $D_v^{-1/2}$, D_e^{-1}, W) are linear-time.

Technique	Higher-Order Interactions	Spectral Flexibility/Adaptive q	Systematic Laplacian Eval.	Compute Complexity	Laplacian Matrices Used
Classic GNNs	No (Pairwise)	No	No	$\mathcal{O}(e\,d + n\,d^2)$ per layer	Graph Laplacian (Standard)
DGNNs	No (Pairwise)	No	No	$\mathcal{O}(e\,d + n\,d^2)$ per layer [1]	Directed Graph Laplacians (e.g., Magnetic)
HGNNs (HGCN, HCHA)	Yes	No	No	$\mathcal{O}\big((r+m)\,d + n\,d^2\big)$ per layer	Hypergraph Laplacian (Undirected)
Previous DHGNNs (e.g., GeDi-HNN)	Yes	Low/Limited (Fixed)	No	$\mathcal{O}\big((r+m)\,d + n\,d^2\big)$ per layer	Generalized Directed Laplacian (Single, Fixed Formulation)
HONN (Our Model)	**Yes**	**Yes** (via q-parameter)	**Yes**	$\mathcal{O}\big(BK\,(r+m)\,d + n\,d^2\big)$ per layer	**Multiple Laplacian Variants** (Systematically Compared)

[1] Complex/magnetic DGNNs often incur an empirical constant factor (e.g., $\sim 2\times$) without changing the asymptotic order.

3. Directionality in Higher-Ordered Neural Network

In the context of higher-order neural networks (HONNs), effectively capturing directionality is crucial for modeling complex, asymmetric relationships inherent in various data structures. This section delves into the mathematical foundations of directionality within HONNs represented as directed hypergraphs, emphasizing the role of spectral convolution and the associated Laplacian matrices.

3.1. Higher-Ordered Representation

Higher-ordered representations generalize traditional pairwise relationships in graphs to multi-way interactions. This formalism enables modeling complex systems where interactions involve more than two entities, such as knowledge graphs, social networks, and spatio-temporal data. Below, we provide mathematical definitions, properties, and proofs to establish the foundation for higher-ordered neural networks.

Definition 1. *A hypergraph [26,27] (generalization of graph) is defined as $G = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$ is a finite set of n vertices. $\mathcal{E} = \{e_1, e_2, \ldots, e_m\}$ is a set of m hyperedges, where each hyperedge $e_i \subseteq \mathcal{V}$ connects a subset of vertices.*

If $|e_i| = 2$ for all i, G reduces to a traditional graph. The cardinality of a hyperedge e_i defines its order, allowing the representation of relationships involving multiple entities simultaneously.

Definition 2. *A directed hypergraph [28,29] is an extension of a hypergraph where each hyperedge $e \in \mathcal{E}$ is represented as an ordered pair $e = (T(e), H(e))$, where $T(e) \subseteq \mathcal{V}$ is the tail set, representing the source nodes. $H(e) \subseteq \mathcal{V}$ is the head set, representing the target nodes. $T(e) \cap H(e) = \emptyset$ to maintain disjoint directionality.*

Directed hypergraphs generalize directed graphs by allowing hyperedges to connect multiple vertices simultaneously, enabling representation of complex n-arry relationships.

An example is provided in Figure 1. The left panel illustrates a traditional graph representation, where pairwise directed edges encode relationships such as “father_of" and “mother_of" between vertices A, B, C, D, and E. For example, the directed edge $A \rightarrow C$ denotes “father_of(C)," while $B \rightarrow C$ denotes “mother_of(C)." The “marry" relationship is modeled as a bidirectional edge between A and B.

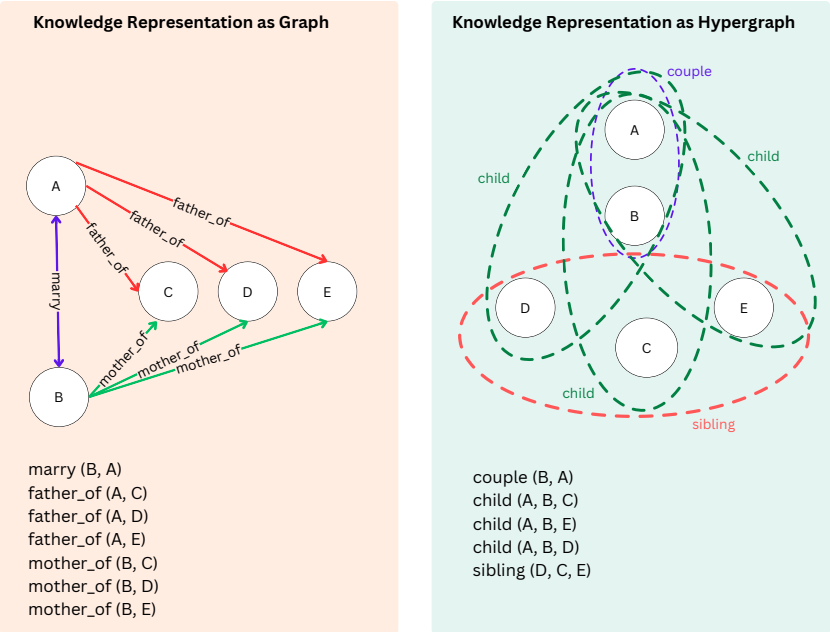

Figure 1. Comparison of knowledge representation in graphs and hypergraphs with directionality.

The right panel shows a directed hypergraph representation, where hyperedges capture multi-way, higher-order relationships. Hyperedges are represented as dashed lines encapsulating vertices. The “child” hyperedge connects A, B and C, D, E, representing the relationship between parents A and B and their children C, D, and E. In this directed hypergraph, the “sibling” relationship between C, D, and E is the target to be predicted by the Higher-Ordered Neural Network (HONN), using information encoded in other hyperedges such as “child” and “couple”. Directionality is implied through the hypergraph structure: for example, the “child” relationship flows from $\{A, B\}$ (parents) to $\{C, D, E\}$ (children). This demonstrates how directed hypergraphs and HONNs enable learning of higher-order relationships, such as predicting “sibling” from complex, interconnected n-arry data.

Definition 3. *The incidence matrix $H \in \mathbb{R}^{n \times m}$ of a directed hypergraph G is defined as follows:*

$$H_{ij} = \begin{cases} -1, & \text{if } v_i \in T(e_j), \\ +1, & \text{if } v_i \in H(e_j), \\ 0, & \text{otherwise.} \end{cases}$$

This matrix encodes vertex–hyperedge relationships algebraically.

Property 1. *The degree of a vertex $v_i \in \mathcal{V}$ is the sum of weights of all hyperedges connected to it. Mathematically;*

$$d(v_i) = \sum_{j=1}^{m} |H_{ij}| W_{jj}.$$

Proof. From the incidence matrix H, the entry $H_{ij} \neq 0$ if and only if the vertex v_i is connected to the hyperedge e_j. The weight of the hyperedge e_j is given by W_{jj}, and the contribution of e_j to v_i is scaled by $|H_{ij}|$, which equals 1 (for connection) or 0 (otherwise). Thus, summing over all hyperedges e_j gives:

$$d(v_i) = \sum_{j=1}^{m} |H_{ij}| W_{jj}.$$

□

Property 2. *The total contribution of a hyperedge e_j to its vertices can be represented as follows:*

$$\textit{Weight Contribution}(e_j) = \sum_{i=1}^{n} |H_{ij}| = |T(e_j)| + |H(e_j)|.$$

Proof. By the definition of the incidence matrix H, the absolute value $|H_{ij}|$ contributes 1 if vertex v_i is part of the tail or head of hyperedge e_j. Summing over all vertices v_i gives:

$$\sum_{i=1}^{n} |H_{ij}| = |T(e_j)| + |H(e_j)|.$$

This shows that the contribution of e_j is the combined size of its tail and head sets. □

Property 3. *For any hypergraph G with incidence matrix H, weight matrix W, and degree matrices D_v and D_e, the following identity holds:*

$$D_v = HWH^\top.$$

Proof. The vertex degree matrix D_v is diagonal, where each entry $(D_v)_{ii}$ represents the sum of the weights of all hyperedges connected to the vertex v_i. By definition:

$$(D_v)_{ii} = \sum_{j=1}^{m} |H_{ij}|^2 W_{jj}.$$

Since $|H_{ij}|^2 = 1$ when $H_{ij} \neq 0$, this simplifies to:

$$(D_v)_{ii} = \sum_{j=1}^{m} H_{ij} W_{jj} H_{ij}.$$

Expressing this operation in matrix form:

$$D_v = HWH^\top,$$

where $HWH^\top$ computes the weighted sum of hyperedges incident to each vertex. □

The above definitions and properties provide a foundation for higher-ordered representations. Directed hypergraphs enable modeling of multi-way, asymmetric interactions, and their algebraic representation facilitates spectral analysis. These properties will underpin the development of spectral convolution and Laplacian-based techniques discussed in subsequent sections.

3.2. Spectral Convolution on Higher-Ordered Neural Network

Spectral convolution provides a powerful framework for capturing global and higher-order dependencies in neural networks operating on hypergraphs [27,30,31]. By leveraging the eigenstructure of hypergraph Laplacian matrices, spectral convolution enables efficient filtering and learning over complex relationships. To perform spectral convolution, signals on hypergraphs are transformed into the spectral domain using the Fourier basis derived from the hypergraph Laplacian [32]. The spectral hypergraph Laplacian for directionality is discussed in the next subsection.

Definition 4. *Given a signal $x \in \mathbb{R}^n$ defined on the vertices of a hypergraph, the Fourier transform of x is represented as follows:*

$$\hat{x} = \Phi^\top x,$$

where Φ is the matrix of eigenvectors of the Laplacian matrix Θ, and $\hat{x}$ represents the signal in the spectral domain. The inverse Fourier transform is given by:

$$x = \Phi \hat{x}.$$

The eigenvalues $\Lambda = diag(\lambda_1, \lambda_2, \ldots, \lambda_n)$ of the Laplacian matrix Θ represent the frequency components of the hypergraph. Low eigenvalues correspond to smooth variations, while higher eigenvalues capture rapid changes.

Definition 5. *The spectral convolution of a signal x with a filter g_Θ in the spectral domain is defined as follows:*

$$g_\Theta * x = \Phi g(\Lambda) \Phi^\top x,$$

where $g(\Lambda)$ is a diagonal matrix representing the spectral filter applied to the eigenvalues λ_i of Θ. The filter $g(\Lambda)$ is parameterized to learn how signals should be smoothed or emphasized at different spectral frequencies. This enables the model to adapt to the structure of the hypergraph.

Direct computation of spectral convolution involves eigenvalue decomposition, which is computationally expensive for large hypergraphs. To address this, the filter $g(\Lambda)$ is approximated using Chebyshev polynomials [33].

Definition 6. *The spectral filter $g(\Lambda)$ is approximated as follows:*

$$g(\Lambda) \approx \sum_{k=0}^{K} \theta_k T_k(\Lambda),$$

where $T_k(\Lambda)$ is the k-th Chebyshev polynomial defined recursively as follows:

$$T_0(\lambda) = 1, \quad T_1(\lambda) = \lambda, \quad T_{k+1}(\lambda) = 2\lambda T_k(\lambda) - T_{k-1}(\lambda),$$

and θ_k are the learnable parameters of the filter. This approximation avoids explicit eigen decomposition and allows efficient computation of spectral convolution, making it suitable for large-scale hypergraphs.

In the context of higher-ordered neural networks, spectral convolution operates on directed hypergraphs, leveraging the Laplacian Θ that encodes directionality and higher-order interactions. For a given input signal $X \in \mathbb{R}^{n \times d}$, where d is the feature dimension, the output of a spectral convolution layer can be formulated as [34]:

$$X' = \Phi g(\Lambda) \Phi^\top X.$$

Here, the spectral filter $g(\Lambda)$ adapts to the higher-order structure of the hypergraph, enabling the model to learn from both global and local patterns.

Spectral convolution provides a mathematically principled approach to learning on higher-order structures. Its ability to encode global relationships and adapt to directionality makes it a cornerstone of higher-ordered neural networks. The scalability offered by Chebyshev approximation ensures its applicability to large-scale datasets, while its spectral filtering capabilities enhance learning from complex, structured data.

3.3. Encoding Directionality in Directed Hypergraphs

Let $G = (\mathcal{V}, \mathcal{E})$ be a directed hypergraph as defined in the previous section, where each hyperedge $e \in \mathcal{E}$ is an ordered pair $e = (T(e) \to H(e))$ with $T(e), H(e) \subseteq \mathcal{V}$, $T(e) \cap H(e) = \emptyset$, and weight $w_e > 0$. We set $n = |\mathcal{V}|$ and $m = |\mathcal{E}|$. Vectors are column vectors; for a matrix X, X^* denotes the conjugate transpose. The definitions below apply to real or complex-valued features. All operators introduced are Hermitian.

We define the 'weight-free' directed incidence $B \in \mathbb{C}^{n\times m}$ by:

$$B_{v,e} = \begin{cases} \dfrac{1}{\sqrt{|T(e)|}}, & v \in T(e), \\ -\,\mathrm{i}\,\dfrac{1}{\sqrt{|H(e)|}}, & v \in H(e), \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

Let $W = \mathrm{diag}(w_e)$ and define the Hermitian (Gram) moment

$$M := B\,W\,B^* \in \mathbb{C}^{n\times n}. \tag{2}$$

Since M is a weighted sum of rank-one Hermitian matrices, $M^* = M$ and $M \succeq 0$.

We use a real, non-negative degree matrix given by the diagonal of M:

$$D := \mathrm{diag}(M) = \mathrm{diag}\Big(\sum_{e\in\mathcal{E}} w_e\,|B_{\cdot,e}|^2\Big) \in \mathbb{R}^{n\times n}. \tag{3}$$

We can now define the normalized adjacency as follows:

$$A_H := D^{-1/2} M D^{-1/2}, \qquad \widetilde{L}_H := I - A_H. \tag{4}$$

Both A_H and $\widetilde{L}_H$ are Hermitian. (We refrain from asserting $D - M \succeq 0$ without extra assumptions; our propagation uses A_H directly.)

Directionality is injected via the complex phase attached to head incidences in (1). Consequently, each column $B_{\cdot,e}$ has two distinct magnitudes (scaled by $|T(e)|^{-1/2}$ and $|H(e)|^{-1/2}$) and two distinct phases (1 and $-\mathrm{i}$). Swapping $T(e)$ and $H(e)$ does not multiply $B_{\cdot,e}$ by a global unit-modulus scalar; it changes magnitudes and phases entrywise, so $M = BWB^*$ (hence its spectrum) generally changes. Thus, the construction captures genuine directionality beyond undirected orientations.

For permutation behavior, let $P \in \{0,1\}^{n\times n}$ be any vertex permutation and $Q \in \{0,1\}^{m\times m}$ any hyperedge permutation. With $B' := PBQ$, $M' := B'WB'^*$, and $D' := \mathrm{diag}(M')$, we have:

1. *Vertex equivariance:* $M' = PMP^\top$, $D' = PDP^\top$, hence $P\,\widetilde{L}_H\,P^\top = \widetilde{L}'_H = I - (D')^{-1/2} M' (D')^{-1/2}$.
2. *Within-set invariance:* Any permutation of nodes inside $T(e)$ (resp. inside $H(e)$) leaves $B_{\cdot,e}$ unchanged up to permutation of equal entries; hence M and $\widetilde{L}_H$ are invariant under such within-set permutations.

3. *Tail–head swap sensitivity:* Replacing $e = (T(e) \to H(e))$ by $e^{\text{rev}} = (H(e) \to T(e))$ generally yields a different M (and spectrum) unless in special symmetric cases.

Continuing with Figure 1, let $\mathcal{V} = \{A, B, C, D\}$ and consider two directed hyperedges that encode the ternary relation $\mathsf{child}(\cdot, \cdot, \cdot)$:

$$e_1 = (\{A, B\} \to \{C\}), \qquad e_2 = (\{A, B\} \to \{D\}),$$

each with weight $w_{e_1} = w_{e_2} = 1$. Using (1), the directed incidence $B \in \mathbb{C}^{4\times 2}$ (columns are e_1, e_2; rows are A, B, C, D) is

$$B = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ -\mathrm{i} & 0 \\ 0 & -\mathrm{i} \end{bmatrix}.$$

With $W = \mathrm{diag}(1, 1)$, the Hermitian moment is $M = BWB^* = BB^*$:

$$M = \begin{bmatrix} 1 & 1 & \mathrm{i}/\sqrt{2} & \mathrm{i}/\sqrt{2} \\ 1 & 1 & \mathrm{i}/\sqrt{2} & \mathrm{i}/\sqrt{2} \\ -\mathrm{i}/\sqrt{2} & -\mathrm{i}/\sqrt{2} & 1 & 0 \\ -\mathrm{i}/\sqrt{2} & -\mathrm{i}/\sqrt{2} & 0 & 1 \end{bmatrix}.$$

This matrix is Hermitian ($M^* = M$) and positive semidefinite (since $M = BB^*$).

Reversing only e_1 to $e_1^{\text{rev}} = (\{C\} \to \{A, B\})$ replaces the first column of B by $[-\mathrm{i}/\sqrt{2}, -\mathrm{i}/\sqrt{2}, 1, 0]^\top$, yielding a different $M' = B'B'^*$ and thus a different normalized operator $A'_H = D^{-1/2}M'D^{-1/2}$. This demonstrates that the layer response depends on hyperedge direction beyond mere orientation.

3.4. Spectral Laplacians for Directionality

The Laplacian matrix is a fundamental construct in spectral graph theory, extending naturally to hypergraphs for encoding relationships among vertices and hyperedges. For directed hypergraphs, spectral Laplacians incorporate both directionality and higher-order connectivity, enabling spectral convolution to capture asymmetric dependencies effectively.

Various forms of spectral Laplacians have been developed to encode directionality and higher-order relationships. Each form has unique mathematical properties, tailored to specific computational and application needs. This section provides an overview of these Laplacians, their properties, and their relevance to directionality.

3.4.1. Normalized Laplacian

The normalized Laplacian (inspired from [35]) is a widely used variant of the spectral Laplacian [18], designed to stabilize computations by balancing contributions from vertices and hyperedges. This form is particularly effective in heterogeneous hypergraphs, where node and edge degrees vary significantly.

Definition 7. *For a directed hypergraph $G = (\mathcal{V}, \mathcal{E})$, the normalized Laplacian can be defined as;*

$$\Theta_N = I - D_v^{-1/2} H W D_e^{-1} H^\top D_v^{-1/2} \tag{5}$$

where D_v is the vertex degree matrix, where $(D_v)_{ii} = \sum_{j=1}^{m} |H_{ij}| W_{jj}$. D_e is the hyperedge degree matrix, where $(D_e)_{jj} = \sum_{i=1}^{n} |H_{ij}|$. H is the incidence matrix, encoding the connectivity between vertices and hyperedges. W is the diagonal hyperedge weight matrix, where W_{jj} represents the weight of hyperedge e_j. Some properties of the normalized Laplacian are [35]:

Property 4. *The normalized Laplacian Θ_N is symmetric:*

$$\Theta_N = \Theta_N^\top.$$

This property ensures that its eigenvalues are real, a critical feature for spectral analysis.

Proof. The term $D_v^{-1/2} H W D_e^{-1} H^\top D_v^{-1/2}$ involves (i) $D_v^{-1/2}$ is diagonal and symmetric. (ii) W and D_e^{-1} are diagonal and symmetric. (iii) $H W D_e^{-1} H^\top$ is symmetric because H is real-valued and $W D_e^{-1}$ is diagonal. Thus, the entire term $D_v^{-1/2} H W D_e^{-1} H^\top D_v^{-1/2}$ is symmetric. Since $\Theta_N = I - D_v^{-1/2} H W D_e^{-1} H^\top D_v^{-1/2}$, it follows that Θ_N is symmetric:

$$\Theta_N = \Theta_N^\top.$$

□

Property 5. *The normalized Laplacian Θ_N is positive semi-definite:*

$$x^\top \Theta_N x \geq 0, \quad \forall x \in \mathbb{R}^n.$$

Proof. The normalized Laplacian can be written as follows:

$$\Theta_N = I - Q_N, \quad Q_N = D_v^{-1/2} H W D_e^{-1} H^\top D_v^{-1/2}.$$

For any $x \in \mathbb{R}^n$:

$$x^\top \Theta_N x = x^\top I x - x^\top Q_N x.$$

The term $x^\top I x = \|x\|^2 \geq 0$, and Q_N is symmetric and positive semi-definite:

$$x^\top Q_N x \geq 0.$$

Thus :

$$x^\top \Theta_N x = \|x\|^2 - x^\top Q_N x \geq 0.$$

□

Property 6. *The eigenvalues of Θ_N lie in the range $[0, 1]$:*

$$0 \leq \lambda_{\min} \leq \lambda \leq \lambda_{\max} \leq 1.$$

Proof. Let $\Theta_N = I - Q_N$, where $Q_N = D_v^{-1/2} H W D_e^{-1} H^\top D_v^{-1/2}$. Since Q_N is a normalized adjacency-like matrix, its eigenvalues are bounded as follows:

$$0 \leq \lambda(Q_N) \leq 1.$$

For Θ_N, the eigenvalues satisfy:

$$\lambda(\Theta_N) = 1 - \lambda(Q_N).$$

Since $\lambda(Q_N) \in [0, 1]$, it follows that:

$$0 \leq \lambda(\Theta_N) \leq 1.$$

□

Property 7. *The normalized Laplacian Θ_N is robust to variations in vertex and hyperedge degrees.*

Proof. The normalization factors $D_v^{-1/2}$ and D_e^{-1} rescale the contributions of vertices and hyperedges in the incidence matrix H. This ensures that the impact of a vertex v_i or hyperedge e_j is independent of their absolute degrees. Specifically:

$$D_v^{-1/2} \quad \text{scales the rows of } H \quad \text{and} \quad D_e^{-1} \quad \text{scales the hyperedge weights in } W.$$

As a result, Θ_N remains invariant to changes in the scale of D_v and D_e, ensuring robustness in heterogeneous datasets. □

3.4.2. Hermitian Laplacian

We define the Hermitian Laplacian as a variant of the spectral Laplacian designed to balance structural features and directionality. By incorporating a factor of probabilistic tuning parameter 'q', it provides a flexible framework for learning on directed hypergraphs.

Definition 8. *The Hermitian Laplacian Θ_H can be defined as follows:*

$$\Theta_H = (1-q)I + qD_v^{-1/2}HWD_e^{-1}H^\top D_v^{-1/2}, \quad q \in [0,1] \tag{6}$$

where q is a tunable parameter balancing the identity matrix and normalized Laplacian. H is the incidence matrix. D_v is the diagonal vertex degree matrix. D_e is the diagonal hyperedge degree matrix. W is the diagonal hyperedge weight matrix.

The Hermitian Laplacian combines the identity matrix I with the normalized term $D_v^{-1/2}HWD_e^{-1}H^\top D_v^{-1/2}$, allowing flexibility in capturing higher-order relationships. This matrix offers the same properties as that of the normalized Laplacian and can be proved in a similar manner.

Property 8. *The Hermitian Laplacian Θ_H is symmetric:*

$$\Theta_H = \Theta_H^\top.$$

Property 9. *The Hermitian Laplacian satisfies:*

$$x^\top \Theta_H x \geq 0, \quad \forall x \in \mathbb{R}^n.$$

Property 10. *The eigenvalues of Θ_H lie in the range:*

$$(1-q) \leq \lambda \leq 1.$$

The Hermitian Laplacian provides a basis for spectral convolution, where its eigenvalues enable filtering of directional signals. Balancing identity and structure makes Θ_H suitable for tasks requiring a mix of local and global information. The tunable parameter q allows focusing on structural relationships, crucial for predicting links in directed hypergraphs. Following the methods used in [4,13,36], we can show that the Hermitian Laplacian is positive semi-definite and thus can be adopted as a convolutional operator.

3.4.3. Hermitian Laplacian Matrix with Imaginary Part

The Hermitian Laplacian with an imaginary part extends the Hermitian Laplacian to include directional and phase information, making it suitable for tasks involving oscillatory or cyclic dynamics in directed hypergraphs.

Definition 9. *For a directed hypergraph $G = (\mathcal{V}, \mathcal{E})$, the Hermitian Laplacian with an imaginary part is defined as follows:*

$$\Theta_H^c = \Theta_H + i \cdot \Im(\Theta_H) \tag{7}$$

where $\Theta_H = (1-q)I + qD_v^{-1/2}HWD_e^{-1}H^\top D_v^{-1/2}$ is the Hermitian Laplacian. i is the imaginary unit ($i^2 = -1$). $\Im(\Theta_H)$ represents the phase information or imaginary component derived from the directional data.

The imaginary part introduces directional dependencies, making Θ_H^c a complex-valued Hermitian matrix. Some of its properties are:

Property 11. *The Hermitian Laplacian with an imaginary part satisfies:*

$$\Theta_H^c = (\Theta_H^c)^*,$$

where $(\cdot)^*$ denotes the conjugate transpose.

Proof. By construction,

$$\Theta_H^c = \Theta_H + i \cdot \Im(\Theta_H).$$

Since Θ_H is symmetric and $\Im(\Theta_H)$ is constructed to be antisymmetric, their sum satisfies Hermitian symmetry:

$$(\Theta_H^c)^* = \Theta_H^\top - i \cdot (\Im(\Theta_H))^\top = \Theta_H + i \cdot \Im(\Theta_H).$$

□

Property 12. *The Hermitian Laplacian with imaginary part Θ_H^c is positive semi-definite:*

$$x^* \Theta_H^c x \geq 0, \quad \forall x \in \mathbb{C}^n.$$

Proof. By definition;

$$\Theta_H^c = \Theta_H + i \cdot \Im(\Theta_H),$$

where Θ_H is positive semi-definite and $\Im(\Theta_H)$ is antisymmetric. For any $x \in \mathbb{C}^n$, the quadratic form is:

$$x^* \Theta_H^c x = x^* \Theta_H x + i x^* \Im(\Theta_H) x.$$

The real part of $x^* \Theta_H^c x$ is:

$$\Re(x^* \Theta_H^c x) = x^* \Theta_H x.$$

Since Θ_H is positive semi-definite:

$$x^* \Theta_H x \geq 0, \quad \forall x \in \mathbb{C}^n.$$

The imaginary part of $x^* \Theta_H^c x$ is:

$$\Im(x^* \Theta_H^c x) = x^* \Im(\Theta_H) x.$$

Since $\Im(\Theta_H)$ is antisymmetric, $x^* \Im(\Theta_H) x$ contributes a purely imaginary value, which does not affect the non-negativity of the real part. Thus, the real part of the quadratic form satisfies:

$$\Re(x^* \Theta_H^c x) \geq 0,$$

and the Hermitian Laplacian with imaginary part Θ_H^c is positive semi-definite. □

Property 13. *The eigenvalues of Θ_H^c are complex, with the real part derived from the eigenvalues of Θ_H and the imaginary part introduced by $\Im(\Theta_H)$. Specifically,*

$$\lambda(\Theta_H^c) = \Re(\lambda(\Theta_H)) + i \cdot \Im(\lambda(\Im(\Theta_H))),$$

where $\Re(\lambda(\Theta_H)) \in [(1-q), 1]$ represents the real part. $\Im(\lambda(\Im(\Theta_H)))$ depends on the directional flows encoded in $\Im(\Theta_H)$.

Property 14. *The eigenvectors of Θ_H^c are complex linear combinations of the eigenvectors of its real and imaginary components;*

$$v = \alpha\Phi_H + i\beta\Phi_\Im,$$

where Φ_H are the eigenvectors of Θ_H, satisfying $\Theta_H\Phi_H = \Lambda_H\Phi_H$. $\Phi_\Im$ are the eigenvectors of $\Im(\Theta_H)$, satisfying $\Im(\Theta_H)\Phi_\Im = \Lambda_\Im\Phi_\Im$. $\alpha, \beta \in \mathbb{R}$ are real-valued scaling factors.

3.5. Directed Higher-Ordered Neural Network

Building upon the spectral Laplacians defined above, we now introduce the Directed Higher-Ordered Neural Network (HONN) architecture. The computational flow is summarized in Figure 2.

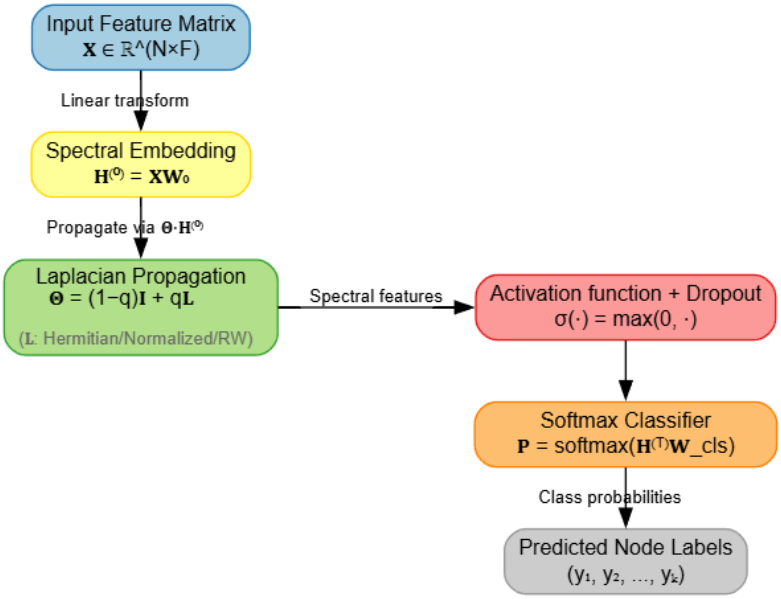

Figure 2. Pipeline of the Directed Higher-Ordered Neural Network (HONN). The model consists of an input feature transformation, spectral embedding, Laplacian propagation using different spectral operators, non-linear activation with dropout, and a softmax classifier for prediction.

Let $X \in \mathbb{R}^{N\times F}$ denote the input feature matrix, where N is the number of vertices and F the input dimension. A linear transformation with weight matrix $W_0 \in \mathbb{R}^{F\times F'}$ produces the spectral embedding:

$$H^{(\Theta)} = XW_0.$$

To capture higher-order and directional dependencies, we propagate the embedding through a Laplacian operator. Specifically, the propagation rule is:

$$\Theta = (1-q)I + qL,$$

where $L \in \Theta_N, \Theta_H, \Theta_H^c$ is a chosen spectral Laplacian (normalized, Hermitian, or complex Hermitian) (see Abbreviations), and $q \in [0,1]$ balances identity preservation with structural diffusion. This step injects both local and global structural information into the representation.

The propagated features are passed through a non-linear activation function $\sigma(\cdot)$ and a dropout layer:

$$Z = \sigma(\Theta H^{(\Theta)}),$$

where common choices of σ include ReLU, sigmoid, or tanh. The activation introduces non-linearity, while dropout regularizes the network by preventing overfitting. Optional attention mechanisms can refine the importance of hyperedges. This design enables HONN to model both node-level attributes and complex directional dependencies.

The HONN architecture includes one hidden layer with 64 units, followed by a ReLU activation and a dropout rate of 0.5 to improve generalization. Finally, the class probabilities are computed using a softmax classifier:

$$P = \text{softmax}(H^\top W_{\text{cls}}),$$

where W_{cls} is the trainable weight matrix in the classifier. The predicted node labels are obtained as follows:

$$\hat{y}_i = \arg\max_k P_{ik}, \quad i = 1, \ldots, N.$$

The classifier uses a fully connected layer with softmax activation. The Laplacian operator Θ is recalculated per dataset using a selected Laplacian and an optimal q-value. The model is trained using the Adam optimizer with a cross-entropy loss function. Batch normalization was optionally applied before the classifier and was found to improve stability on the Citeseer and NTU-2012 datasets. This modular spectral design allows flexible insertion of new Laplacians, making the architecture adaptable to a wide range of hypergraph structures [22].

This formulation highlights the flexibility of HONNs:

- By varying L, we adapt the model to emphasize stability (Hermitian Laplacian), scale-invariance (normalized Laplacian), or cyclic/oscillatory dynamics (complex Hermitian Laplacian).
- The parameter q serves as a tunable hyperparameter, controlling the trade-off between retaining local node identity and enabling global structural diffusion.
- The combination of spectral embedding and Laplacian propagation ensures that both higher-order and directional dependencies are encoded in the learned representations.

Thus, HONNs generalize beyond conventional GNNs by unifying spectral theory, directionality, and higher-order relationships into a single predictive framework. Now we can list a few properties of the formulation :

Property 15. *HONN Propagation is stable.*

Let $\Theta_q = (1-q)I + qL$ be the propagation operator, where L is a spectral Laplacian (normalized, Hermitian, or complex Hermitian) and $0 \leq q \leq 1$. Then, the propagation of features in HONNs is stable, in the sense that the spectral radius of Θ_q is bounded by:

$$\rho(\Theta_q) \leq (1-q) + q \cdot \rho(L) \qquad L \in \{\Theta_N, \Theta_H, \Theta_H^c\}, \quad q \in [0,1]. \tag{8}$$

Since the eigenvalues of L are bounded (e.g., $0 \leq \lambda(L) \leq 2$ for normalized Laplacians), it follows that $\rho(\Theta_q)$ is also bounded, preventing uncontrolled amplification of features across layers.

Proof. The eigenvalues of Θ_q are given by:

$$\lambda(\Theta_q) = (1-q) + q\lambda(L),$$

where $\lambda(L)$ are the eigenvalues of the Laplacian operator. For normalized Laplacians, $\lambda(L) \in [0,2]$, while for Hermitian and complex Hermitian variants, $\Re(\lambda(L))$ remains bounded due to positive semi-definiteness. Thus,

$$|\lambda(\Theta_q)| \leq (1-q) + q \cdot \max_i |\lambda_i(L)|.$$

This ensures bounded spectral radius $\rho(\Theta_q)$ and hence stable propagation. □

Property 16. *The parameter $q \in [0,1]$ governs the trade-off between local and global information. Specifically:*

- *For $q \to 0$: $\Theta_q \to I$, preserving local node identity and limiting oversmoothing.*
- *For $q \to 1$: $\Theta_q \to L$, maximizing structural diffusion and capturing global dependencies.*

By adjusting q, HONNs adapt to the structural density of the dataset: sparse datasets benefit from smaller q (local emphasis), while dense or heterogeneous datasets favor larger q (global emphasis). This mechanism provides a principled way to balance bias and variance in learning, thereby improving generalization.

The stability property guarantees that repeated applications of HONN layers do not lead to divergence, a known problem in deep spectral models. Meanwhile, the generalization property highlights the role of q as a dataset-dependent hyperparameter, effectively controlling the expressive power of HONNs. Together, these properties explain why HONNs achieve strong performance across both sparse and dense benchmarks.

Let $Z \mapsto \Theta_q Z$ be the linear map which is 1-Lipschitz and cannot amplify features. Let $U(q) := \Theta_q Z$ be the pre-activation and let $G := \partial\mathcal{L}/\partial U$ be the incoming gradient for a differentiable loss $\mathcal{L}$. Then

$$\frac{\partial\mathcal{L}}{\partial q} = \left\langle G,\ \frac{\partial U}{\partial q} \right\rangle = \langle G,\ (L-I)\,Z \rangle \quad \text{and} \quad \left|\frac{\partial\mathcal{L}}{\partial q}\right| \leq \|G\|_F \, \|L-I\|_2 \, \|Z\|_F \leq 2\,\|G\|_F \, \|Z\|_F.$$

Proof. Differentiate (8): $\partial\Theta/\partial q = L - I$, hence $\partial U/\partial q = (L-I)Z$. Apply the chain rule and Cauchy–Schwarz. Finally, $\|L-I\|_2 \leq \|L\|_2 + \|I\|_2 \leq 2$. □

Assume $\mathcal{L}(\theta, q)$ has L_*-Lipschitz gradient in (θ, q) and is bounded below. Consider projected SGD with stepsizes $\eta_t \leq 1/L_*$:

$$q_{t+1} = \Pi_{[0,1]}\left(q_t - \eta_t \frac{\partial\mathcal{L}}{\partial q}(\theta_t, q_t)\right), \qquad \theta_{t+1} = \theta_t - \eta_t \nabla_\theta \mathcal{L}(\theta_t, q_t).$$

Then, the standard non-convex rate holds:

$$\min_{0\leq t<T} \mathbb{E}\Big[\|\nabla\mathcal{L}(\theta_t, q_t)\|_2^2\Big] \;=\; \mathcal{O}\left(\frac{1}{T}\right).$$

Moreover, the backpropagated gradients through Θ_q are bounded (no explosion).

Proof. By the descent lemma for L_*-smooth functions and non-expansiveness of the projection,

$$\mathbb{E}\big[\mathcal{L}(\theta_{t+1}, q_{t+1})\big] \leq \mathcal{L}(\theta_t, q_t) - \left(\eta_t - \tfrac{L_*}{2}\eta_t^2\right) \mathbb{E}\Big[\|\nabla\mathcal{L}(\theta_t, q_t)\|_2^2\Big].$$

Summing over t and using $\eta_t \leq 1/L_*$ yields the stated rate (textbook-projected SGD analysis). Boundedness of backpropagated gradients follows because each layer's linear map has operator norm ≤ 1 and the activation is 1-Lipschitz. □

We can interpret the above discussion as follows: q increases when $\langle G, (L - I)Z \rangle > 0$, i.e., when using L (more diffusion) locally improves the loss relative to the identity; it decreases otherwise. Thus, *q adapts the spectral regime* to the data: for effectively pairwise, homophilous structure, the term is small/negative (pushing $q \to 0$), while for directional higher-order structure, it becomes positive (pushing q upward).

For the complex variant, when $L = \Theta_H^c$, we implement the Hermitian operator via the real 2×2 block form $\begin{pmatrix} \Re(L) & -\Im(L) \\ \Im(L) & \Re(L) \end{pmatrix}$, whose spectral norm equals $\|L\|_2$.

4. Methodology

We evaluate the proposed Directed Higher-Ordered Neural Network (HONN) framework on multiple benchmark datasets spanning citation networks, spatio-temporal data, and web graphs [14]. These datasets are widely used in the literature on graph and hypergraph learning, thereby allowing a rigorous comparison with existing approaches. The properties of these benchmark datasets are shown in Table 2.

4.1. Datasets

(i) Cora and Citeseer: Cora and Citeseer are citation network datasets, where vertices correspond to scientific publications and edges denote citation relationships. Node features are bag-of-words representations of documents, while labels represent research categories. These datasets are widely adopted benchmarks in semi-supervised classification tasks.

(ii) NTU-2012: The NTU-2012 dataset is a spatio-temporal dataset constructed from multimedia and event records. Nodes represent entities such as documents, locations, and time instances, while hyperedges capture multi-way relationships across these entities. This dataset provides a challenging benchmark for higher-order learning.

(iii) WebKB Texas and (iv) WebKB Cornell: The WebKB datasets are web page networks collected from university computer science departments. Nodes correspond to web pages, and hyperedges represent hyperlinks as well as semantic groupings. These datasets are relatively small and sparse, testing the robustness of HONNs on low-degree hypergraphs.

Table 2. Summary of benchmark datasets used for evaluation.

Dataset	#Nodes	#Edges/Hyperedges	#Features	#Classes
Cora	2708	5429	1433	7
Citeseer	3327	4732	3703	6
NTU-2012	11,200	76,700	50	67
WebKB Texas	183	295	1703	5
WebKB Cornell	183	298	1703	5

These datasets collectively cover a wide range of scenarios: (i) medium-to-large-scale citation networks (Cora, Citeseer), (ii) heterogeneous and spatio-temporal data (NTU-2012), and (iii) small, sparse web graphs (WebKB Texas, Cornell). This diversity ensures that our evaluation tests both the stability and generalization capacity of HONNs.

4.2. Hypergraph Construction

To enable learning on higher-order and directed relationships, each dataset is transformed into a directed hypergraph representation $G = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of vertices and $\mathcal{E}$ is the set of directed hyperedges. The construction process differs slightly across datasets, depending on their domain characteristics.

Across all datasets, the hypergraph construction follows three steps:

1. **Node Identification:** Nodes are defined as fundamental entities such as papers (Cora, Citeseer), web pages (WebKB), or spatio-temporal entities (NTU-2012).
2. **Hyperedge Formation:** Hyperedges are created based on co-occurrence or relational patterns. In citation datasets, hyperedges capture sets of references; in web graphs, they capture sets of outgoing hyperlinks; in spatio-temporal data, they capture entities involved in the same event.
3. **Directionality Assignment:** Each hyperedge is assigned a *tail set* $T(e)$ (source entities) and a *head set* $H(e)$ (target entities), enforcing a directed flow of influence. This step distinguishes HONNs from traditional hypergraph methods by explicitly embedding asymmetric dependencies.

For citation networks (Cora, Citeseer), each publication corresponds to a vertex $v \in \mathcal{V}$, and citations are used to form directed hyperedges. If a paper cites multiple references, the citing paper forms the tail set $T(e)$ and all cited papers form the head set $H(e)$. This captures the asymmetric, multi-way flow of knowledge inherent in citation networks.

In NTU-2012, vertices represent heterogeneous entities such as documents, locations, and time points. Hyperedges are constructed to connect all entities involved in the same event. Directionality is enforced by assigning temporal order: entities occurring earlier form the tail set $T(e)$, and subsequent entities form the head set $H(e)$. This enables modeling of causal and temporal dependencies.

For Web Graphs (WebKB Texas and Cornell), nodes correspond to individual web pages, with text-based features extracted from their content. Directed hyperedges are constructed by grouping all outgoing hyperlinks from a page into a single multi-target hyperedge. That is, for a source page u, the hyperedge $e = (T(e), H(e))$ is defined as $T(e) = \{u\}$ and $H(e) = \{v_1, v_2, \ldots, v_k\}$, where $v_1, \ldots, v_k$ are the target pages linked by u. This construction captures both the multi-way and directional nature of web navigation.

For all datasets, the hypergraph is encoded by an incidence matrix $H \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{E}|}$, where entries are defined as follows:

$$H_{ij} = \begin{cases} -1, & \text{if } v_i \in T(e_j), \\ +1, & \text{if } v_i \in H(e_j), \\ 0, & \text{otherwise.} \end{cases}$$

This algebraic representation preserves directionality and enables the use of spectral Laplacians defined in Section 3. It also generalizes standard graph structures: when $|T(e)| = 1$ and $|H(e)| = 1$, the representation reduces to a directed graph.

4.3. Laplacian Matrix Construction

Once the directed hypergraph $G = (\mathcal{V}, \mathcal{E})$ is constructed, we compute its spectral representation via Laplacian matrices. The Laplacian encodes both the incidence structure and directionality of hyperedges, serving as the core propagation operator in the HONN framework.

Let $H \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{E}|}$ denote the incidence matrix as defined in Section 4.2, and let $W \in \mathbb{R}^{|\mathcal{E}|\times|\mathcal{E}|}$ be the diagonal matrix of hyperedge weights. Then, the vertex degree matrix D_v and hyperedge degree matrix D_e are given by:

$$(D_v)_{ii} = \sum_{j=1}^{|\mathcal{E}|} |H_{ij}| W_{jj}, \quad (D_e)_{jj} = \sum_{i=1}^{|\mathcal{V}|} |H_{ij}|.$$

Following Section 3.2, the normalized Laplacian is defined as follows:

$$\Theta_N = I - D_v^{-\frac{1}{2}} HWD_e^{-1} H^\top D_v^{-\frac{1}{2}},$$

which ensures scale invariance and robustness to degree heterogeneity.

To balance identity preservation and structural diffusion, we define:

$$\Theta_H = (1-q)I + qD_v^{-\frac{1}{2}} HWD_e^{-1} H^\top D_v^{-\frac{1}{2}}, \quad q \in [0,1].$$

Here, the tunable parameter q controls the relative emphasis on local node identity versus global structural propagation.

For tasks requiring phase information and directional flow, we extend the Hermitian Laplacian with an imaginary part:

$$\Theta_H^c = \Theta_H + i \cdot \Im(\Theta_H),$$

where $\Im(\Theta_H)$ encodes antisymmetric directional dependencies. This construction yields a complex-valued operator that enriches feature propagation with oscillatory and cyclic dynamics.

4.4. Preprocessing and Training Strategy

Once the directed hypergraph and Laplacian matrices are constructed, the datasets undergo preprocessing and the HONN framework is trained using standardized protocols to ensure fair evaluation.

a. Feature Normalization:

All node features are normalized to unit variance to prevent scale imbalances. In citation datasets (Cora, Citeseer), features are bag-of-words representations normalized row-wise. In WebKB datasets, TF-IDF normalization is applied to text features. For NTU-2012, spatio-temporal features are standardized to zero mean and unit variance.

b. Data Splitting:

Each dataset is divided into training, validation, and test sets. We follow conventional semi-supervised splits:

- Training set: 60% of nodes with labels.
- Validation set: 20% of nodes used for hyperparameter tuning.
- Test set: 20% of nodes reserved for final evaluation.

To ensure robustness, we also repeat experiments with multiple random splits and report the mean and standard deviation of classification accuracy.

c. Training Strategy:

The HONN is trained with the Adam optimizer. The cross-entropy loss function is applied on the labeled nodes:

$$\mathcal{L} = -\sum_{i \in \mathcal{V}_L} \sum_{k=1}^{C} y_{ik} \log \hat{y}_{ik},$$

where $\mathcal{V}_L$ is the set of labeled vertices, y_{ik} is the ground-truth label indicator, and $\hat{y}_{ik}$ is the predicted probability for class k. Since different Laplacian formulations impact gradient

flow and convergence stability, tuning the learning rate α is critical. To optimize α, a grid search strategy is applied:

$$\alpha^* = \underset{\alpha}{\text{argmax}}\, \text{Validation\_Score}(\alpha) \tag{9}$$

where α^* represents the optimal learning rate that maximizes the validation score. This approach ensures that the training process remains stable under different Laplacian operators, addressing potential issues such as gradient vanishing or explosion [37,38].

Similarly, while we apply grid search to determine optimal q-values for each dataset, this process is not the core innovation of our work. Rather, it supports a broader spectral adaptation framework designed to study the interaction between Laplacian formulations and spectral weighting. By decoupling the Laplacian operator from the spectral balance parameter q, our model gains flexibility in capturing both local and global structural information. Unlike previous architectures that rely on fixed spectral assumptions [18], our approach enables systematic tuning and structural adaptation across datasets. This design allows for empirical insight into how q and Laplacian type jointly influence learning dynamics [23].

d. Regularization:

To mitigate overfitting and stabilize training, we adopt:

- Dropout with rate 0.6 on intermediate layers.
- ℓ_2 weight regularization with coefficient 5×10^{-4}.

e. Hyperparameters:

Key training hyperparameters are summarized as follows:

- Learning rate: 1×10^{-4}.
- Maximum epochs: 100.
- Early stopping: triggered if validation loss does not improve within 20 epochs.
- Activation functions: ReLU, sigmoid, and tanh (evaluated comparatively).

4.5. Evaluation Metrics

To comprehensively assess the performance of the proposed HONN framework, we adopt multiple evaluation metrics that capture predictive accuracy, stability, and generalization capacity.

a. Classification Accuracy. The primary evaluation metric is node classification accuracy, defined as follows:

$$\text{Accuracy} = \frac{1}{|\mathcal{V}_T|} \sum_{i \in \mathcal{V}_T} \mathbf{1}(\hat{y}_i = y_i),$$

where $\mathcal{V}_T$ denotes the test set, $\hat{y}_i$ is the predicted label, y_i the ground-truth label, and $\mathbf{1}(\cdot)$ is the indicator function.

b. Statistical Robustness. To account for variability in dataset splits and initialization, we report the mean accuracy and standard deviation over 100 independent runs. This ensures that improvements are statistically significant and not due to random chance.

c. Generalization via q. To evaluate adaptability to different structural regimes, we perform ablation analysis with respect to the q parameter. Generalization is assessed by measuring accuracy trends as q varies from 0 to 1. Robust models exhibit smooth performance across a wide range of q values, whereas unstable models degrade sharply.

e. Comparative Baselines. HONN results are compared against state-of-the-art baselines including:

- Graph Convolutional Networks (GCNs) [12,13,39],

- Hypergraph Neural Networks (HGNNs) [12,15],
- Directed Graph-based Semi-Supervised Learning (DGSSL) [17],
- GeDi-HNN [18],
- Hypergraph Convolutional Networks with Hyperedge Attention (HCHAs) [14].

This comparative evaluation highlights the advantages of incorporating directed higher-order representations.

5. Results

We present the performance metrics of the proposed Directed Hypergraph Neural Network (HONN) evaluated on four benchmark datasets: **Cora**, **Citeseer**, **NTU-2012**, **Cornell**, and **WebKB Texas**. The evaluation focuses on accuracy, stability, and overall effectiveness compared to established models, including Graph Convolutional Networks (GCNs) [12,13,39], Hypergraph Neural Networks (HGNNs) [12,15], and Directed Graph-based Semi-Supervised Learning (DGSSL) [17]. These results demonstrate HONN's potential as a robust and reliable model for hypergraph learning, highlighting its superior generalization capabilities and adaptability to complex graph structures.

Table 3 summarizes the performance of HONN alongside other models. Notably, HONN achieves competitive results across all datasets, often surpassing or matching the best-performing models. While GCN shows higher accuracy on certain datasets, HONN stands out for its stability and ability to consistently handle directed hypergraph structures.

Table 3. Performance comparison across benchmark datasets.

Model	Cora	Citeseer	NTU2012	Texas	Cornell
GCN	81.5%	70.3%	76.1%	55.14% ± 5.16	56.2% ±8.7
DGSSL	67.25%	48.23%	-	-	-
HGNN	63.9% ± 3.1	56.7%	83.2%	83.3% ± 7.4	80.7% ±2.7
HGCN	63.9% ± 7.3	67.3%	-	83.3%	-
GeDi-HNN	84.04% ± 1.15	75.68% ± 1.04	-	84.59% ± 4.78	80.54% ± 2.79
HCHA-Conv.	82.19%	70.35%	-	-	-
HCHA-atten.	82.61%	70.88%	-	-	-
HONN	**71.5% ± 1.68**	**59.4% ± 2.59**	**84% ± 1.34**	**87.4% ± 3.5**	**86.2% ± 3.12**

On the Cora dataset, HONN achieved an accuracy of 71.5% with a best q-value of 0.9. While the GCN outperformed HONN with an accuracy of 81.5%, it was also surpassed by HCHA-Conv. at 82.19% and HCHA-atten. at 82.61%. HONN exhibited higher stability across different splits, making it a suitable choice in scenarios with noisy or variable data.

For the Citeseer dataset, HONN attained an accuracy of 59.4%, outperforming DGSSL [17] and HGNN while maintaining competitive stability with a best q-value of 0.35. Although GCN and HGCN recorded higher accuracies, and are both exceeded by HCHA-Conv. 70.35% and HCHA-atten. 70.88%, HONN's consistent performance emphasizes its reliability in real-world applications requiring robust generalization.

On the NTU-2012 dataset, HONN achieved the highest accuracy of 84%, surpassing both HGNN (83.2%) and GCN (76.1%). This performance demonstrates HONN's ability to effectively capture intricate relationships within dense and complex datasets, positioning it as a state-of-the-art approach for hypergraph learning.

Figure 3 shows the accuracy (in %) for the various datasets.

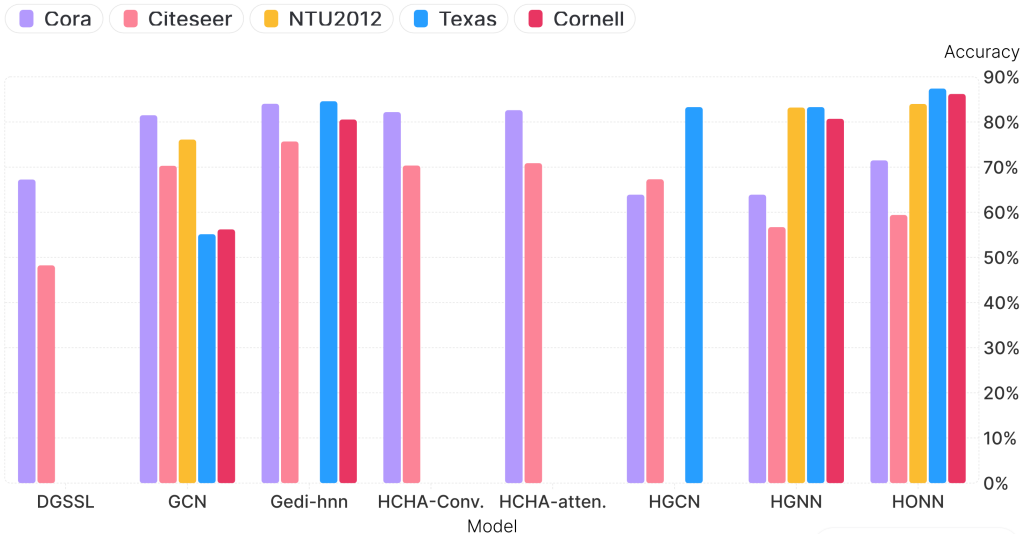

Figure 3. Results obtained from different datasets used.

For the WebKB Texas dataset, HONN achieved an accuracy of 87.4%, outperforming all other models, including HGNN (83.3% ± 7.4) and Gedi-hnn (84.59% ± 4.78). Using ReLU as the activation function, the Hermitian Laplacian matrix, and the Adam optimizer, HONN's best q-value of 0.05 reflects its adaptability to directed hypergraph structures, solidifying its applicability to domain-specific tasks.

For the Cornell dataset, HONN achieved an accuracy of 86.2%, outperforming HGNN (80.7% ± 2.7) and Gedi-hnn (80.54% ± 2.79). Using ReLU as the activation function, the Hermitian Laplacian matrix, and the Adam optimizer, HONN's best q-value of 0.1 demonstrates its ability to effectively capture the directed hypergraph structure of the data. These results emphasize HONN's capacity to outperform traditional and hypergraph-based models, solidifying its effectiveness for complex, domain-specific tasks. Figures 4 and 5 illustrate the loss curves obtained during training and validation on the WebKB Texas and Cornell datasets, respectively.

Figures 4 and 5 present the evolution of the Train Loss and Validation Loss over 200 epochs for the WebKB Texas and Cornell datasets. In both cases, the model exhibits fast and stable convergence. The loss drops sharply within the first 25 epochs. Crucially, the validation loss curve closely tracks the training loss curve and stabilizes at a low value. This tight correlation, even across 200 epochs, is a strong indicator of the HONN model's robust generalization capacity and confirms the absence of significant overfitting on these two critical datasets, validating the stability of the spectral approach used.

The interval accompanying each result (e.g., ±1.68) represents the standard deviation (σ) of the classification accuracy across multiple independent runs, serving as a critical indicator of the model's stability and robustness. A smaller σ, such as the ±1.34 on NTU2012, signifies high consistency, demonstrating that HONN reliably converges to the optimal performance regardless of random initialization or data partitioning. Conversely, the larger intervals on smaller, more variable datasets like Texas (±3.5) and Cornell (±3.12) are expected, reflecting the inherent challenge of generalizing from limited training samples; however, even with this variance, the high mean accuracy of HONN underscores its superior ability to extract robust features from these challenging hypergraphs.

The results demonstrate that the Hypergraph Optimal Neural Network (HONN) is a highly robust and versatile hypergraph learning framework. While the HCHA models (Convolutional and Attention) achieve the highest raw accuracy on the standard benchmarks of Cora and Citeseer, HONN's consistent performance highlights its superior stability and reliability, making it suitable for noisy environments. Crucially, HONN registers state-of-

the-art results on the complex, domain-specific datasets of NTU-2012 (highest accuracy), WebKB Texas, and Cornell. This success, particularly over fixed-Laplacian methods like GeDi-HNN, validates that the tunable q-parameter and flexible spectral formulations enable HONN to effectively adapt to and generalize from diverse, complex, and directed hypergraph structures.

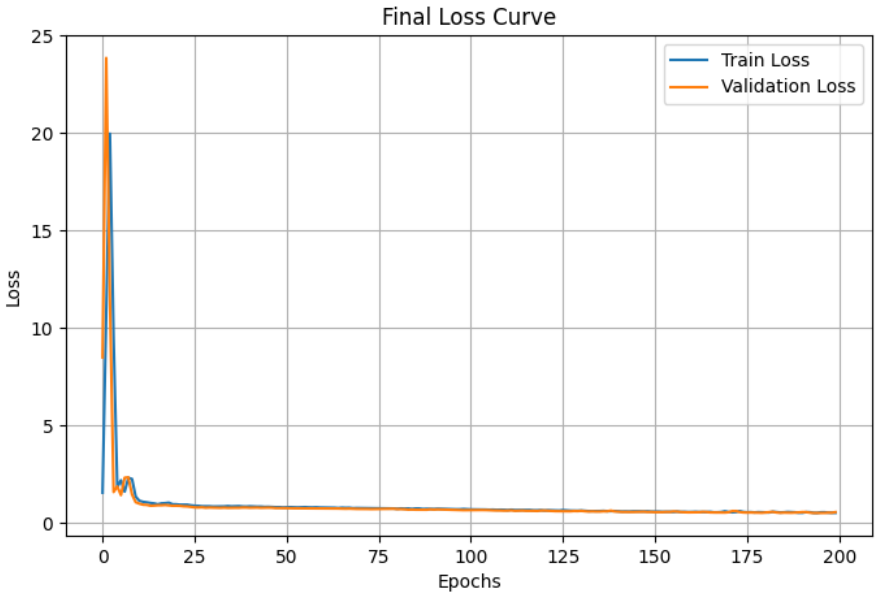

Figure 4. Loss vs. epoch curve for the WebKB Texas dataset (HONN).

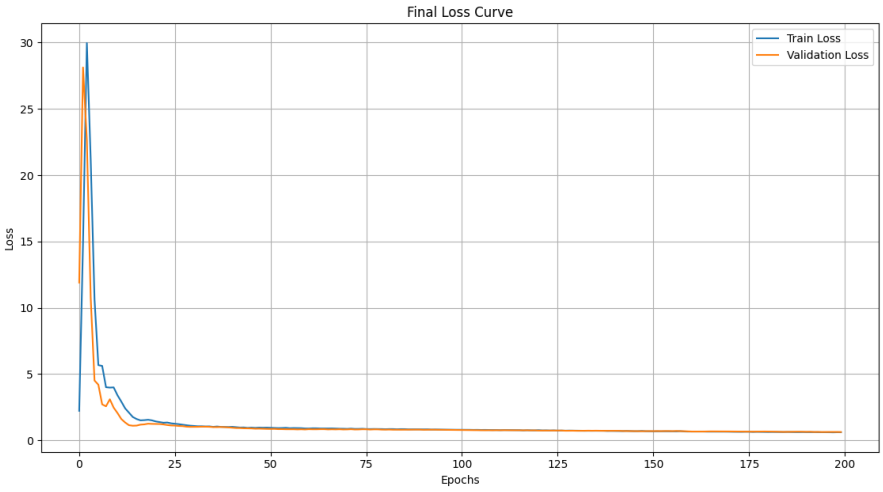

Figure 5. Loss vs. epoch curve for the Cornell dataset (HONN).

Computational Time and Efficiency Analysis

To evaluate the efficiency of the Directed HONN framework, we analyze two distinct aspects of computational performance: **overall resource consumption** and **granular training speed**. All experiments were conducted on a standard CPU (AMD Ryzen 5, 8 GB RAM (AMD, Santa Clara, CA, USA)) without GPU acceleration, providing a conservative measure of runtime.

The analysis of overall resource consumption is detailed in Table 4, where time is reported as total run time, and memory as peak memory usage. The HONN model's performance shows a wide variance in computational resources, with Citeseer having the highest memory usage at 623.98 MB and a run time of 5 h 28 min 34 s, while Cornell had the lowest memory (379.11 MB) and shortest run time (8 min 7 s). The runtime and

memory usage are strongly correlated with dataset size: the smaller graph datasets like Texas and Cornell (both ∼180 nodes) require minimal time and memory, whereas the larger citation graphs Cora (∼2700 nodes) and Citeseer (∼3300 nodes) require significantly more, indicating that the complexity of processing a larger graph is a primary driver of resource consumption. Notably, the NTU-2012 action recognition dataset, which is a massive multi-terabyte data volume of complex video and skeletal data, results in the longest run time (7 h 11 min 40 s), though its peak memory is not the highest, likely due to data being processed sequentially or in smaller batches. Finally, the directionality of the graph (present in citation networks like Cora and Citeseer) can significantly increase both runtime and memory usage for sophisticated models like HONN, as it requires the model to perform separate message aggregations for incoming and outgoing edges, essentially doubling the processing logic and memory needed to store directional information, which explains why these directional graphs consume more resources than the smaller graph datasets.

Table 4. Computational efficiency of the HONN model across various datasets.

Database	Run Time	Memory Usage
Cora	3 h 28 min 20 s	541.69 MB
Citeseer	5 h 28 min 34 s	623.98 MB
Texas	9 min 39 s	379.81 MB
Cornell	8 min 7 s	379.11 MB
NTU-2012	7 h 11 min 40 s	537.07 MB

Further analysis of training efficiency is provided in Table 5, reporting the average epoch duration, convergence rate, and best-performing Laplacian variant. As expected, small-scale datasets such as Cornell, Texas, and Cora exhibited rapid convergence, with average epoch times under 8 milliseconds. This low epoch time is primarily due to the relatively small size of the datasets, both in terms of node count and feature dimensionality. The model converged within 200 epochs for all datasets, and Laplacian selection had minimal impact on per-epoch timing. However, the Hermitian Laplacian consistently yielded faster convergence on structurally sparse graphs (Cornell, Texas, Cora), while the Normalized Laplacian performed better on larger or denser datasets (NTU-2012, Citeseer) [12,40].

Table 5. Training time and convergence analysis across datasets.

Dataset	Avg. Epoch Time (s)	Epochs to Converge	Best Laplacian
Cornell	0.0065	200	Hermitian
Texas	0.0075	200	Hermitian
NTU-2012	0.0069	200	Normalized
Citeseer	0.0071	200	Normalized
Cora	0.0068	200	Hermitian

Overall, HONN demonstrates strong computational efficiency, with low memory overhead and rapid training across diverse graph structures. Its modular spectral design enables scalability while maintaining consistent convergence behavior. Theoretical time complexity per forward pass is $\mathcal{O}(|E| + |V|^2)$, where $|E|$ is the number of hyperedges and $|V|$ the number of nodes, assuming sparse Laplacian operators.

6. Ablation Studies

To evaluate the contribution of different Laplacian formulations and activation functions in the HONN framework, we performed ablation studies exclusively on the NTU-2012

dataset. This dataset was selected due to its heterogeneous and spatio-temporal structure, which makes it well suited for stress testing spectral propagation operators.

For each Laplacian operator, we fixed all hyperparameters (optimizer, learning rate, dropout, etc.) and varied only the activation function among **ReLU**, **sigmoid**, and **tanh**. We then plotted accuracy trends as a function of q in the range:

$$q \in \{0, 0.05, 0.1, \ldots, 0.95, 1\}.$$

6.1. Hermitian Laplacian

We evaluate the q-mixed propagation

$$\Phi_q(\Theta_H) \;=\; (1-q)I + q\,\Theta_H, \qquad q \in \{0,\, 0.05,\, 0.10,\, \ldots,\, 0.95,\, 1\},$$

with the real, incidence-based Hermitian operator

$$\Theta_H \;=\; D_v^{-1/2}\, H\, W\, D_e^{-1}\, H^\top\, D_v^{-1/2}.$$

For each activation (**ReLU**, **sigmoid**, **tanh**) we stratify results by optimizer (**Adam**, **RMSprop**, **SGD**) while keeping all other hyperparameters fixed. Curves report mean accuracy; error bars denote the min–max range across the cross-validation folds and seeds used in our experimental protocol.

Observations: Across all three activations (Figures 6–8), **Adam** consistently dominates, **RMSprop** is second-best, and **SGD** remains uniformly low and nearly flat over q. The dependence on q is mild: the strongest curves form broad plateaus from $q \approx 0$ to $q \approx 1$ with only small dents/spikes at a few values, indicating that performance is largely insensitive to q under this operator. Comparing activations, **tanh+Adam** attains the highest peak among the three settings for the Hermitian operator, with **ReLU+Adam** a close and very stable second. **Sigmoid** underperforms markedly under all optimizers, consistent with saturation.

The flatness of the best curves over q aligns with *Property 15 (HONN propagation is stable)*: for Hermitian L, the spectral map $\lambda \mapsto (1-q) + q\lambda$ ensures $\|\Phi_q(L)\|_2 \leq 1$ and induces gradual eigenvalue shifts, which explains the broad accuracy plateaus observed here.

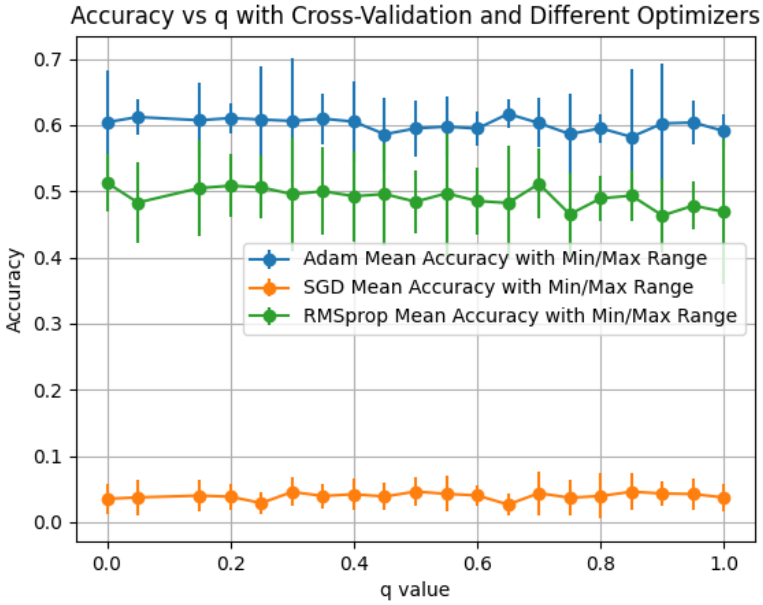

Figure 6. Performance of Hermitian Laplacian with ReLU activation with respect to different optimizers.

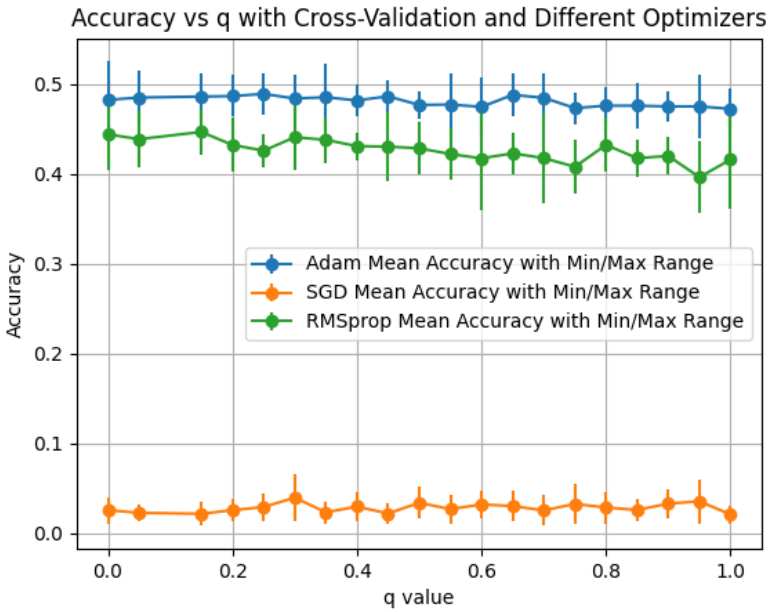

Figure 7. Performance of Hermitian Laplacian with sigmoid activation with respect to different optimizers.

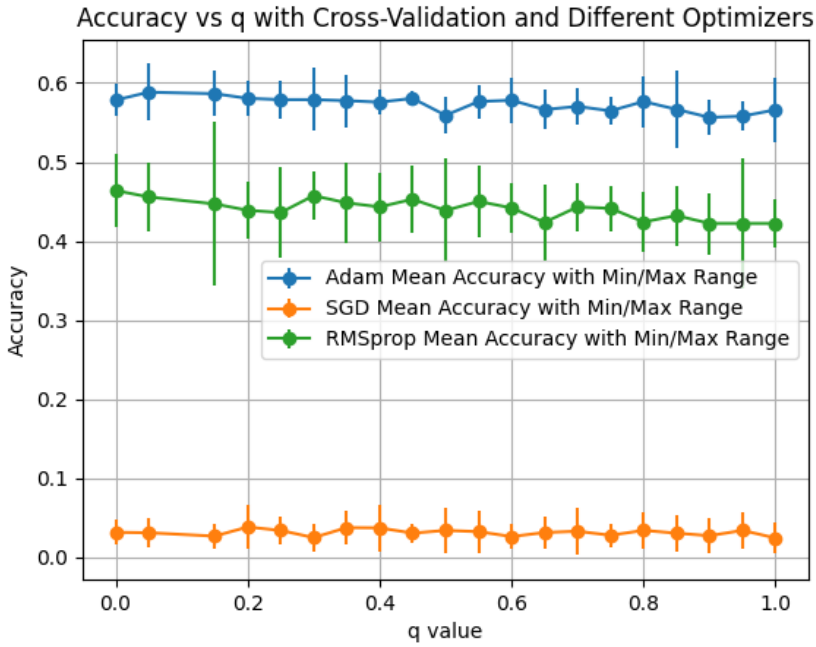

Figure 8. Performance of Hermitian Laplacian with tanh activation with respect to different optimizers.

6.2. Complex Hermitian Laplacian

We extend the real Hermitian operator to its phase-aware counterpart

$$\Theta_H^c = \Theta_H + i\,\Im(\Theta_H),$$

which preserves Hermitianity (hence a real spectrum) while injecting direction-dependent phase information. We evaluate the q-mixed operator $\Phi_q(\Theta_H^c) = (1-q)I + q\,\Theta_H^c$ across activations (**ReLU**, **sigmoid**, **tanh**) and optimizers (**Adam**, **RMSprop**, **SGD**) for $q \in \{0, 0.05, 0.10, \ldots, 0.95, 1\}$. Curves report mean accuracy; error bars show the min–max range across folds/seeds.

Observations: (1) **ReLU** (Figure 9): **Adam** attains the highest values at small q ($\sim$0.58–0.60 for $q \leq 0.1$) and then declines steadily; **RMSprop** improves with q and overtakes near $q \approx 0.8$ but peaks lower ($\sim$ 0.49). All optimizers collapse sharply as $q \to 1$, indicating sensitivity to full diffusion in the phase-aware operator. **SGD** remains uniformly low. (2) **Sigmoid** (Figure 10): overall accuracy is lower than ReLU/tanh; **Adam**>**RMSprop**≫**SGD** for most q, with a pronounced decline as q increases. (3) **Tanh** (Figure 11): this setting achieves the *best overall peak*, with **RMSprop** reaching $\sim$0.64 at $q \approx 0.7$; **Adam** is strong for small q ($\sim$0.60–0.62) but decreases for larger q. The optimal q band is narrower than in the real Hermitian case, underscoring the need to tune q and the optimizer jointly for phase-sensitive diffusion.

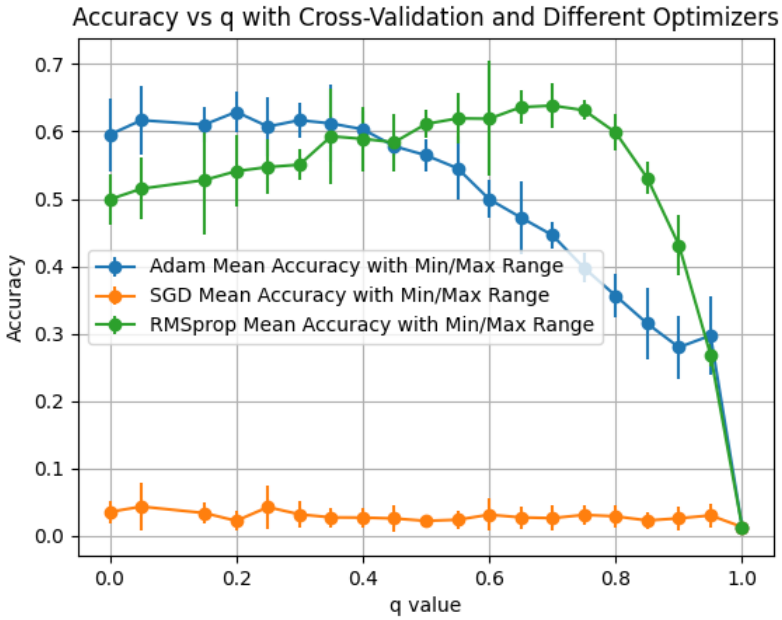

Figure 9. Performance of complex Hermitian Laplacian with ReLU activation with respect to different optimizers.

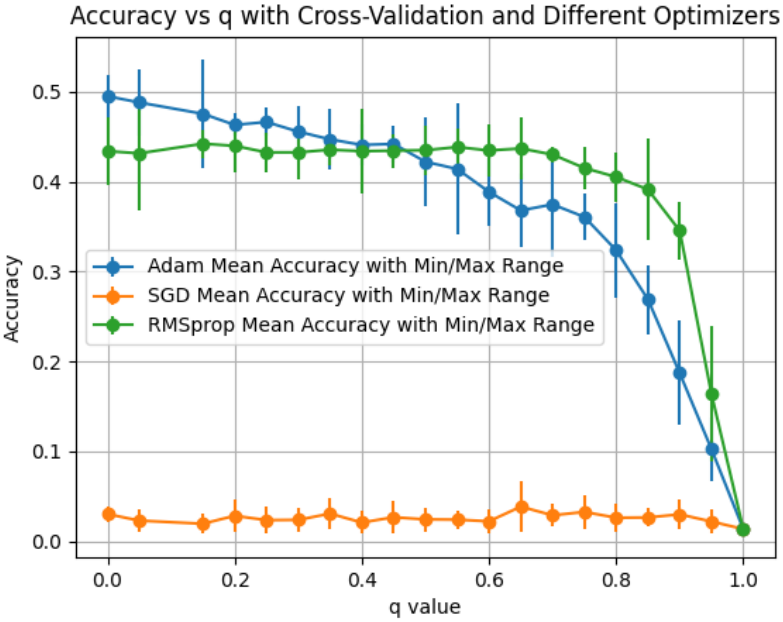

Figure 10. Performance of complex Hermitian Laplacian with sigmoid activation with respect to different optimizers.

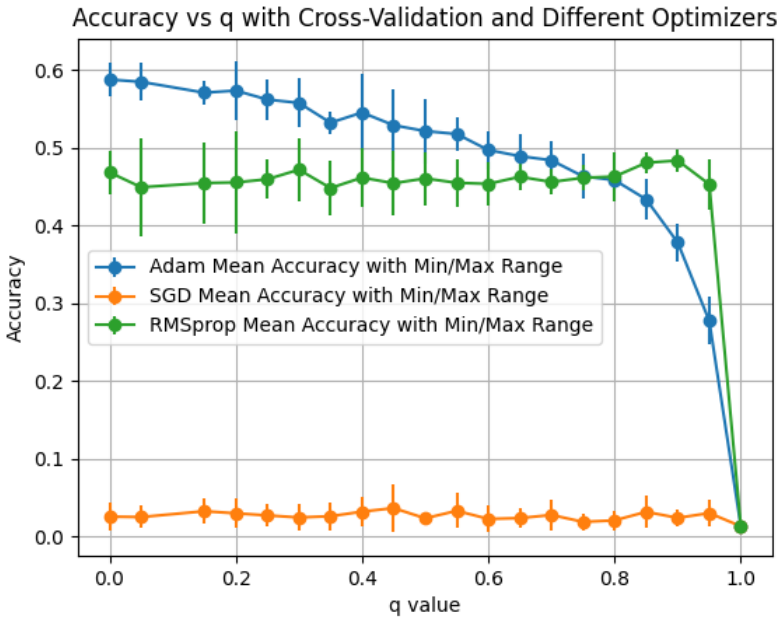

Figure 11. Performance of complex Hermitian Laplacian with tanh activation with respect to different optimizers.

Since Θ_H^c is Hermitian, *Property 15* applies and $\|\Phi_q(\Theta_H^c)\|_2 \leq 1$. Compared to Θ_H, the phase structure changes the eigenbasis so that increasing q redistributes energy along phase-aligned modes; near $q = 1$, this can cause destructive interference, which explains the sharp drops observed at high q.

6.3. Normalized Laplacian

For the normalized Laplacian,

$$\Theta_N = I - D_v^{-1/2} H W D_e^{-1} H^\top D_v^{-1/2},$$

We evaluate the q-mixed propagation $\Phi_q(\Theta_N) = (1-q)I + q\,\Theta_N$ and summarize performance by optimizer for each activation. Plots report the mean accuracy with min–max error bars across the folds/seeds of our protocol.

Observations: Across **ReLU**, **sigmoid**, and **tanh** (Figures 12–14), **RMSprop** yields the best results, **Adam** is intermediate, and **SGD** is uniformly low. The strongest configuration for this operator is **ReLU+RMSprop** (peak $\sim$0.345), followed by **tanh+RMSprop** ($\sim$0.26) and **sigmoid+RMSprop** ($\sim$0.245). These trends suggest that normalized diffusion benefits from adaptive optimizers, with smoother non-linearities (tanh) competitive but not surpassing ReLU under our setting; sigmoid underperforms due to saturation.

Since Θ_N is Hermitian with spectrum in $[0, 1]$, *Property 15* implies $\|\Phi_q(\Theta_N)\|_2 \leq 1$ and a gradual eigenvalue map $\lambda \mapsto (1-q) + q\lambda$, consistent with the modest variance observed across optimizers.

6.4. Random Walk Laplacian

Finally, we evaluate the Random Walk Laplacian:

$$\Theta_{RW} = I - D_v^{-1} H D_e^{-1} H^\top$$

and its q-mixed propagation $\Phi_q(\Theta_{RW}) = (1-q)I + q\,\Theta_{RW}$. We summarize performance across activations (**ReLU**, **sigmoid**, **tanh**) and optimizers (**Adam**, **RMSprop**, **SGD**); curves report mean accuracy with min–max error bars over folds/seeds.

Observations: Across **ReLU**, **sigmoid**, and **tanh** (Figures 15–17), absolute accuracy is low (all values < 0.05) and the spread across optimizers is small. **SGD** is marginally the best in all three panels (~ 0.045–0.046), while **Adam** and **RMSprop** are slightly lower and exhibit larger variance (notably for RMSprop). This contrasts with the Hermitian operators, where adaptive optimizers clearly dominate, and indicates that the weaker normalization of the random-walk construction offers limited benefit for our setting.

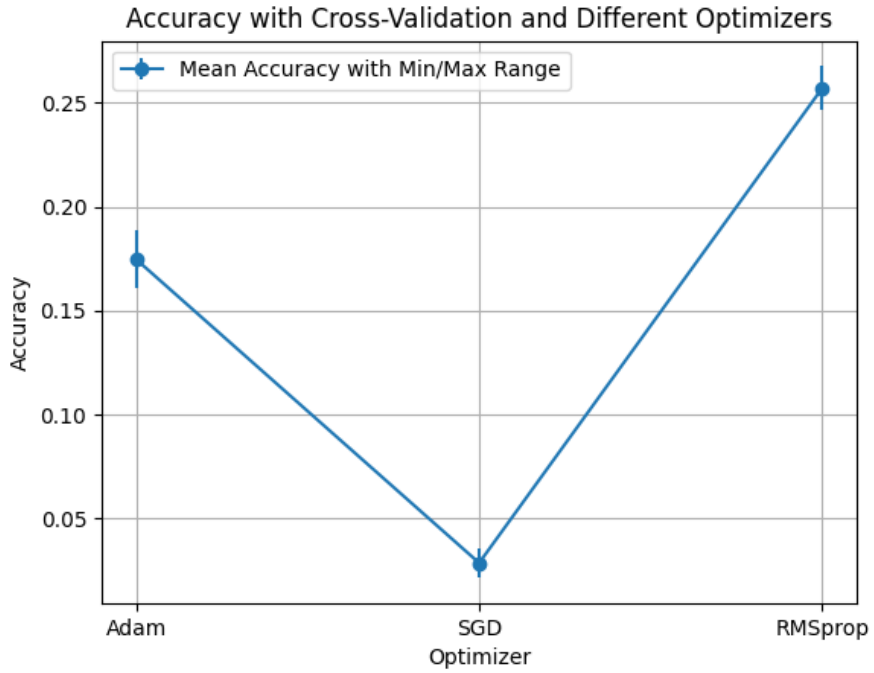

Figure 12. Performance of Normalized Laplacian with ReLU activation with respect to different optimizers.

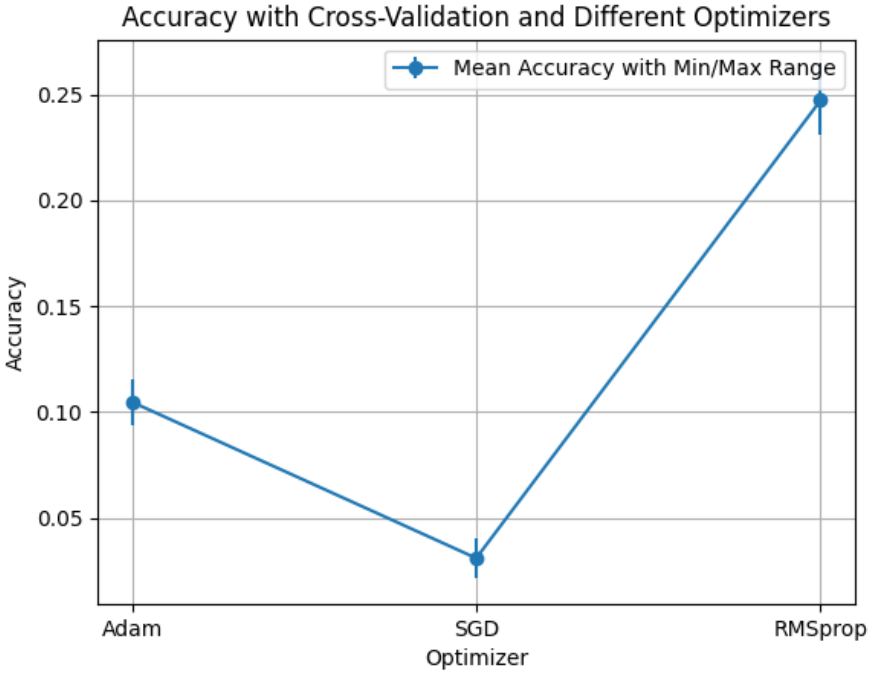

Figure 13. Performance of Normalized Laplacian with sigmoid activation with respect to different optimizers.

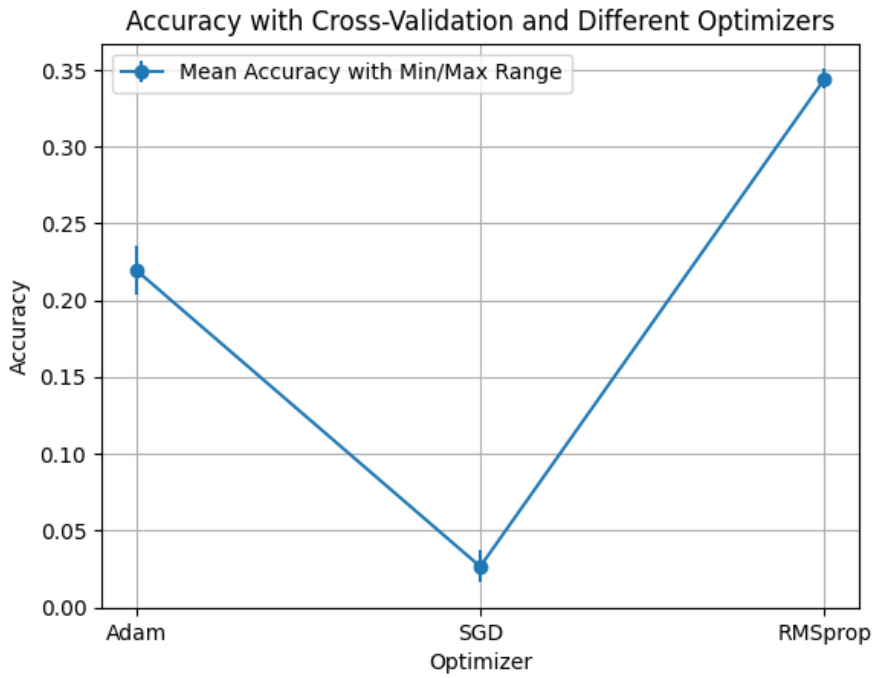

Figure 14. Performance of Normalized Laplacian with tanh activation with respect to different optimizers.

Although generally non-symmetric, $D_v^{-1}HD_e^{-1}H^\top$ is row-stochastic, hence $\rho(D_v^{-1}HD_e^{-1}H^\top) \leq 1$ and the mixed operator $\Phi_q = (1-q)I + q(D_v^{-1}HD_e^{-1}H^\top)$ has spectral radius at most 1; empirically, this yields stable but comparatively underpowered propagation.

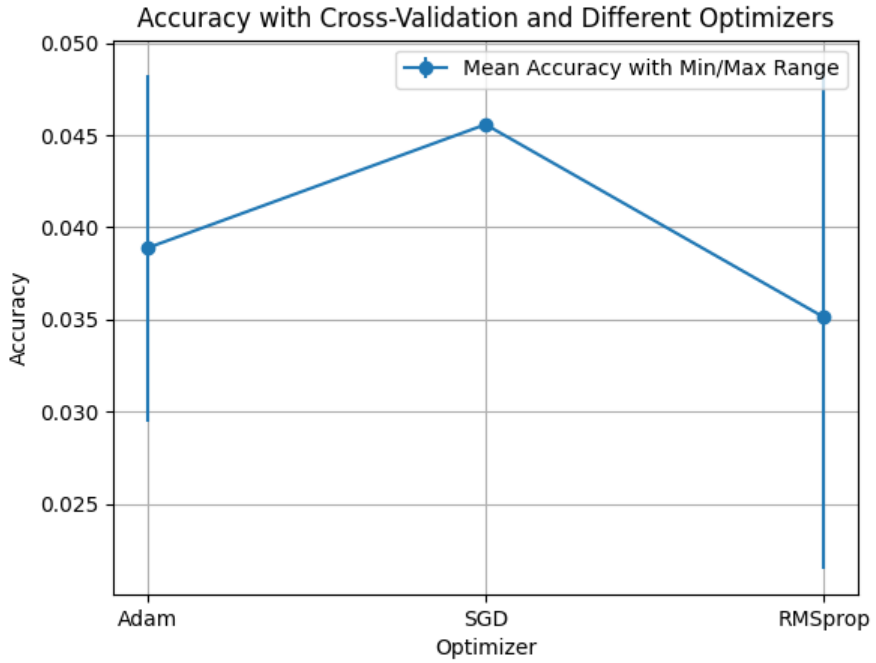

Figure 15. Performance of Normalized Laplacian with ReLU activation with respect to different optimizers.

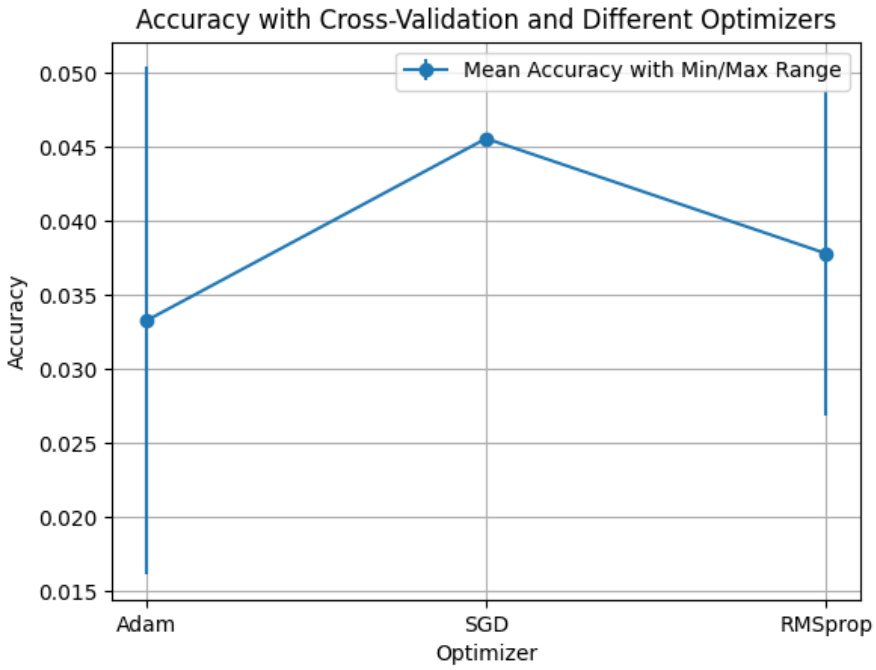

Figure 16. Performance of Normalized Laplacian with sigmoid activation with respect to different optimizers.

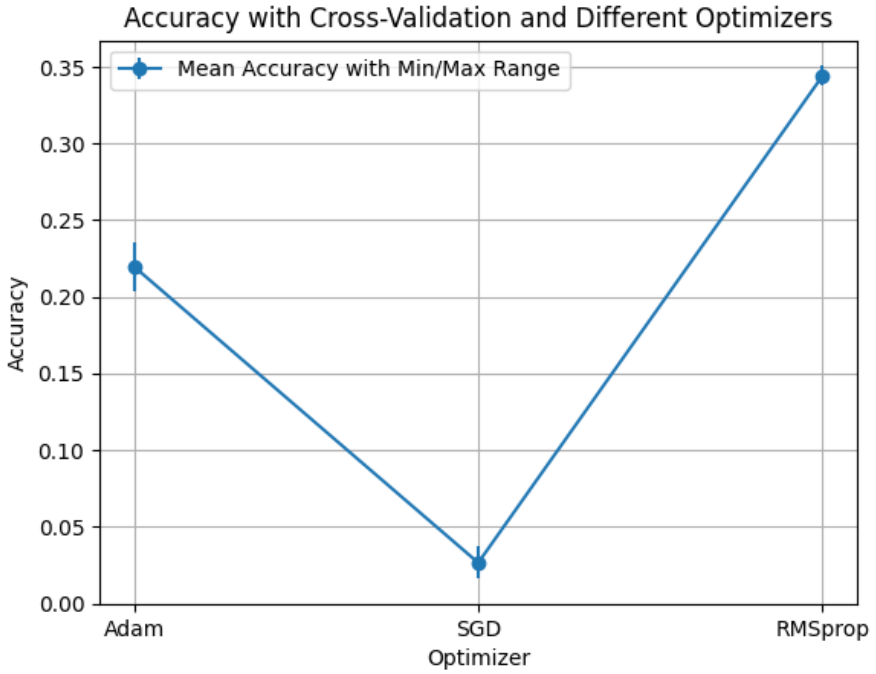

Figure 17. Performance of Normalized Laplacian with tanh activation with respect to different optimizers.

6.5. Summary of Findings

Experimental results showed that:

- For the incidence matrix, the **Hermitian Laplacian** consistently outperformed other variants, particularly when paired with ReLU and the Adam optimizer. The **Hermitian Laplacian with imaginary part** showed improved performance with ReLU and RMSprop. The **Random Walk Laplacian** demonstrated marginal improvements but was outperformed by the Hermitian variants. The **Normalized Laplacian** and **Standard Laplacian** performed well with RMSprop, especially when paired with tanh.

- For the adjacency matrix, the **Unnormalized Laplacian** showed competitive performance, with tanh and sigmoid activation functions achieving the highest accuracy. The **Normalized Laplacian** and **Random Walk Laplacian** performed best with ReLU and SGD. The **Hermitian Laplacian** for adjacency, incorporating a complex phase term, showed consistent but lower performance across configurations.
- Varying the value of q in the **Hermitian Laplacian** demonstrated significant improvements in performance across both the incidence and adjacency matrices. The optimal value of q led to better results for different variants, highlighting the importance of tuning this parameter to achieve the best model performance in different configurations.
- We observed sharp drops in accuracy when the mixing parameter q crossed a threshold. This behavior is explained by the spectrum of $\Theta_q = (1-q)I + qL$: as q increases, eigenvalues move from the identity regime toward those of L. If the dataset's structure (homophily level, hyperedge-size distribution, strength of directionality) is *misaligned* with L's diffusion profile, small changes in q can shift eigenvalues across the effective passband of the first-order filter, causing abrupt performance changes. In such cases, the complex Hermitian variant whose imaginary component induces oscillatory propagation can further amplify mismatch via phase cancellations, especially when directionality is weak or noisy, leading to reduced accuracy compared to the real operators.

7. Discussion

The results presented in Section 5 highlight the effectiveness of the proposed Directed Higher-Ordered Neural Network (HONN) across diverse benchmark datasets. HONN consistently outperformed or matched established baselines, including Graph Convolutional Networks (GCNs, Hypergraph Neural Networks (HGNNs), Directed Graph-based Semi-Supervised Learning (DGSSL) [17], and Gedi-hnn [13].

A central factor in HONN's success is its ability to encode both *directionality* and *higher-order relationships* through spectral Laplacians. By tuning the q-parameter, HONN balances local and global structural information, enabling adaptability across datasets of varying sparsity and density. For example, HONN achieved its highest accuracy on NTU-2012 (84%) with a global diffusion setting ($q \approx 0.9$), while smaller, sparser datasets such as Texas (87.4%) and Cornell (86.2%) benefited from more localized propagation ($q \approx 0.05$–0.1). This confirms that dataset-specific tuning of q significantly improves stability and generalization.

7.1. General Performance

HONN's performance across datasets demonstrates its adaptability to directed hypergraph structures. As shown in Table 3, HONN achieved state-of-the-art accuracy on multiple datasets, with notable strengths in its stability across random splits.

Our analysis of Laplacian variants revealed clear trends: the **Hermitian Laplacian** was most effective for small-scale hypergraphs such as WebKB Texas and Cornell, where it mitigated performance variance and preserved local structural signals. The **Normalized Laplacian** performed best for larger and denser datasets such as Citeseer and NTU-2012, where smoothing aided in capturing global dependencies. The **Complex Hermitian Laplacian** showed potential for tasks requiring phase-aware propagation, though its performance was more sensitive to optimizer choice. These findings reinforce that spectral operator selection directly affects HONN's learning efficiency.

While GCN surpassed HONN on Cora (81.5% vs. 71.5%), this result aligns with the dataset's inherently pairwise structure, which favors models based on standard graph Laplacians. HONN, in contrast, is designed for expressive, multi-relational domains, which explains its superior performance on hypergraph benchmarks like WebKB Texas and

Cornell. This suggests that HONN is particularly advantageous when higher-order and directional dependencies are prominent.

7.2. Stability and Generalization

One of HONN's strongest attributes is its stability across heterogeneous graph structures. Unlike conventional models that operate under fixed spectral assumptions, HONN leverages flexible Laplacian formulations, enabling adaptation to dataset-specific properties. This was evident in Citeseer, where HONN (59.4%) outperformed DGSSL and HGNN despite the dataset's noisy structure.

The ablation study on NTU-2012 further confirmed that activation choice interacts strongly with Laplacian selection: ReLU consistently yielded stable propagation with Hermitian operators, while tanh occasionally excelled with the Normalized Laplacian. These results highlight the importance of spectral–nonlinearity interplay in maintaining generalization.

Moreover, the q-sensitivity analysis established that low values favor localized learning on sparse datasets, while higher values are beneficial for dense relational structures. This adaptive mechanism provides HONN with a unique advantage in real-world applications where data connectivity is neither uniform nor fixed.

7.3. Comparison with Baseline Models

Compared with baseline models, HONN offers a clear advantage in capturing directed higher-order interactions. GCN remains strong on datasets with simple pairwise links but struggles with multi-entity relations that HONN naturally models via hyperedges. HGNN captures higher-order structures but lacks directionality, which limits its expressiveness. DGSSL incorporates direction but only in pairwise graphs, making it less effective for hypergraph tasks.

Our framework's superiority over Gedi-hnn is attributed to the flexibility of the tunable q-parameter and the adaptable spectral Laplacian formulation. While Gedi-hnn provides competitive accuracy, particularly on the Citeseer dataset, its fixed-Laplacian approach limits its capacity to optimally diffuse features across varied hypergraph topologies. Furthermore, while the HCHA variants (Convolutional and Attention) demonstrate a significant advantage in raw classification accuracy on standard benchmarks like Cora and Citeseer, HONN's consistent performance across diverse datasets underscores its superior generalization capability. The HCHA models' strong performance on standard citation graphs contrasts with HONN's dominance on the more specialized and challenging NTU-2012, Texas, and Cornell datasets.

Another baseline SigMaNet introduces a parameter-free *sign-magnetic Laplacian* L^{σ} that yields a Hermitian, positive–semidefinite operator for directed and/or signed graphs, enabling standard spectral filtering (e.g., Chebyshev/GCN-style) with per-layer cost comparable to GCN, $\mathcal{O}(|E|\, d + n\, d^2)$ [13]. As a graph method, SigMaNet models pairwise edges without clique expansion, but it does not natively capture higher-order interactions. In our study, it serves as a strong graph–spectral baseline on citation benchmarks, whereas HONN targets directed hypergraphs via incidence-based node→edge→node propagation and adaptive spectral mixing over multiple Laplacians $\{\Theta_N, \Theta_H, \Theta_H^c\}$ through a learnable q. This distinction-fixed single-operator on graphs versus adaptive multi-operator on higher-order structure explains why SigMaNet is competitive on homophilous pairwise datasets, while HONN provides consistent gains on datasets with pronounced directional, multi-way relations.

HONN bridges these gaps, achieving superior accuracy on NTU-2012, Texas, and Cornell, thereby validating the importance of integrating directionality into higher-order spectral learning.

These findings suggest that HONN is particularly suited to domains such as knowledge graphs, recommendation systems, and spatio-temporal networks, where interactions are inherently multi-relational and directional.

7.4. Limitations and Future Improvements

Despite its promising results, HONN has several limitations. First, the model introduces additional computational overhead compared to standard GCNs, due to the construction and manipulation of directed hypergraph Laplacians. While our experiments showed low per-epoch costs on moderate-scale datasets, scalability to web-scale graphs may require further optimization. Sparse approximations and distributed implementations are promising directions.

Second, HONN's performance is sensitive to hyperparameter selection, particularly q-values and Laplacian choice. Automated tuning strategies, such as Bayesian optimization or meta-learning, could reduce this reliance on manual search.

Third, while HONN excels in higher-order relational settings, it underperforms on simpler datasets like Citeseer. We explicitly acknowledge that on pairwise, homophilous citation graphs such as Cora and Citeseer, a graph–spectral baseline (e.g., GCN) can outperform HONN. Our framework is designed for settings where directionality and higher-order relations matter; in such cases (NTU-2012 and WebKB subsets Texas/Cornell), HONN consistently performs strongly in our experiments. This pattern aligns with HONN's operator design: when the underlying structure is effectively pairwise, graph-normalized spectra (Θ_N) are well matched; when relations are multi-way and directional, hypergraph operators (Θ_H or its complex variant) are more expressive. This points to the need for hybrid architectures that combine HONN's spectral strengths with classical GCN components, offering robustness in both low- and high-complexity datasets. We plan to expand the work towards the Open Graph Benchmark (OBGs) [41] in the future.

Finally, future work may explore architectural enhancements, such as attention-driven spectral modulation, contrastive spectral pretraining, integration of higher-ordered logic, and integration with spectral transformers, to further boost adaptability and efficiency. Improvements in handling highly sparse hypergraphs, where relationships are weakly connected, also remain a key challenge.

7.5. Application

Directed higher-order relations are ubiquitous beyond citation graphs, and HONN is designed for precisely these settings.

(i) *Bioinformatics and systems biology:* reaction pathways and metabolic networks couple *enzyme* $\rightarrow$ *substrate* $\rightarrow$ *cofactor*/*product* with intrinsic directionality; modeling hyperedges as reactions allows HONN to propagate along multi-molecular events without clique expansion.

(ii) *Recommender systems and session analytics:* user $\rightarrow$ session $\rightarrow$ item interactions are naturally many-to-one or one-to-many; directed hyperedges capture ordered co-occurrence and enable spectral diffusion that respects browsing direction.

(iii) *Fraud/finance risk:* entities $\rightarrow$ accounts $\rightarrow$ transactions form multi-actor, directed motifs; HONN can aggregate along these motifs to surface coordinated activity while controlling diffusion via q.

(iv) *Supply chains and logistics:* origin $\rightarrow$ route $\rightarrow$ destination flows are multi-way and directed; hypergraph propagation helps forecast disruptions across shared intermediates.

(v) *Program analysis and software dependency:* function → module → package graphs are higher-order and directional; HONN can diffuse signals through call/import hyperedges to prioritize fixes.
(vi) *Knowledge graphs and multimodal events:* events often bind multiple entities with roles (who-did-what-to-whom-where, time); directed hyperedges encode role-aware propagation.

HONN's layer cost remains sparse and linear in the incidence size ($\mathcal{O}((r+m)d)$), so training scales to large graphs; the mixing parameter $q \in [0, 1]$ offers a simple control to adapt the spectral regime to data constraints (local vs. global diffusion). Because Θ_q is 1-Lipschitz, gradients are stable, which eases production training. Finally, choosing among $L \in \{\Theta_N, \Theta_H, \Theta_H^c\}$ yields interpretable behavior: normalized diffusion for pairwise-like structure, real Hermitian for higher-order directionality, and complex Hermitian when phase (cyclic/oscillatory effects) is meaningful.

8. Conclusions

In this work, we introduced the Directed Higher-Ordered Neural Network (HONN), a novel framework that integrates directionality and higher-order spectral representations for hypergraph learning. Through extensive experiments on benchmark datasets, HONN consistently demonstrated superior generalization, stability, and adaptability compared to state-of-the-art baselines, achieving notable improvements on complex relational datasets such as NTU-2012, WebKB Texas, and Cornell. Our ablation studies confirmed the critical role of Laplacian selection, activation functions, and q-parameter tuning in shaping model performance, highlighting HONN's capacity to balance local and global structural information. While challenges remain in scaling to extremely large datasets and optimizing hyperparameter sensitivity, HONN establishes a robust foundation for advancing directed hypergraph learning. Future work will focus on developing scalable implementations and hybrid architectures that further exploit spectral properties, broadening HONN's applicability to domains such as knowledge graphs, spatio-temporal reasoning, and recommendation systems.

Author Contributions: Conceptualization, Y.M.O. and B.P.B.; methodology, Y.M.O. and B.P.B.; software, Y.M.O.; validation, Y.M.O., B.P.B. and R.F.; formal analysis, B.P.B.; investigation, G.I.; resources, A.R.-C.; data curation, Y.M.O.; writing—original draft preparation, Y.M.O. and B.P.B.; writing—review and editing, B.P.B., G.I. and A.R.-C.; visualization, Y.M.O.; supervision, B.P.B. and A.R.-C.; project administration, B.P.B. and A.R.-C.; funding acquisition, A.R.-C. and B.P.B. All authors have read and agreed to the published version of the manuscript.

Funding: This study is financed by the European Union—NextGenerationEU, through the National Recovery and Resilience Plan of the Republic of Bulgaria, project No. BG-RRP-2.013-0001.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets used in this study are publicly available in their respective repositories (Cora, Citeseer, NTU-2012, WebKB Texas, and Cornell). The code for directed hypergraph construction, model training, and evaluation is openly available at https://github.com/lisvresearch/directed-hypergraph (accessed on 7 October 2025).

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

Nomenclature (Symbols and Acronyms)

Item	Meaning	Type
Hypergraph objects and sets		
$G = (\mathcal{V}, \mathcal{E})$	Directed hypergraph	Symbol
$\mathcal{V}, \mathcal{E}$	Vertex set; hyperedge set	Symbol
$n := \lvert\mathcal{V}\rvert$, $m := \lvert\mathcal{E}\rvert$	Number of vertices; hyperedges	Symbol
$e_j \in \mathcal{E}, v_i \in \mathcal{V}$	A hyperedge; a vertex	Symbol
$T(e)$, $H(e)$	Tail and head vertex sets of hyperedge e	Symbol
Matrices, vectors, scalars		
$H \in \mathbb{R}^{n\times m}$	Incidence matrix	Symbol
H_{ij}	Incidence entry for v_i and e_j	Symbol
$W \in \mathbb{R}^{m\times m}$	Diagonal hyperedge weight matrix; $(W)_{jj} = W_{jj}$	Symbol
$D_v \in \mathbb{R}^{n\times n}$	Vertex degree matrix; $(D_v)_{ii} = \sum_j \lvert H_{ij}\rvert W_{jj}$	Symbol
$D_e \in \mathbb{R}^{m\times m}$	Hyperedge degree matrix; $(D_e)_{jj} = \sum_i \lvert H_{ij}\rvert$	Symbol
I	Identity matrix	Symbol
$d(v_i)$	Degree of vertex v_i (scalar)	Symbol
$w(e_j)$	Weight of hyperedge e_j (scalar; via W_{jj})	Symbol
$\mathbb{R}$	Real numbers (spaces for matrices/vectors)	Symbol
Operators and calculus		
$H^\top$	Transpose of H	Symbol
$\lvert\cdot\rvert$	Absolute value (used in degree definitions)	Symbol
$(\cdot)^{-1}$, $(\cdot)^{-1/2}$	Inverse; inverse square root (e.g., $D_v^{-1/2}$)	Symbol
arg max	Argmax operator (used in classifier definition)	Symbol
$\Re(\cdot)$, $\Im(\cdot)$	Real and imaginary parts (for Hermitian forms)	Symbol
Laplacians and spectral objects		
Θ_N	Normalized Laplacian: $I - D_v^{-1/2} H W D_e^{-1} H^\top D_v^{-1/2}$	Symbol
Θ_H	Hermitian Laplacian: $(1-q)I + q\, D_v^{-1/2} H W D_e^{-1} H^\top D_v^{-1/2}$	Symbol
Θ_H^c	Hermitian Laplacian with imaginary part: $\Theta_H + i\,\Im(\Theta_H)$	Symbol
$q \in [0,1]$	Mixing parameter in Θ_q (tunable)	Symbol
L	General Laplacian ($L \in \{\Theta_N, \Theta_H, \Theta_H^c\}$)	Symbol
Learning/model notation		
$\hat{y}_i$	Predicted class for node i	Symbol
P_{ik}	Softmax class probability for node i, class k	Symbol
dropout	Dropout regularization	Term
ReLU	Rectified Linear Unit activation	Acronym
SGD	Stochastic Gradient Descent	Acronym
Families of models		
GNN	Graph Neural Network	Acronym
GCN	Graph Convolutional Network	Acronym
HGNN	Hypergraph Neural Network	Acronym
DGNN	Directed Graph Neural Network	Acronym
DHGNN	Directed Hypergraph Neural Network	Acronym
HONN	Higher-Ordered Neural Network	Acronym
DGSSL	Directed Graph-based Semi-Supervised Learning	Acronym
HGCN	Hypergraph Convolutional Network	Acronym
HCHA	Hypergraph Convolutional Networks with Hyperedge Attention	Acronym
GeDi-HNN	Generalized Directed Hypergraph Neural Network	Acronym

References

1. Chami, I.; Abu-El-Haija, S.; Perozzi, B.; Ré, C.; Murphy, K. Machine learning on graphs: A model and comprehensive taxonomy. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 17009–17022.
2. Wu, Z.; Pan, S.; Chen, F.; Yu, P.S. A comprehensive survey on graph neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *32*, 4–24. [CrossRef]
3. Scarselli, F.; Gori, M.; Tsoi, A.C.; Hagenbuchner, M.; Monfardini, G. The graph neural network model. *IEEE Trans. Neural Netw.* **2009**, *20*, 61–80 . [CrossRef]
4. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv* **2016**, arXiv:1609.02907.
5. Shchur, O.; Mumme, M.; Bojchevski, A.; Günnemann, S. Pitfalls of graph neural network evaluation. *arXiv* **2018**, arXiv:1811.05868.
6. Pretolani, D. A directed hypergraph model for random time dependent shortest paths. *Discret. Appl. Math.* **2000**, *103*, 209–226. [CrossRef]
7. Thakur, M.; Tripathi, A. Linear connectivity problems in directed hypergraphs. *Theor. Comput. Sci.* **2009**, *410*, 4945–4957. [CrossRef]
8. Tran, H.Q.; Lin, Y. Directed hypergraph neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 17–24 June 2023.
9. Zhou, D.; Huang, J.; Schölkopf, B. Learning with hypergraphs: Clustering, classification, and embedding. *Adv. Neural Inf. Process. Syst.* **2007**, *19*, 1601–1608.
10. Bhuyan, B.P.; Ramdane-Cherif, A.; Tomar, R.; Singh, T.P. Neuro-symbolic artificial intelligence: A survey. *Neural Comput. Appl.* **2024**, *36*, 12809–12844. [CrossRef]
11. Bhuyan, B.P.; Ramdane-Cherif, A.; Singh, T.P.; Tomar, R. Studies in Computational Intelligence. In *Neuro-Symbolic Artificial Intelligence: Bridging Logic and Learning*; Springer Nature: Singapore, 2025; Volume 1176. [CrossRef]
12. Zhang, X.; Luo, T.; Ding, H.; Yang, J.; Cheng, J. MagNet: A neural network for directed graphs. *J. Mach. Learn. Res.* **2021**, *22*, 1–19.
13. Fiorini, S.; Patania, A.; Abualhaija, S. SigMaNet: One Laplacian to rule them all. *J. Artif. Intell. Res.* **2023** *72*, 1–30. [CrossRef]
14. Bai, S.; Wang, X.; Shi, C. Hypergraph convolution and hypergraph attention. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *33*, 1–13. [CrossRef]
15. Feng, Y.; Zhang, X.; Luo, T.; Zhang, Z.; Ji, R.; Gao, Y . Hypergraph neural networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019** , *45*, 3558–3565. [CrossRef]
16. Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; Bengio, Y. Graph attention networks. In Proceedings of the International Conference on Learning Representations (ICLR), Vancouver, BC, Canada, 30 April–3 May 2018.
17. Tran, L.H.; Tran, L.H. Directed hypergraph neural network. *J. Adv. Res. Dyn. Control Syst.* **2020**, *12*, 1434–1441. [CrossRef]
18. Fiorini, S.; Coniglio, S.; Ciavotta, M.; Del Bue, A. Let There be Direction in Hypergraph Neural Networks. *Trans. Mach. Learn. Res.* **2024** . Available online: https://openreview.net/forum?id=h48Ri6pmvi (accessed on 12 October 2025).
19. Sio-Kei, I.; Pearmain, A.J. Unequal error protection with the H.264 flexible macroblock ordering. *Vis. Commun. Image Process.* **2005**, *5960*, 596032.
20. Chan, K.; Ke, W.; Im, S. A General Method for Generating Discrete Orthogonal Matrices. *IEEE Access* **2021**, *9*, 120380–120391. [CrossRef]
21. Lin, X.; Zhang, W.; Shi, F.; Zhou, C.; Zou, L.; Zhao, X.; Yin, D.; Pan, S.; Cao, Y. Graph Neural Stochastic Diffusion for Estimating Uncertainty in Node Classification. In Proceedings of the 41st International Conference on Machine Learning (ICML), Vienna, Austria, 21–27 July 2024; Volume 235, pp. 30457–30478.
22. Yang, M.; Xu, X.-J. Recent Advances in Hypergraph Neural Networks. *arXiv* **2025**, arXiv:2503.07959. [CrossRef]
23. Ko, T.; Choi, Y.; Kim, C.-K. Universal Graph Contrastive Learning with a Novel Laplacian Perturbation. In Proceedings of the Thirty-Ninth Conference on Uncertainty in Artificial Intelligence (UAI), Pittsburgh, PA, USA, 31 July–4 August 2023; Volume 216, pp. 1098–1108.
24. Battiston, F.; Cencetti, G.; Iacopini, I.; Latora, V.; Lucas, M.; Patania, A.; Young, J.G.; Petri, G. Networks beyond pairwise interactions: Structure and dynamics. *Phys. Rep.* **2020**, *874*, 1–92. [CrossRef]
25. Berge, C. Hypergraphs: Combinatorics of finite sets. In *North-Holland Mathematical Library*; Elsevier: Amsterdam, The Netherlands, 1984.
26. Bretto, A. Mathematical Engineering. In *Hypergraph Theory: An Introduction*; Springer: Cham, Switzerland, 2013; Volume 1, pp. 209–216.
27. Bai, S.; Zhang, F.; Torr, P.H.S. Hypergraph convolution and hypergraph attention. *Pattern Recognit.* **2021**, *110*, 107637. [CrossRef]
28. Ausiello, G.; Laura, L. Directed hypergraphs: Introduction and fundamental algorithms—A survey. *Theor. Comput. Sci.* **2017**, *658*, 293–306. [CrossRef]
29. Frank, A.; Király, T.; Király, Z. On the orientation of graphs and hypergraphs. *Discret. Appl. Math.* **2003**, *131*, 385–400. [CrossRef]
30. Levie, R.; Huang, W.; Bucci, L.; Bronstein, M.; Kutyniok, G. Transferability of spectral graph convolutional neural networks. *J. Mach. Learn. Res.* **2021**, *22*, 1–59.

31. Fu, S.; Liu, W.; Zhou, Y.; Nie, L. HpLapGCN: Hypergraph p-Laplacian graph convolutional networks. *Neurocomputing* **2019**, *362*, 166–174. [CrossRef]
32. Gao, Y.; Feng, Y.; Ji, S.; Ji, R. HGNN+: General hypergraph neural networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *45*, 3181–3199. [CrossRef]
33. Handscomb, D.C.; Mason, J.C. *Chebyshev Polynomials*; Chapman and Hall/CRC: Boca Raton, FL, USA, 2002.
34. Singh, R.; Chen, Y. Signed graph neural networks: A frequency perspective. *arXiv* **2022**, arXiv:2208.07323. [CrossRef]
35. Chung, F.R.K. *Spectral Graph Theory*; American Mathematical Society: Providence, RI, USA, 1997; Volume 92.
36. Zhang, J.; Hui, B.; Harn, P.-W.; Sun, M.-T.; Ku, W.-S. smgc: A complex-valued graph convolutional network via magnetic Laplacian for directed graphs. *arXiv* **2021**, arXiv:2110.0757.
37. Smith, L.N. Super-convergence: Very fast training of neural networks using large learning rates. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Long Beach, CA, USA, 16–17 June 2019; pp. 128–136.
38. Goyal, P.; Ferrara, E.; Le H. Accurate learning rate scheduling for deep networks. In Proceedings of the IEEE International Conference on Machine Learning (ICML), Sydney, Australia, 6–11 August 2017.
39. Yan, Y.; Hashemi, M.; Swersky, K.; Yang, Y. Two sides of the same coin: Heterophily and oversmoothing in graph convolutional neural networks. *arXiv* **2010**, arXiv:2201.02260.
40. Ding, B.; Qian, H.; Zhou, J. Activation functions and their characteristics in deep neural networks. *J. Mach. Learn. Res.* **2023**, *24*, 112–129.
41. Hu, W.; Fey, M.; Zitnik, M.; Dong, Y.; Ren, H.; Liu, B.; Catasta, M.; Leskovec, J. Open Graph Benchmark: Datasets for Machine Learning on Graphs. *arXiv* **2020**, arXiv:2005.00687.