Beyond Local Neighborhoods: Fractional Diffusion with Lévy Flights on Simplicial Complexes for Link Prediction

Jaidev Goel
Department of Statistics
Virginia Tech
Blacksburg, VA, 24060
jaidev@vt.edu

Yuzhou Chen
Department of Statistics
University of California, Riverside
Riverside, CA 92521
yuzhou.chen@ucr.edu

Konstantin Avrachenkov
Neo Lab
INRIA, France
k.avrachenkov@inria.fr

Yulia R. Gel
Department of Statistics
Virginia Tech
Blacksburg, VA, 24060
ygl@vt.edu

Abstract

Owing to enhanced capabilities to capture a broad range of higher-order interactions, simplicial neural networks (SNNs) have emerged as a new powerful methodology for graph learning. However, prevailing SNNs are often limited in efficient exploration of heterogeneous graphs, thereby restricting the SNN utility on downstream tasks. To address this challenge, we introduce the concept of Lévy flights over simplices, offering a new efficient alternative to learning higher-order graph (sub)structures under heterogeneous scenarios. Specifically, we develop a new Fractional Hodge Laplacian Simplicial Neural Network (FHL-SNN), a novel approach leveraging fractional powers of the Hodge-Laplacian and allowing for a more efficient exploration of the underlying higher-order graph organization in conjunction with the link prediction. We establish a theoretical connection between a fractional diffusion process and graph exploration over simplices. In particular, we prove that the Lévy flight over simplices yields a lower mixing time than that of its non-fractional counterpart. We provide extensive experiments on both directed and undirected link prediction tasks with performance gains illuminating the power of non-local search of graph space via fractional dynamics. Finally, we show that the idea of Lévy flights over simplices is versatile and that the integration of fractional dynamics to existing SNNs has the potential to further boost the model performance.

1 Introduction

Simplicial Neural Networks (SNNs) have recently emerged as a powerful alternative to Graph Neural Networks (GNNs), offering a systematic solution for capturing higher-order interactions on graphs. By extending the concept of diffusion from node to node via edge to diffusion via higher order graph substructures which can be holistically described as simplices, SNNs leverage the notion of simplicial convolution, rooted in the discrete Hodge theory.

Theoretical and practical advantages of SNNs in graph learning have been actively explored in the last few years. For instance, [24, 8, 65, 50] investigate the impact of the embedding message passing

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

mechanisms within simplicial complexes, showing the potential to enhance the geometric invariance and expressivity. In turn, some of the most recent results offer insights into the SNN capabilities to tackle over-smoothing and over-squashing [58, 59, 25, 69]. SNNs have also shown superior results on such diverse graph learning tasks as link prediction, trajectory forecasting, and simplicial closure prediction [22, 55, 72, 3, 29, 47, 77, 25].

However, despite their premise, SNNs still struggle with efficiently capturing graph heterogeneity and long-range dependencies. This limitation becomes particularly apparent for heterogeneous networks in tasks which require understanding of distant but related topological structures, such as link prediction for formation of collaboration teams in citation networks, or tracking hierarchical hidden connectivity in criminal networks. This bottleneck is largely referred to as the "over-squashing" which occurs when large amounts of information must be compressed into fixed-dimensional node representations [21, 5, 63].

To address this fundamental challenge, we introduce the concept of Lévy flights over simplicial learning. In particular, our approach – Fractional Hodge Laplacian Simplicial Neural Network (FHL-SNN) – leverages fractional powers of the Hodge Laplacian to induce Lévy flights over simplicial complexes, enabling highly effective exploration of heterogenous graph structures. By combining the stable distribution characteristics of Lévy flights with the expressiveness of simplicial complexes, FHL-SNN achieves an important balance between local and non-local interactions, while maintaining computational efficiency. Through theoretical analysis and extensive experiments for directed and undirected link prediction tasks, we demonstrate that FHL-SNN not only sheds new light on simplicial graph exploration and the associated mixing time required but also provides a practical solution for capturing complex structural patterns in real-world networks, where both higher-order interactions and long-range dependencies play crucial roles. Finally, our experiments indicate that FHL is a versatile idea, and integration of fractional dynamics to existing SNNs has a potential to further boost the original model performance.

Our contributions can be summarized four-fold:

- To the best of our knowledge, we are the first to explore and experimentally validate the mathematical machinery of Lévy flights and the associated fractional Hodge Laplacians to simplicial learning on graphs, offering a principled approach to balance both local and non-local interactions in heterogenous topological structures.
- We derive theoretical guarantees on the convergence properties of fractional diffusion processes in higher-order topological learning, demonstrating better mixing time for Lévy flights over simplices than that of its non-fractional counterpart.
- We develop a novel neural architecture Fractional Hodge Laplacian Simplicial Neural Network (FHL-SNN) that efficiently combines fractional dynamics with hierarchical simplicial learning, enabling effective processing of both local and non-local dependencies.
- We illustrate the utility of FHL-SNN in application to both directed and undirected link prediction tasks on a wide range of benchmark datasets, including several heterogenous networks, criminal networks, and citation graphs. FHL-SNN demonstrates the superior performance compared to the state-of-the-art models which yields statistically significant gains.

2 Background and Related Work

Let $G = (V, E, \boldsymbol{X})$ be a connected graph, where V is the set of nodes ($|V| = n$) and $E \subseteq V \times V$ is the set of edges ($|E| = m$). Each node v_i is associated with a d-dimensional vector of attributes $\boldsymbol{x}_i$ which is the i-th row of matrix $\boldsymbol{X} \in \mathbb{R}^{n \times d}$ (here $d \geq 1$ is the input feature dimension). Connectivity of G can be encoded via adjacency matrix $\boldsymbol{A} \in \{0, 1\}^{n \times n}$ with entries $[\boldsymbol{A}]_{ij} = 1$ if nodes i and j are connected and 0, otherwise. (For an undirected graph G, $\boldsymbol{A} = \boldsymbol{A}^\top$.) The degree matrix $\boldsymbol{D} \in \mathbb{R}^{n \times n}$ is a diagonal matrix defined by $[\boldsymbol{D}]_{ii} = \sum_j [\boldsymbol{A}]_{ij}$. (See Appendix A for a list of notations.)

Hodge Theory and Graph Laplacians. Graph Laplacians serve as a cornerstone in both graph theory and graph machine learning, providing a crucial bridge between discrete structures (graphs) and continuous spaces (manifolds) [54]. The classical graph Laplacian, in its unnormalized form, is defined as $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A}$, where its symmetric and positive-semidefinite properties mirror those of the continuous Laplacian operator in vector calculus. This discrete Laplacian operator measures

local differences in graph structure, analogous to how the continuous Laplacian captures variations in differentiable functions. Specifically, just as the continuous Laplacian represents the divergence of a gradient flow, the graph Laplacian $\boldsymbol{L}$ quantifies how node properties differ from their local neighborhood averages, effectively characterizing diffusion processes across edges. (See Appendix E for a intuitive example on computation of Hodge Laplacian.)

However, many real-world phenomena require understanding beyond simple node-to-node relationships. For instance, schemes of financial fraud involve complex patterns of multiple interacting entities that cannot be adequately captured by pairwise relationships alone. This limitation motivates the extension to higher-order topological structures through the discrete Hodge theory [35, 42].

Definition 2.1 (Abstract Simplicial Complex)**.** A collection Δ of finite subsets of a node set V forms an abstract simplicial complex if it is closed under the subset operation: for every $\sigma \in \Delta$ and $\tau \subseteq \sigma$, we have $\tau \in \Delta$. A subset σ with $|\sigma| = N + 1$ is called a N-simplex, and its dimension is N. The faces of a simplex σ are its proper subsets, while co-faces are simplices that contain σ.

In this framework, nodes represent 0-simplices, edges are 1-simplices, triangles form 2-simplices, and so on. Higher-dimensional simplices ($N > 0$) carry orientation, defined by an ordering of their nodes modulo even permutations. For a N-simplex σ oriented as $[i_0, i_2, \ldots, i_N]$, each face inherits an induced orientation by omitting one node: $[i_0, i_1, \ldots, i_{j-1}, i_{j+1}, \ldots, i_N]$.

To study diffusion on these higher-order structures, we construct vector spaces C^N spanned by oriented N-simplices, whose elements are called N-chains. These chain spaces are connected through coboundary operators $\delta_N : C^N \to C^{N+1}$ and their adjoints δ_N^*, defined as:

$$\delta_N(f)([v_0, ..., v_{N+1}]) = \sum_{i=0}^{N+1} (-1)^i f([v_0, ..., \hat{v_i}, ..., v_{N+1}])$$

These operators naturally give rise to the d-Hodge Laplacian $\boldsymbol{L}_N : C^N \to C^N$, decomposing to:

$$\boldsymbol{L}_N = \underbrace{\boldsymbol{L}_N^{down}}_{\delta_{N-1}\delta_{N-1}^*} + \underbrace{\boldsymbol{L}_N^{up}}_{\delta_N^*\delta_N}.$$

Simplicial Neural Networks and Spectral Convolutions Building upon the Hodge Laplacian framework defined above, Simplicial Neural Networks (SNNs) generalize GNNs by enabling the processing of these higher-dimensional topological structures through spectral convolutions [22]. The spectral convolution on simplicial complexes is defined through the Fourier transform of N-chains:

$$\mathcal{F}_N : C^N \to \mathbb{R}^{|C^N|}, \quad \mathcal{F}_N(c) = (\langle c, e_1 \rangle_N, ..., \langle c, e_{|C^N|} \rangle_N),$$

where $\{e_i\}$ are eigenvectors of the Hodge Laplacian $\boldsymbol{L}_N$ ordered by eigenvalues $\lambda_1 \leq ... \leq \lambda_{|C^N|}$. This allows us to define convolutions between N-chains $c, c' \in C^N$ as:

$$c *_N c' = \mathcal{F}_N^{-1}(\mathcal{F}_N(c)\mathcal{F}_N(c')).$$

A simplicial convolutional layer then takes the form:

$$\text{SConv}_N(c) = \psi \circ (\mathcal{F}_N^{-1}(\varphi_W) *_N c),$$

where $\varphi_W = \sum_{i=0}^{N} W_i \Lambda^i$ is a learnable spectral filter represented as a polynomial in the eigenvalues of $\boldsymbol{L}_N$ and $\psi()$ is a non-linear activation function. This formulation naturally extends convolution operations to higher-dimensional graph structures, while preserving the topological information encoded in the simplicial complex. The spectral approach maintains computational efficiency and enables learning at multiple scales through the hierarchical nature of simplicial complexes. The resulting architecture effectively models higher-order interactions, making it particularly well suited for link prediction tasks in which relationships arise from complex multi-entity interactions rather than solely pairwise connections. Such representative applications include, for instance, gang formation analysis and drug–target association prediction.

Random Walk on Graphs and Simplicial Complexes Given an undirected graph $G = (V, E)$ with a vertex set V and an edge set E, the graph Laplacian $\boldsymbol{L}$ plays a fundamental role in understanding

random walks. The Laplacian can be defined as $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A}$. Alternatively, for any function $f : V \to \mathbb{R}$, we can define it combinatorially as $\boldsymbol{L}f(v) = \sum_{u \sim v}(f(v) - f(u))$, where $u \sim v$ denotes adjacent vertices. The connection to random walks emerges through the transition matrix $\boldsymbol{P} = \boldsymbol{D}^{-1}\boldsymbol{A}$, which relates to the Laplacian via $\boldsymbol{L} = \boldsymbol{D}(\boldsymbol{I} - \boldsymbol{P})$. For a continuous-time random walk starting at vertex v, the probability distribution p_t at time t evolves according to the differential equation $\frac{dp_t}{dt} = -\boldsymbol{L}p_t$, with solution $p_t = e^{-t\boldsymbol{L}}p_0$. The normalized Laplacian $\boldsymbol{\mathcal{L}} = \boldsymbol{D}^{-1/2}\boldsymbol{L}\boldsymbol{D}^{-1/2} = \boldsymbol{I} - \boldsymbol{D}^{-1/2}\boldsymbol{A}\boldsymbol{D}^{-1/2}$ provides a normalized version of this framework. The spectral properties of the Laplacian provide deep insights into random walk behavior. The eigenvalues $\lambda_1 \leq \lambda_2 \leq ... \leq \lambda_n$ of $\boldsymbol{L}$ determine crucial properties: the smallest eigenvalue is always 0 (corresponding to constant functions), while the second smallest eigenvalue, known as the algebraic connectivity, determines the *mixing rate*.

Random walks on graphs have been a fundamental tool in discrete mathematics, probability theory, and computer science, serving as an important tool for understanding both theoretical properties and practical applications in network analysis. While these walks effectively capture pairwise interactions and zero-dimensional homological features, many real-world networks have more complicated structure that cannot be adequately modeled by dyadic mechanisms alone. This limitation has motivated the extension of random walk theory to simplicial complexes, which provide a natural framework for studying higher-dimensional structural relationships.

The generalization of random walks to simplicial complexes has been an important and active research area [10, 51, 52, 57] introduces several mathematical challenges not present in the graph case. Most notably, for dimensions $N > 0$, one must consider oriented simplices, effectively doubling the state space of the Markov chain. Two primary approaches have emerged: up-random walks, which traverse between N-simplices through $(N + 1)$-simplices, and down-random walks, which move through $(N - 1)$-simplices. These walks are intimately connected to the discrete Hodge Laplacian operators $\boldsymbol{L}_N^{up} = \delta_N^*\delta_N$ and $\boldsymbol{L}_N^{down} = \delta_{N-1}\delta_{N-1}^*$, respectively, which encode crucial topological and geometric information about the complex.

A fundamental distinction from graph random walks lies in the analysis of convergence properties. While graph random walks converge to a unique stationary distribution on connected components, higher-dimensional walks require studying the limit of differences between probability distributions. This complexity arises from the orientation structure and manifests in the convergence rate expressions, which depend on spectral gaps of the respective Laplacians [51].Furthermore, the limiting behavior exhibits dependence on the initial state, a phenomenon not observed in the graph case.

The relationship between these walks and homology theory provides a powerful tool for understanding topological features of the underlying space. The spectrum of the Hodge Laplacian connects directly to Betti numbers and homology groups, while the convergence properties of the walks can detect non-trivial homological features. This connection has significant implications for applications in topological data analysis, network science, and machine learning, where understanding higher-order structure is crucial. However, challenges remain in characterizing the irreducibility conditions and developing efficient computational methods for analyzing these higher-dimensional processes.

3 Lévy Flights on Simplices

From Lévy Flights on Nodes to Lévy Flights on Higher Order Structures Random walks on graphs have been foundational for developing graph-based learning algorithms, but their effectiveness can be limited by local exploration patterns and oversampling of previously visited nodes. Lévy flights offer a compelling alternative by introducing scale-free, power-law distributed jumps characterized by

$$m_{u \to v}^{(\gamma)} = \delta_{uv} - (\boldsymbol{L}^{\gamma})_{uv}/k_u^{(\gamma)},$$

where γ controls the fractional power of the Laplacian matrix $\boldsymbol{L}$. The key advantage of Lévy flights lies in their lower average return probability ($p_0^{\gamma}(t) \sim t^{-1/2\gamma}$) compared to standard random walks (SRWs) ($p_0^1(t) \sim t^{-1/2}$), enabling more efficient graph exploration. This property makes them particularly effective for complex network structures, especially heterogeneous graphs where traditional random walks may struggle with local trapping and redundant sampling [16, 49]. Indeed, Lévy flights enable for long-range, non-local jumps to distant nodes, unlike SRWs focus on local connectivity. Intuitively, SRW is memoryless while Lévy flights integrate memory by defining steps as a weighted sum of past increments. This is essential when remote nodes are dependent, e.g.,



Figure 1: Random walks vs. Lévy flights. **Left**: Traditional random walk shows local exploration with small steps between neighboring nodes, leading to frequent revisits and slower exploration. **Right**: Lévy flight has long-range jumps and explores space more efficiently through power-law distributed step sizes, directly reaching distant nodes. Green & red dots are the start & end positions.

genetics or web networks, or various long memory processes (finance) [56]. Our γ here is closely interlinked with self-similarity and related to the Hurst exponent H which measures long memory strength in stochastic processes [48, 19]. Inspired by these ideas, we hypothesize that fractional diffusion is particularly well-suited for exploring higher-order (sub)structures in graphs – those characterized by long-range dependencies and complex topologies. This approach aims to reveal both local and distant latent multi-node interactions, which are particularly critical in applications such as fraudulent link prediction. As such, we introduce the concept of Lévy flights on simplices and discrete Fractional Hodge Laplacian.

Definition 3.1 (Discrete Fractional Hodge Laplacian)**.** *Fractional* Hodge N-Laplacian is given by:

$$\boldsymbol{L}_N^{\gamma} = \boldsymbol{V}\boldsymbol{\Lambda}^{\gamma}\boldsymbol{V}^T, \quad 0 < \gamma \leq 1, \tag{1}$$

where $\boldsymbol{L}_N = \boldsymbol{L}_N^{up} + \boldsymbol{L}_N^{down}$. (See Appx E and Appx G for more details on numerical methods for fractional powers of matrices.)

Since the eigenvalues of the fractional Hodge Laplacian are non-negative, we can introduce the following stable differential evolution:

$$\dot{x} = -\boldsymbol{L}x, \quad \text{or} \quad x(t) = \exp(-t\boldsymbol{L})x_0. \tag{2}$$

Remark The elements of $\delta_N^T\delta_N$ and $\delta_{N-1}\delta_{N-1}^T$ involve sums of ± 1 generated from oriented incidences, and hence, the off-diagonal elements of $\boldsymbol{L}_N$ may not necessarily be interpreted as transition rates for some diffusion process. Nevertheless, the process (2) is still well-defined and stable.

Fractional Random Walks on Simplices There are some important *non-trivial* cases when we can still interpret Eq. (2) as diffusion process. (The trivial case of $N = 0$ corresponds to standard graphs). In particular, we can consider an oriented simplex with maximal degree N, where the degree of all $(N-1)$-simplexes is 2 [51, 23]. The N-th down-Laplacian is then

$$\boldsymbol{L}_N^{down} f[\sigma] = \frac{(N+1)f([\sigma])}{2} - \frac{1}{2}\sum_{\sigma',\ \rho\in\sigma\cap\sigma'} f([\sigma']). \tag{3}$$

Now consider

$$\tilde{\boldsymbol{P}} = \boldsymbol{I} - \frac{2(1-p)}{N+1}(\boldsymbol{L}_N^{down})^{\gamma}, \quad 0 < \gamma \leq 1, \quad 0 < p < 1, \tag{4}$$

where p is an arbitrary parameter, governing the laziness of the walk (i.e., probability of not changing states). We claim that $\tilde{P}$ is a stochastic matrix defining transition probabilities for a fractional random walk over simplices. Towards this goal, we first establish an important general result on characterization of singular irreducible M-matrices, which is of interest on its own.

Lemma 3.2 (Characterization of Singular Irreducible M-matrices)**.** *Let A be a singular irreducible M-matrix and $s > 0$. Then $A(sI + A)^{-1}$ is a singular M-matrix. Moreover, $A(sI + A)^{-1}$ has "property c", i.e., A can be represented as $A = sI - B$, where $s > 0$, $B \geq 0$ and matrix $T = B/s$ is semi-convergent.*

Proof: First, note that

$$A(sI + A)^{-1} = I - s(sI + A)^{-1}.$$

Since A is a singular M-matrix, $sI + A$ is a nonsingular M-matrix. Hence, from N38 of [4], $s(sI + A)^{-1} \geq 0$. Furthermore, since A is irreducible, $sI + A$ is also irreducible. Hence, off-diagonal elements of matrix $I - s(sI + A)^{-1}$ are non-positive, which implies that

$$A(sI + A)^{-1} = I - s(sI + A)^{-1}$$

is a Z-matrix.

Second, in view of Theorem 4.16 of [4], we know that since A is a singular irreducible M-matrix, there exists a vector v, $v \gg 0$, such that $Av = 0$. Notice that $(sI + A)v = sv + Av = sv$, so

$$(sI + A)^{-1}v = \frac{1}{s}v$$

and

$$A(sI + A)^{-1}v = \frac{1}{s}Av = 0.$$

Hence, v is also an eigenvector of $A(sI + A)^{-1}$, corresponding to its 0 eigenvalue.

Hence, we conclude that $A(sI + A)^{-1}$ is a Z-matrix with a positive vector in its kernel. Then, by standard characterization of Z-matrices (see Exercise 6.4.14, page 155 of [4]) we find that $A(sI + A)^{-1}$ is a singular M-matrix. Furthermore, in view of Theorem 4.16 by [4], this implies that $A(sI + A)^{-1}$ has "property c".

□

Note that in numerical analysis, "property c" is essential for guaranteeing convergence of numerical algorithms and ensuring that the solutions are non-negative [4, 36, 53].

Armed with Lemma 3.2, we now show that fractional Laplacian $(\boldsymbol{L}_N^{down})^\gamma$ is an M-matrix.

Lemma 3.3 (Fractional Hodge Laplacian is M-matrix)**.** *The operator $(\boldsymbol{L}_N^{down})^\gamma$ is an M-matrix.*

Proof: Consider the Balakrishnan formula for the fractional Laplacian [2]:

$$(\boldsymbol{L}_N^{down})^\gamma = \frac{\sin(\pi\gamma)}{\pi} \int_0^\infty L_N^{down}[s\boldsymbol{I} + \boldsymbol{L}_N^{down}]^{-1} s^{\gamma-1} ds. \tag{5}$$

Note that $\boldsymbol{L}_N^{down}$ is an irreducible M-matrix [23, 23] and $s > 0$. Hence, by Lemma 3.2, $\boldsymbol{L}_N^{down}[s\boldsymbol{I} + \boldsymbol{L}_N^{down}]^{-1}$ is an M-matrix. The integration with the positive kernel $s^{\gamma-1}$ preserves the positivity/negativity of the elements, concluding the proof. □

We are now ready to define the fractional random walk over simplices and its transition probabilities.

Theorem 3.4 ($\tilde{P}$ is Stochastic Matrix)**.** *Let $\frac{1}{2} \leq p < 1$ and*

$$\tilde{\boldsymbol{P}} = \boldsymbol{I} - \frac{2(1-p)}{N+1}(\boldsymbol{L}_N^{down})^\gamma, \quad 0 < \gamma \leq 1.$$

Then, $\tilde{P}$ is stochastic matrix defining the transition probabilities of the random walk on simplices.

Proof: First, we note that by Lemma 3.3 the fractional Laplacian $(\boldsymbol{L}_N^{down})^\gamma$ is an M-matrix.

Next, let us show that

$$|[(\boldsymbol{L}_N^{down})^\gamma]_{ij}| \leq N + 1. \tag{6}$$

For the diagonal elements, we note that due to semi-definiteness we have

$$|[(\boldsymbol{L}_N^{down})^\gamma]_{ii}| = |e_i^T(\boldsymbol{L}_N^{down})^\gamma e_i| \leq \lambda_{max}^\gamma ||e_i||_2^2 = \lambda_{max}^\gamma. \tag{7}$$

Then, for the off-diagonal elements, we can write

$$|[(\boldsymbol{L}_N^{down})^\gamma]_{ij}| \leq \sum_{l=2}^{n} \lambda_l^\gamma |v_{il}||v_{lj}| \leq \frac{1}{2}\sum_{l=2}^{n} \lambda_l^\gamma ((v_{il})^2 + (v_{lj})^2)$$
$$= \frac{1}{2}([(\boldsymbol{L}_N^{down})^\gamma]_{ii} + [(\boldsymbol{L}_N^{down})^\gamma]_{jj}) \leq \lambda_{max}^\gamma.$$

Since $\lambda_{max} \leq N+1$, the bound (6) follows.

Since $\boldsymbol{L}_N^{down}$ is a Laplacian, $\boldsymbol{L}_N^{down}\mathbb{1} = 0$. The semigroup definition of fractional Laplacian via the Balakrishnan formula (5) implies that $ker(\boldsymbol{L}_N^{down})$ and $ker((\boldsymbol{L}_N^{down})^\gamma)$ coincide, and hence, $(\boldsymbol{L}_N^{down})^\gamma\mathbb{1} = 0$. This implies that $\tilde{\boldsymbol{P}}\mathbb{1} = \mathbb{1}$ and row-sum of $\tilde{\boldsymbol{P}}$ equals to 1.

Finally, since

$$|[(\boldsymbol{L}_N^{down})^\gamma]_{ii}| < N+1,$$

to ensure $\tilde{p}_{ii} \geq 0$, it suffices to have

$$1 - \frac{2(1-p)}{N+1}(N+1) = 2p - 1 \geq 0,$$

which holds for $p \geq 1/2$. In turn, $\tilde{p}_{ij} \geq 0$ follows from the fact that $(\boldsymbol{L}_N^{down})^\gamma$ is an M-matrix.

Hence, we obtain $0 \leq \tilde{p}_{ij} \leq 1$, and consequently, $\tilde{\boldsymbol{P}}$ is a stochastic matrix. □

Now, let us show that Lévy flights on simplices reduce mixing time in comparison with the traditional non-fractional random walk on simplices.

Definition 3.5 (Mixing Time)**.** Let π_t be the marginal distribution of the random walk and at time step t and be π_∞ the corresponding limiting distribution. Then $t_{\text{mix}}^\epsilon = \min\{t : \|\pi_t - \pi_\infty\|_{TV} \leq \epsilon\}$ is the corresponding mixing time.

We now turn to establishing the bounds for time associated with exploration of higher-order graph (sub)structures using Lévy flights over simplicial complexes and its standard non-fractional counterpart.

Theorem 3.6 (Mixing Time Bound and Speedup)**.** *Let conditions of Theorem 3.4 hold and let $\lambda_2 < 1$, where λ_2 is the smallest non-zero eigenvalue of L_N^{down}. Let $p \geq 1/2$ and $\varrho = 2(1-p)/(N+1)$. Then, the mixing time of the Lévy flights over simplices with the transition probability matrix $\tilde{P}$ (4) and $0 < \gamma \leq 1$ is*

$$\tau_{\text{mix}}^\epsilon = \mathcal{O}\bigg(\frac{1}{\varrho\lambda_2^\gamma}\log\bigg[\frac{C}{\epsilon}\bigg]\bigg)$$

and the mixing time of its non-fractional counterpart is

$$t_{\text{mix}}^\epsilon = \mathcal{O}\bigg(\frac{1}{\varrho\lambda_2}\log\bigg[\frac{C}{\epsilon}\bigg]\bigg),$$

implying that the speedup factor due to to fractional exploration is $\mathcal{O}\big(\lambda_2^{-(1-\gamma)}\big)$.

Proof is in Appendix C.

Theorem 3.6 provides a theoretical guarantee that the mixing time associated with exploration of higher-order graph (sub)structures using Lévy flights on simplices is reduced in comparison with its non-fractional counterpart. In particular, the gains become more pronounced for sparser heterogeneous graphs with poorly connected complexes, i.e. $\lambda_2 \ll 1$ implies higher speedup due to fractional exploration. A bound on the mixing time can be further refined using the results of [1] on the upper bound of the second eigenvalue, which we leave for future standalone study.

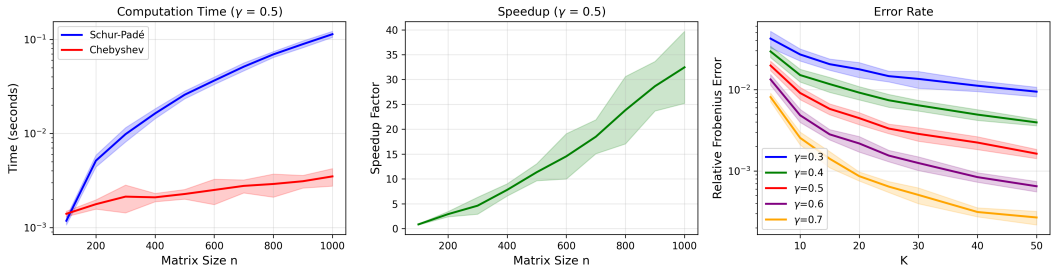


Figure 2: Chebyshev approximation of $(\mathbf{L}_N^{\mathrm{down}})^\gamma$. **Left:** Schur–Padé vs. Chebyshev runtime at $\gamma = 0.5$. **Center:** speedup factor. **Right:** relative Frobenius error vs. degree K for $\gamma \in \{0.3, 0.4, 0.5, 0.6, 0.7\}$.

4 Approximation of Fractional Powers of Hodge Laplacian by the Chebyshev Polynomials

The mixing-time guarantees of Section 3 concern the exact fractional Hodge Laplacian $(\mathbf{L}_N^{\mathrm{down}})^\gamma$, whose direct computation by the Schur–Padé algorithm [32] incurs $\mathcal{O}(n^3)$ cost — prohibitive for the simplicial complexes of practical interest. We approximate $(\mathbf{L}_N^{\mathrm{down}})^\gamma$ by a degree-K Chebyshev polynomial in $\mathbf{L}_N^{\mathrm{down}}$, exploiting the geometric error decay of Chebyshev expansions for functions analytic in a neighborhood of the spectrum [62]; the branch-point singularity of $t \mapsto t^\gamma$ at the origin governs the constant. Algorithm 1 uses only sparse matrix–vector products via the three-term recurrence $T_{k+1}(x) = 2xT_k(x) - T_{k-1}(x)$, yielding $\mathcal{O}(Km)$ complexity for a Hodge Laplacian with m non-zero entries.

Algorithm 1 Chebyshev Approximation for $\mathbf{L}_N^\gamma c$

Require: Hodge Laplacian $\mathbf{L}_N$ with m non-zero entries; fractional power $\gamma \in (0, 1]$; polynomial degree K; signal c; largest eigenvalue $\lambda_{\max}$ of $\mathbf{L}_N$ (obtained e.g. via Lanczos in $\mathcal{O}(m)$ per iteration)

Ensure: $\hat{y} \approx \mathbf{L}_N^\gamma c$

$\tilde{\mathbf{L}}_N \leftarrow \frac{2}{\lambda_{\max}}\mathbf{L}_N - \mathbf{I}$ ▷ rescale spectrum to $[-1, 1]$

$x_\ell \leftarrow \cos\left(\frac{\pi(\ell+0.5)}{K}\right)$ for $\ell = 0, \ldots, K-1$ ▷ Chebyshev nodes

for $k = 0$ to $K-1$ **do**

  $c_k \leftarrow \frac{2}{K}\sum_{\ell=0}^{K-1}\left(\frac{(x_\ell+1)\lambda_{\max}}{2}\right)^\gamma \cos\left(\frac{\pi k(\ell+0.5)}{K}\right)$

end for

$T_0 \leftarrow c$; $T_1 \leftarrow \tilde{\mathbf{L}}_N c$; $\hat{y} \leftarrow \frac{c_0}{2}T_0 + c_1 T_1$

for $k = 2$ to $K-1$ **do**

  $T_{\text{new}} \leftarrow 2\tilde{\mathbf{L}}_N T_1 - T_0$

  $\hat{y} \leftarrow \hat{y} + c_k T_{\text{new}}$

  $T_0 \leftarrow T_1$; $T_1 \leftarrow T_{\text{new}}$

end for

return $\hat{y}$

Figure 2 reports a 35× speedup over Schur–Padé at $n = 1000$ with sub-percent relative Frobenius error at $K = 50$ across $\gamma \in [0.3, 0.7]$. The error curves exhibit the geometric decay in K predicted by Bernstein-ellipse theory, with the constant deteriorating for smaller γ as the branch-point singularity tightens. We use Algorithm 1 as the computational primitive in Section 5; an approximation-error theorem is discussed further in Appendix G

5 From Operator to Architecture: FHL-SNN

Theorem 3.6 establishes that the fractional Hodge Laplacian $(\mathbf{L}_N^{\mathrm{down}})^\gamma$ induces a random walk on simplicial complexes with reduced mixing time relative to its non-fractional counterpart. We

instantiate this operator within a simplicial neural network, FHL-SNN, to examine whether the predicted phenomena leave measurable signatures on a downstream learning task with a neural architecture. FHL-SNN replaces the standard Hodge Laplacian in the block-simplicial message-passing scheme of [17] with $(\mathbf{L}_N^{\mathrm{down}})^\gamma$, computed via Algorithm 1 to keep per-layer cost linear in the non-zero entries. Remaining architectural details we defer to Appendix B. Across link-prediction benchmarks spanning a wide range of clustering coefficients (Appendix D), the within-architecture gap between FHL-SNN and its non-fractional baseline tracks higher-order structural richness: the gap is largest on complexes rich in triangles and higher-order cliques (e.g., Meetings, PCQM-Contact) and vanishes on complexes where higher-order structure is degenerate (e.g., Disease, with zero triangles). The same pattern persists when the fractional operator is substituted into other simplicial backbones (SNN, SCNN, SAT, HiGCN), with the marginal gain shrinking for architectures that already encode multi-scale structure, consistent with the spectral interpretation that the effect is mediated by the operator rather than by particular architectural choices. Both observations match the regime-dependence predicted by Theorem 3.6: the fractional speedup is informative precisely where higher-order structure exists to be exploited.

Bridging mixing time and learning performance. We briefly discuss the mixing-time guarantee of Theorem 3.6 which concerns a Markov chain on simplicial chains, and gains we observe in Appendix D. The bridge is supplied by two established results from the oversquashing literature. For $\lambda_2 < 1$ and $0 < \gamma \leq 1$, Theorem 3.6 implies $\lambda_2^\gamma \geq \lambda_2$, so the fractional operator widens the spectral gap on the chain space. [7] show that effective resistance R_{uv} is bounded above by a quantity inversely proportional to the spectral gap, and [18] show that the sensitivity of a K-layer message-passing network at u to a perturbation at v is controlled by R_{uv}. Composing these, a wider spectral gap contracts R_{uv} on the chain space, which in turn relaxes the oversquashing bottleneck. For instance, on Cora, the gap widens monotonically as γ decreases ($\lambda_2 = 0.3011$ at $\gamma = 1$ versus $\lambda_2^\gamma = 0.3395, 0.432, 0.549, 0.697$ at $\gamma = 0.9, 0.7, 0.5, 0.3$), and relative gains are attained on PCQM-Contact from the Long Range Benchmark, where oversquashing is the dominant failure mode; conversely, the gap over the non-fractional baseline vanishes on Disease, whose complex contains no triangles and where there is no higher-order resistance to contract. Both patterns match the regime-dependence anticipated by Theorem 3.6.

Choice of γ. The exponent γ controls the strength of long-memory in the walk: a standard random walk is memoryless, while a fractional walk expresses each step as a weighted sum of past increments, the discrete analogue of fractional ARIMA and fractional Brownian motion with γ tied to the Hurst exponent [48, 19]. Both endpoints of $(0, 1]$ are degenerate, at $\gamma = 1$ the walk reduces to the standard walk, while as $\gamma \to 0$ the operator approaches the projector onto $(\ker \mathbf{L}_N^{\mathrm{down}})^\perp$, so the informative regime is strictly interior. Within it, Theorem 3.6 predicts the speedup factor $\lambda_2^{-(1-\gamma)}$ grows as γ decreases and λ_2 shrinks, matching the classical principle that end points for fractional parameter are optimal when the relevant structure is sparse and distant [56]. We select γ by grid search; across all datasets the empirical optimum falls in $[0.3, 0.7]$, consistent with this interior prediction.

6 Conclusion and Future Directions

We have introduced a new concept, higher-order topological learning through Lévy flights over simplices, enabling us to capture non-local and complex dependencies in graph learning. Our FHL-SNN approach, operating through fractional powers of the Hodge Laplacian provides a mathematically principled way to balance local and non-local interactions in graph (sub)structures. The experiments have shown that FHL-SNN delivers significant improvements in link prediction across multiple benchmarks, particularly under scenarios involving sparse and heterogeneous graphs. Our key theoretical contribution lies in establishing the connection between fractional diffusion and higher-order topological learning, providing better mixing times for Lévy flights over simplices. This in turn has a potential to improve the long range dependence reasoning and opens new directions for research in topological learning. In the future, we plan to investigate the expressiveness of FHL-SNN, which necessitates a more involved machinery of expressiveness of SNN and GNNs for weighted graphs, an area that remains substantially less investigated [41, 73]. We hypothesize that one approach for understanding expressiveness of fractional simplicial diffusion is via the limiting graphon representation [9]. We also conjecture that the expressiveness results of [44] can be adapted to the fractional case. Finally, another interesting future direction is to address a special case of mixing

time for fractional random walks on expander graphs, which can be tackled using the theoretical argument of [1].

7 Impact Statements

This paper presents work whose goal is to advance the field of machine learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

Acknowledgement

This work has been supported in part by NSF DMS-2533984/2533985 and OAC-2530469/253047, by the U.S. Department of Energy (DOE), Office of Science, Advanced Scientific Computing Research (ASCR) program under the Scientific Discovery through Advanced Computing (SciDAC) Institute "LEADS: LEarning-Accelerated Domain Science". This research was conducted in part while Y.R.G. visited the Okinawa Institute of Science and Technology (OIST) through the Theoretical Sciences Visiting Program (TSVP). The authors are also very grateful to the Area Chair and the three reviewers for their constructive feedback, their invaluable suggestions for improving the clarity of the paper and future research directions, and also for their exemplary patience.

References

[1] V. L. Alev and L. C. Lau. Improved analysis of higher order random walks and applications. In *Proc. of STOC*, pages 1198–1211, 2020.

[2] A. Balakrishnan. Fractional powers of closed operators and the semigroups generated by them. *Pacific J. Math.*, 10(4):419–437, 1960.

[3] C. Battiloro, L. Testa, L. Giusti, S. Sardellitti, P. Di Lorenzo, and S. Barbarossa. Generalized simplicial attention neural networks. *IEEE Transactions on Signal and Information Processing over Networks*, 2024.

[4] A. Berman and R. J. Plemmons. *Nonnegative Matrices in the Mathematical Sciences*. SIAM, 1994.

[5] M. Besta, F. Scheidl, L. Gianinazzi, G. Kwasniewski, S. Klaiman, J. Müller, and T. Hoefler. Demystifying higher-order graph neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

[6] G. Bianconi. The topological dirac equation of networks and simplicial complexes. *Journal of Physics: Complexity*, 2(3):035022, 2021.

[7] M. Black, Z. Wan, A. Nayyeri, and Y. Wang. Understanding oversquashing in GNNs through the lens of effective resistance. In *Proc. of ICML*, 2023.

[8] C. Bodnar, F. Frasca, Y. Wang, N. Otter, G. F. Montufar, P. Lio, and M. Bronstein. Weisfeiler and Lehman go topological: Message passing simplicial networks. In *Proc. of ICML*, 2021.

[9] J. Böker, R. Levie, N. Huang, S. Villar, and C. Morris. Fine-grained expressivity of graph neural networks. In *Proc. of NeurIPS*, 2023.

[10] T. Bonis, L. Decreusefond, V. C. Tran, and Z. I. Zhang. Random walks on simplicial complexes. *arXiv preprint arXiv:2404.08803*, 2024.

[11] X. Bresson and T. Laurent. Residual gated graph convnets. *arXiv preprint arXiv:1711.07553*, 2017.

[12] L. Calmon, J. G. Restrepo, J. J. Torres, and G. Bianconi. Dirac synchronization is rhythmic and explosive. *Communications Physics*, 5(1):253, 2022.

[13] L. Calmon, M. T. Schaub, and G. Bianconi. Higher-order signal processing with the Dirac operator. In *Proc. of ACSSC*, 2022.

[14] I. Chami, Z. Ying, C. Ré, and J. Leskovec. Hyperbolic graph convolutional neural networks. In *Proc. of NeurIPS*, 2019.

[15] M. Chen, Z. Wei, Z. Huang, B. Ding, and Y. Li. Simple and deep graph convolutional networks. In *Proc. of ICML*, 2020.

[16] Y. Chen, Y. R. Gel, and K. Avrachenkov. LFGCN: Levitating over graphs with Levy flights. In *Proc. of ICDM*, 2020.

[17] Y. Chen, Y. R. Gel, and H. V. Poor. BScNets: Block simplicial complex neural networks. In *Proc. of AAAI*, 2022.

[18] F. Di Giovanni, T. K. Rusch, M. M. Bronstein, A. Deac, M. Lackenby, S. Mishra, and P. Veličković. How does over-squashing affect the power of GNNs? *TMLR*, 2024.

[19] P. Doukhan, G. Oppenheim, and M. Taqqu. *Theory and applications of long-range dependence*. Springer Science & Business Media, 2002.

[20] V. P. Dwivedi and X. Bresson. A generalization of transformer networks to graphs. In *Proc. of DLG-AAAI*, 2021.

[21] V. P. Dwivedi, L. Rampášek, M. Galkin, A. Parviz, G. Wolf, A. T. Luu, and D. Beaini. Long range graph benchmark. In *Proc. of NeurIPS*, 2022.

[22] S. Ebli, M. Defferrard, and G. Spreemann. Simplicial neural networks. In *Proc. of NeurIPS TDA Beyond 2020 Workshop*, 2020.

[23] M. Eidi and S. Mukherjee. Irreducibility of Markov chains on simplicial complexes, the spectrum of the discrete Hodge Laplacian and homology. *arXiv preprint arXiv:2310.07912*, 2023.

[24] F. Eijkelboom, R. Hesselink, and E. J. Bekkers. E (n) equivariant message passing simplicial networks. In *Proc. of ICML*, 2023.

[25] A. Einizade, D. Thanou, F. D. Malliaros, and J. H. Giraldo. Continuous Simplicial Neural Networks. In *Proc. of NeurIPS*, 2025.

[26] O. Ganea, G. Bécigneul, and T. Hofmann. Hyperbolic neural networks. In *Proc. of NeurIPS*, 2018.

[27] Y. Gao, Y. Feng, S. Ji, and R. Ji. HGNN+: General hypergraph neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3):3181–3199, 2022.

[28] C. W. J. Goh, C. Bodnar, and P. Lio. Simplicial attention networks. In *Proc. of ICLR GTRL*, 2022.

[29] S. Gurugubelli and S. P. Chepuri. Sann: Simple yet powerful simplicial-aware neural networks. In *Proc. of ICLR*, 2024.

[30] W. Hamilton, Z. Ying, and J. Leskovec. Inductive representation learning on large graphs. In *Proc. of NeurIPS*, 2017.

[31] N. J. Higham. *Functions of matrices: theory and computation*. SIAM, 2008.

[32] N. J. Higham and L. Lin. A schur–padé algorithm for fractional powers of a matrix. *SIAM Journal on Matrix Analysis and Applications*, 32(3):1056–1078, 2011.

[33] W. Hu, M. Fey, M. Zitnik, Y. Dong, H. Ren, B. Liu, M. Catasta, and J. Leskovec. Open graph benchmark: Datasets for machine learning on graphs. In *Proc. of NeurIPS*, 2020.

[34] Y. Huang, Y. Zeng, Q. Wu, and L. Lü. Higher-order graph convolutional network with flower-petals Laplacians on simplicial complexes. In *Proc. of AAAI*, 2024.

[35] J. L. Johnson and T. Goldring. Discrete Hodge theory on graphs: a tutorial. *Computing in Science & Engineering*, 15(5):42–55, 2012.

[36] A. Kalauch, S. Lavanya, and K. Sivakumar. Singular irreducible M-matrices revisited. *Linear Algebra and its Applications*, 565:47–64, 2019.

[37] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. In *Proc of ICLR*, 2017.

[38] G. Kollias, V. Kalantzis, T. Idé, A. Lozano, and N. Abe. Directed graph auto-encoders. In *Proc. of AAAI*, 2022.

[39] D. Kreuzer, D. Beaini, W. Hamilton, V. Létourneau, and P. Tossou. Rethinking graph transformers with spectral attention. In *Proc. of NeurIPS*, 2021.

[40] J. Leskovec and C. Faloutsos. Sampling from large graphs. In *Proc. of ACM SIGKDD*, 2006.

[41] P. Li and J. Leskovec. The expressive power of graph neural networks. *Graph Neural Networks: Foundations, Frontiers, and Applications*, pages 63–98, 2022.

[42] L.-H. Lim. Hodge Laplacians on graphs. *SIAM Review*, 62(3):685–715, 2020.

[43] S. Lloyd, S. Garnerone, and P. Zanardi. Quantum algorithms for topological and geometric analysis of data. *Nature communications*, 7(1):10138, 2016.

[44] C. López Amado, T. Schwarz, Y. Tian, and R. Lambiotte. Complex-weighted convolutional networks: provable expressiveness via complex diffusion. In *Proc. of LoG*, 2025.

[45] W. Ma, Y. Wang, X. Wang, and M. Zhang. Reconsidering the Performance of GAE in Link Prediction. *Proc. of CIKM*, 2025.

[46] Z. Ma, W. Zhao, and Z. Yang. Directed hypergraph representation learning for link prediction. In *Proc. of AISTATS*, 2024.

[47] H. Madhu, S. Gurugubelli, and S. P. Chepuri. Unsupervised parameter-free simplicial representation learning with scattering transforms. In *Proc. of ICML*, 2024.

[48] B. Mandelbrot. Long-run linearity, locally gaussian process, h-spectra and infinite variances. *International Economic Review*, 10(1):82–111, 1969.

[49] S. Maskey, R. Paolino, A. Bacho, and G. Kutyniok. A fractional graph Laplacian approach to oversmoothing. In *Proc. of NeurIPS*, 2024.

[50] C. Morris, G. Rattan, and P. Mutzel. Weisfeiler and Leman go sparse: Towards scalable higher-order graph embeddings. In *Proc. of NeurIPS*, 2020.

[51] S. Mukherjee and J. Steenbergen. Random walks on simplicial complexes and harmonics. *Random structures & algorithms*, 49(2):379–405, 2016.

[52] O. Parzanchevski and R. Rosenthal. Simplicial complexes: spectrum, homology and random walks. *Random Structures & Algorithms*, 50(2):225–261, 2017.

[53] R. J. Plemmons. M-matrices leading to semiconvergent splittings. *Linear Algebra and its Applications*, 15(3):243–252, 1976.

[54] E. Ribando-Gros, R. Wang, J. Chen, Y. Tong, and G.-W. Wei. Combinatorial and Hodge Laplacians: Similarities and Differences. *SIAM Review*, 66(3):575–601, 2024.

[55] T. M. Roddenberry, N. Glaze, and S. Segarra. Principled simplicial neural networks for trajectory prediction. In *Proc. of ICML*, 2021.

[56] G. Samorodnitsky. Long range dependence. *Foundations and Trends® in Stochastic Systems*, 1(3):163–257, 2007.

[57] M. T. Schaub, A. R. Benson, P. Horn, G. Lippner, and A. Jadbabaie. Random walks on simplicial complexes and the normalized Hodge 1-Laplacian. *SIAM Review*, 62(2):353–391, 2020.

[58] D. Taha, J. Chapman, M. Eidi, K. Devriendt, and G. Montufar. Demystifying topological message-passing with relational structures: A case study on oversquashing in simplicial message-passing. In *Proc. of ICLR*, 2025.

[59] C. Tang, Z. Ye, H. Zhao, L. Bai, and J. Lin. DeepSCNN: a simplicial convolutional neural network for deep learning. *Applied Intelligence*, 55(4):281, 2025.

[60] K.-C. Toh and L. N. Trefethen. The chebyshev polynomials of a matrix. *SIAM Journal on Matrix Analysis and Applications*, 20(2):400–419, 1998.

[61] Z. Tong, Y. Liang, C. Sun, X. Li, D. Rosenblum, and A. Lim. Digraph inception convolutional networks. In *Proc. of NeurIPS*, 2020.

[62] L. N. Trefethen. *Approximation theory and approximation practice, extended edition*. SIAM, 2019.

[63] Q. Truong and P. Chin. Weisfeiler and Lehman go paths: Learning topological features via path complexes. In *Proc. of AAAI*, 2024.

[64] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio. Graph attention networks. In *Proc. of ICLR*, 2018.

[65] Y. Verma, A. H. Souza, and V. Garg. Topological neural networks go persistent, equivariant, and continuous. *Proc. of ICML AI4Science*, 2024.

[66] Z. Wang, Y. Zhou, L. Hong, Y. Zou, H. Su, and S. Chen. Pairwise learning for neural link prediction. *arXiv preprint arXiv:2112.02936*, 2021.

[67] F. Wu, A. Souza, T. Zhang, C. Fifty, T. Yu, and K. Weinberger. Simplifying graph convolutional networks. In *Proc. of ICML*, 2019.

[68] K. Xu, W. Hu, J. Leskovec, and S. Jegelka. How powerful are graph neural networks? In *Proc. of ICLR*, 2019.

[69] Y. Yan and E. E. Kuruoglu. Binarized simplicial convolutional neural networks. *Neural Networks*, 183:106928, 2025.

[70] Z. Yan, T. Ma, L. Gao, Z. Tang, and C. Chen. Link prediction with persistent homology: An interactive view. In *Proc. of ICML*, 2021.

[71] J.-H. Yang, C.-M. Chen, C.-J. Wang, and M.-F. Tsai. Hop-rec: high-order proximity for implicit recommendation. In *Proc. of ACM RecSys*, 2018.

[72] M. Yang, E. Isufi, and G. Leus. Simplicial convolutional neural networks. In *Proc. of IEEE ICASSP*, 2022.

[73] B. Zhang, C. Fan, S. Liu, K. Huang, X. Zhao, J. Huang, and Z. Liu. The expressive power of graph neural networks: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 2024.

[74] H. Zhang, J. Xia, and M. Xu. Differential encoding for improved representation learning over graphs. *IEEE Transactions on Big Data*, 12(1):276–287, 2026.

[75] M. Zhang and Y. Chen. Link prediction based on graph neural networks. In *Proc. of NeurIPS*, 2018.

[76] Q. Zhao, Z. Ye, C. Chen, and Y. Wang. Persistence enhanced graph neural network. In *Proc. of AISTATS*, 2020.

[77] C. Zhou, X. Wang, and M. Zhang. Facilitating graph neural networks with random walk on simplicial complexes. In *Proc. of NeurIPS*, 2024.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and Section 1 enumerate four contributions: (i) the introduction of Lévy flights and the fractional Hodge Laplacian for simplicial learning, (ii) a mixing-time speedup result for the fractional walk, (iii) the FHL-SNN architecture, and (iv) empirical link-prediction gains of up to 9%. Each is supported in the body: (i) by Definition 3.1 and Section 3, (ii) by Theorems 3.4 and 3.6, (iii) by Section 5 and Appendix B, and (iv) by Tables 2–6 and the ablation in Figure 4.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations are discussed throughout: the regime-dependence of the mixing-time speedup ($\lambda_2 < 1$, sparser heterogeneous graphs) is stated alongside Theorem 3.6; the degenerate behaviour of FHL-SNN on complexes with no higher-order structure (e.g., Disease) is acknowledged in Section 5 and confirmed empirically; the computational bottleneck of constructing the simplicial complex for very large graphs is discussed in Section 6 and Appendix D; and the deferred formal approximation-error theorem for the Chebyshev primitive is flagged at the end of Section 4.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.

- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: All theoretical results (Lemmas 3.2 and 3.3, Theorems 3.4 and 3.6).

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The architecture is fully described in Appendix B, the Chebyshev primitive in Algorithm 1, the data splits and optimizer in Appendix G, and the dataset statistics in Table 7. All datasets used are public benchmarks. An anonymized code repository is provided at `https://anonymous.4open.science/r/FHLSNN-9A21/`.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.

(b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.

(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: An anonymized code repository is released at `https://anonymous.4open.science/r/FHLSNN-9A21/`. All datasets used (Cora, Citeseer, Disease, Meetings, Calls, Wisconsin, Texas, PCQM-Contact, ogbl-collab) are publicly available standard benchmarks accessed via their original distributions.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Appendix G specifies the data splits ($85\%/5\%/10\%$ for train/validation/test), optimizer (Adam), loss (binary cross-entropy with negative sampling), number of runs (25), and PyTorch implementation. The fractional exponent γ is selected by grid search over $(0, 1]$ as discussed in Section 5, with the empirically optimal range $[0.3, 0.7]$ reported. The Chebyshev order K and architectural hyperparameters $\alpha, \beta, \delta, \eta$ are described in Section 5 and Appendix B.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.

- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: All main result tables (Tables 2, 3, 4, 6, and Figure 4) report mean and standard deviation over 25 runs with the same partitions, as stated in Appendix G. Statistical-significance markers ($*$ for $p < 0.1$ and $**$ for $p < 0.05$) are used to flag improvements that pass paired-test thresholds, e.g., Tables 2 and 3.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Appendix G reports that all experiments are run on a single NVIDIA RTX 4080 GPU with 12 GiB RAM. Section 4 and Appendix D report wall-clock runtimes for the Chebyshev primitive (Figure 2) and discuss per-epoch cost relative to baselines, including a $35\times$ speedup over Schur–Padé at $n = 1000$ and one-time preprocessing of 0.035s on Cora and 14.56s at full ogbl-collab scale.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes]

Justification: The research conforms with the NeurIPS Code of Ethics. All datasets used are publicly available benchmarks; no human subjects are involved; no personally identifying information is processed; and anonymity is preserved at submission via the anonymized code repository.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: A brief impact statement is included as Section 7. The work is foundational research on higher-order graph learning; we do not identify a specific direct path to negative societal applications beyond those generic to graph machine learning more broadly. Positive applications include improved link prediction in collaborative and citation networks and in criminal-network analysis where uncovering hidden hierarchical structure can support investigative work.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The paper does not release pre-trained generative models, scraped datasets, or other assets carrying a high risk of misuse. The released artefact is research code that operates on existing public benchmarks.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All datasets and baselines used are cited. All are used under their original public-research licenses, and we use them only for the research purposes for which they are distributed.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The released code at `https://anonymous.4open.science/r/FHLSNN-9A21/` contains a README with installation instructions, dataset preparation steps, and commands to reproduce the main results.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing or research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The paper does not involve human subjects research.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: LLMs are not part of the core method. The contributions are the fractional Hodge Laplacian operator, mixing-time analysis, Chebyshev primitive, and the FHL-SNN architecture, none of which involve LLMs.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.

Appendices

A Description of Notations

Table 1 lists description of all the notations used in the paper.

Table 1: Table of Notations.

Notation	Description
$G = (V, E, X)$	Graph with vertex set V, edge set E, and feature matrix X
$\|V\| = n$	Number of nodes
$\|E\| = m$	Number of edges
$A \in \{0,1\}^{n\times n}$	Adjacency matrix
$L_0 = D - A$	(Unnormalized) graph Laplacian
Δ	Abstract simplicial complex
σ	d-simplex with $\|\sigma\| = k+1$
$\tau \subseteq \sigma$	Face of simplex σ
C_N	Vector space of N-chains
L_N	N-Hodge Laplacian
L_N^{down}	Down-Laplacian ($\delta_{N-1}^T \delta_{N-1}$)
L_N^{up}	Up-Laplacian ($\delta_N \delta_N^T$)
L_N^{γ}	Fractional Hodge N-Laplacian
$V\Lambda V^T$	Spectral decomposition of Laplacian
$m_{u\to v}^{(\gamma)}$	Lévy flight transition probability
$p_0^{\gamma}(t)$	Return probability for Lévy flight
$p_0^{1}(t)$	Return probability for standard random walk
π_t	Marginal distribution at time t
π_∞	Limiting distribution
p	Arbitrary laziness parameter $0 < p < 1$ height

B Fractional Hodge Laplacian Meets Block Simplicial Neural Networks

Armed with the derived theoretical underpinnings, we propose the Fractional Hodge Laplacian Simplicial Neural Network (FHL-SNN), a novel approach combining fractional dynamics with hierarchical simplicial structure learning. The model leverages Lévy flight characteristics through fractional Hodge Laplacians (FHL) to capture long-range dependencies, while maintaining the ability to learn hierarchical topological features. This unique combination allows the model to better capture both local and non-local interactions among graph (sub)structures. At its core, FHL-SNN employs a block operator structure, incorporating fractional Hodge Laplacians:

$$\mathfrak{L}_2^{\gamma} = \left[\begin{array}{c|c} \boldsymbol{L}_{N_1}^{\gamma} & f(\mathbb{P}_{N_2}\boldsymbol{L}_{N_1}^{\gamma}, \boldsymbol{L}_{N_2}^{\gamma}) \\ \hline f(\mathbb{P}_{N_2}\boldsymbol{L}_{N_1}^{\gamma}, \boldsymbol{L}_{N_2}^{\gamma})^{\top} & \boldsymbol{L}_{N_2}^{\gamma} \end{array}\right],$$

where $\gamma \in (0, 1]$ controls the fractional power enabling Lévy flight-like behavior. (For non-fractional case such block structure has interpretation in terms of the Dirac operator [6, 12, 43], with its application to graph learning discussed by [3, 13, 17].) The similarity function $f(\cdot, \cdot)$ captures interactions between different dimensional simplices and can be implemented either as a simple inner product or through a learned embedded representation:

$$f([\mathbb{P}_{N_2}\boldsymbol{L}_{N_1}^{\gamma}]_i, [\boldsymbol{L}_{N_2}^{\gamma}]_j) = \begin{cases} < [\mathbb{P}_{N_2}\boldsymbol{L}_{N_1}^{\gamma}]_i, [\boldsymbol{L}_{N_2}^{\gamma}]_j > \\ < \boldsymbol{\Theta}_{\xi}[\mathbb{P}_{N_2}\boldsymbol{L}_{N_1}^{\gamma}]_i, \boldsymbol{\Theta}_{\psi}[\boldsymbol{L}_{N_2}^{\gamma}]_j >, \end{cases}$$

where $\boldsymbol{\Theta}_{\xi}$ and $\boldsymbol{\Theta}_{\psi}$ are trainable parameters. The block operator is normalized through a ReLU-softmax combination to ensure proper scaling and eliminate weak connections:

$$\tilde{\mathfrak{L}}_2^{\gamma} = \text{softmax}(\text{ReLU}(\mathfrak{L}_2^{\gamma})).$$

Information processing occurs through two parallel pathways, i.e., a fractional Hodge convolution path and a standard graph convolution path. The fractional Hodge convolution is defined as:

$$\boldsymbol{Z}^{(\ell+1)} = (\tilde{\mathfrak{L}}_2^{\gamma}\boldsymbol{Z}^{(\ell)}\boldsymbol{\Theta}_1^{(\ell)})\boldsymbol{\Theta}_2^{(\ell)}, \tag{8}$$

where $\mathbf{\Theta}_1^{(\ell)}$ and $\mathbf{\Theta}_2^{(\ell)}$ are trainable parameters. Then we compute distances in both pathways:

$$\text{dist}_{uv}^{\text{FH}} = (\boldsymbol{Z}_u^{(\ell+1)} - \boldsymbol{Z}_v^{(\ell+1)})^2,$$
$$\text{dist}_{uv}^{\text{GC}} = (\boldsymbol{H}_u^{(\ell+1)} - \boldsymbol{H}_v^{(\ell+1)})^2.$$

Finally, predictions are made by combining information from both pathways through a weighted concatenation and MLPs:

$$Prob(u,v) = \left[\exp\left(\frac{\text{MLPs}([\alpha \cdot \text{dist}_{uv}^{\text{GC}}, \beta \cdot \text{dist}_{uv}^{\text{FH}}]) - \delta}{\eta}\right) + 1\right]^{-1},$$

where α, β are hyperparameters to control the importance of two derived distances.

The proposed architecture offers several key advantages. First, the incorporation of fractional Hodge Laplacians enables us to capture long-range dependencies through Lévy flights, allowing the model to better understand non-local relationships in the graph structure. Second, the dual-pathway design effectively combines local and global structural information, providing a more comprehensive view of the graph topology. Third, the learned similarity function enables adaptive interaction between simplices of different dimensions, capturing complex hierarchical relationships. Finally, the weighted combination of pathways allows the model to dynamically balance different types of structural information based on the specific requirements of the task.

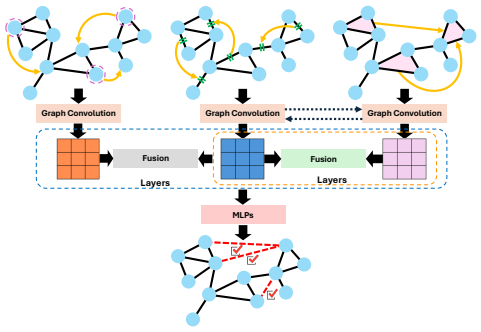


Figure 3: FHL-SNN architecture.

The entire model is end-to-end trainable using binary cross-entropy loss with negative sampling. This makes it suitable for tasks such as link prediction where understanding both local and global structural patterns is crucial. Additionally FHL provides theoretical advantages in terms of mixing time and exploration efficiency.

C Proof of Theorem 3.6

Proof: Let π_t be the marginal distribution of the random walk at time step t and be π_∞ the corresponding limiting distribution. Under the condition on λ_2, invoking Theorem 5.6 [51] (see also [23] for the intuitive exposition) yields

$$||\pi_t - \pi_\infty||_{TV} \leq C\left(1 - \frac{2(1-p)}{N+1}\lambda_2\right)^t, \tag{9}$$

where $||\cdot||_{TV}$ stands for total variation distance. Since $\frac{2(1-p)}{N+1}\lambda_2 < 1$, using the Taylor series expansion, we can re-formulate (9) as

$$||\pi_t - \pi_\infty||_{TV} \leq Ce^{-\varrho\lambda_2 t}, \quad \varrho = \frac{2(1-p)}{N+1}. \tag{10}$$

Recall that mixing time $t_{\text{mix}}(\epsilon)$ is given by

$$t_{\text{mix}}^\epsilon = \min\{t : \|\pi_t - \pi_\infty\|_{TV} \leq \epsilon\}.$$

Armed with the bound (10), we find that mixing time t_{mix} for the non-fractional random walk described by the transition probability matrix P can be estimated as

$$t_{\text{mix}}^\epsilon = \mathcal{O}\left(\frac{1}{\varrho\lambda_2}\log\left[\frac{C}{\epsilon}\right]\right). \tag{11}$$

Similarly, we derive mixing time τ_{mix} for Lévy flights over simplices with the transition probability matrix $\tilde{P}$ and $0 < \gamma \leq 1$:

$$\tau_{\text{mix}}^\epsilon = \mathcal{O}\left(\frac{1}{\varrho\lambda_2^\gamma}\log\left[\frac{C}{\epsilon}\right]\right). \tag{12}$$

Hence, under the assumptions on λ_2 and γ,

$$\tau^{\epsilon}_{\text{mix}} < t^{\epsilon}_{\text{mix}}, \tag{13}$$

and the mixing time delivered by Lévy flights is $\lambda_2^{\gamma-1}$ times lower than its non-fractional counterpart:

$$\frac{t^{\epsilon}_{\text{mix}}}{\tau^{\epsilon}_{\text{mix}}} = \frac{1/\lambda_2}{1/\lambda_2^{\gamma}} = \lambda_2^{\gamma-1}, \tag{14}$$

This implies that the speedup factor due to to fractional exploration is $\mathcal{O}\big(\lambda_2^{-(1-\gamma)}\big)$, which concludes the proof.

□

D Experimental Results

Experimental Settings. We now turn to numerical evaluation of FHL-SNN and consider link prediction tasks, both on directed and undirected graphs, under various scenarios of heterogeneity, including criminal networks (with separate datasets for Meetings and Phone Calls [14], disease networks, citation networks (Cora, Citeseer), long range dependence benchmark for link prediction (PCQM) [21] and large scale co-authorship benchmark graph (ogbl-collab) [33]. These datasets represent diverse application domains which display graph heterogeneity, including long-range dependencies, is crucial for reliable link prediction. We compare FHL-SNN against 33 baselines (see details in Appendix F.1) including emerging state-of-the-art (SOTA) methods such as Differential Encoding [74] Moreover, to demonstrate versatility of the FHL idea, we integrate FHL into other competing model architectures. Finally, we provide an ablation study, where we investigate contribution of FHL. The dataset summaries are presented in Appendix F and our code can be found at `https://anonymous.4open.science/r/FHLSNN-9A21/`.

Model	Meetings	Calls	Disease
MLP	63.20 ± 6.22	60.10 ± 6.72	72.62 ± 0.61
HNN	71.00 ± 3.28	60.90 ± 4.25	75.10 ± 0.35
GCN	72.08 ± 4.19	61.50 ± 5.80	64.70 ± 0.56
GAT	74.00 ± 4.68	63.40 ± 5.20	69.99 ± 0.32
SAGE	72.30 ± 5.25	62.07 ± 5.49	65.91 ± 0.33
SGC	73.38 ± 3.49	63.80 ± 5.71	65.21 ± 0.23
SEAL	71.09 ± 7.50	62.96 ± 4.17	85.23 ± 0.79
HGCN	83.20 ± 4.15	70.20 ± 3.77	90.80 ± 0.30
PEGN	74.17 ± 5.00	65.23 ± 4.15	83.61 ± 1.26
TLC-GNN	73.20 ± 5.32	66.17 ± 3.90	86.19 ± 1.23
BScNets	88.05 ± 5.51	$\underline{79.43} \pm 6.04$	$\mathbf{98.60 \pm 0.58}$
SNN	84.28 ± 4.88	75.66 ± 6.92	94.34 ± 0.77
Diff. Encoding	89.20 ± 7.74	78.10 ± 6.36	93.70 ± 0.55
GEAET	$\underline{89.80} \pm 3.79$	78.50 ± 5.55	97.60 ± 0.39
FHL-SNN	$\mathbf{90.21 \pm 4.44}$*	$\mathbf{81.27 \pm 3.78}$**	$\underline{98.35} \pm 0.56$

Table 2: **Undirected** link prediction on social and biological data. Best accuracy results are in **bold**, and the 2nd best are underlined.

Undirected Link Prediction. As Table 2 shows, FHL-SNN demonstrates better ROC AUC scores on the criminal networks (Meetings and Calls) over all competitors, highlighting the effectiveness of FHL-SNN's fractional dynamics in capturing polyadic structural patterns. This is not surprising as criminal networks are characterized by intrinsic heterogeneous hierarchical structure. In turn, FHL-SNN achieves comparable performance to BScNets and performs second best on Disease dataset, which maybe expected since the Disease data exhibit the lowest clustering coefficient, indicating smaller number of higher order interactions. This demonstrates that our approach maintains competitive performance even in cases where cliques of higher order maybe sparse.

Undirected Link Prediction under Long Range Dependence. We evaluate FHL-SNN on the PCQM-Contact dataset from the Long Range Benchmark [21], which comprises 529,434 molecular graphs with 15.9 million nodes (average 30.14 nodes per graph, mean degree 2.03), specifically

designed to challenge models with long-range dependency modeling, where atoms are separated by more than 5 hops in the molecular graph but remain physically close in 3D space. In Table 3, we report Hits@1, Hits@3 and Hits@10 for the test data. FHL-SNN achieves statistically significant result of 0.1471 Hits@1 score over the second best model. Similar gains for FHL-SNN are also observed for Hits@3 and Hits@10. This demonstrates that FHal can efficiently encode distant node information in molecular networks, while conventional approaches struggle with the fundamental trade-off between local message passing and global structural understanding.

Model	**Test Hits@1**	**Test Hits@3**	**Test Hits@10**
GCN	0.1298±0.0011	0.3745±0.0005	0.8222±0.0008
GCNII	0.1319±0.0010	0.3623±0.0004	0.8111±0.0009
GINE	0.1339±0.0011	0.3633±0.0041	0.8148±0.0061
GatedGCN	0.1219±0.0012	0.3751±0.0006	0.8431±0.0010
GatedGCN+RWSE	0.1288±0.0013	0.3808±0.0006	0.8517±0.0005
Transformer+LapPE	0.1223±0.0012	0.3670±0.0043	0.8528±0.0040
SAN+LapPE	0.1351±0.0018	0.4002±0.0021	0.8471±0.0043
SAN+RWSE	0.1312±0.0016	0.4030±0.0008	0.8550±0.0024
Differential Encoding	<u>0.1431</u>± 0.0048	0.409 ± 0.0029	<u>0.8601</u> ± 0.0049
GEAET	0.1392 ± 0.0033	<u>0.4108</u> ± 0.0047	0.8594 ± 0.0056
FHL-SNN (ours)	**0.1471 ± 0.0039***	**0.4177 ± 0.0028***	**0.8690 ± 0.0032***

Table 3: **Undirected** link prediction performance on LRB PCQM-Contact data in terms of Hits@1, and Hits@3, Hits@10; best results are in **bold**, and the 2nd best are <u>underlined</u>; * and ** denote significance (p-val. < 0.1) and stat. sign. (p-val. < 0.05), respectively.

Undirected Link Prediction for Large Scale Graphs. Table 4 presents evaluation of FHL-SNN on the ogbl-collab dataset, a large-scale collaboration network from the Open Graph Benchmark [33]. Here nodes are authors and edges denote co-authorship relationships. This dataset presents a

Model	Hit@50
PNLP with RW	0.33 ± 0.02
Refined-GAE	0.31 ± 0.03
Hop-Rec	0.27 ± 0.02
TopoLink	0.26 ± 0.04
FHL-SNN (ours)	**0.37 ± 0.02***

Table 4: **Undirected** link prediction on ogbl-collab; * is significant

challenging link prediction task due to its complex hierarchical structure. To address computational scalability while maintaining a representative graph structure, we implement a sampling strategy to select 10% of the nodes. We use inverse degree sampling as the probability of selecting a node to mitigate bias towards high-degree nodes [40]. Our FHL-SNN achieves statistically significant improvements over established SOTA baselines, including Refined-GAE [45]. With 20% sub-sample, we get a performance of 00.553 ± 0.018, whereas on full OGB, we get 0.685 ± 0.0051 which is competitive with leaderboard. These results highlight that fractional FHL effectively captures the multiscale collaborative patterns inherent in academic networks, where successful link prediction requires model to explore both local research clusters and distant patterns across different disciplines.

Directed Link Prediction We now turn to directed citation networks Cora and Citeseer and report ROC AUC for 10 baselines in Table 5, including the emerging powerful SOTA models specialized for directed link prediction task such as DHGNN [46]. Similarly to the undirected case, we observe that FHL-SNN yields a competitive performance, suggesting that the underlying complex hierarchical structure of citation networks benefits from FHL-SNN's ability to capture multi-scale interactions. Our analysis reveals that the FHL-SNN consistently improves the capture of long-range dependencies, as evidenced by better performance on citation networks, where influential papers may be several hops apart. Specifically in the case of directed graphs, the orientation-preserving properties of our simplicial complex representation provide valuable structural information.

Model	AUC	
	Cora	Citeseer
GCN	91.73	87.98
GAT	91.35	86.18
GraphSAGE	92.12	88.42
PNA	93.49	88.21
HGNN	92.77	90.85
HNHN	91.88	88.41
HGNN+	93.69	92.58
DiGCN	93.80	92.86
DiGAE	93.68	90.34
DHGNN	<u>94.20</u>	<u>93.77</u>
FHL-SNN (ours)	**94.42**	**95.18**

Table 5: **Directed** link prediction on citation networks. Best results are in **bold**, and the 2nd best are <u>underlined</u>.

Model	Accuracy	
	Cora	Citeseer
SNN	85.68 ± 0.031	82.39 ± 0.036
SNN**+FHL**	$\mathbf{87.25 \pm 0.026}$	$\mathbf{84.72 \pm 0.035}$
SCNN	87.22 ± 0.022	84.31 ± 0.018
SCNN**+FHL**	$\mathbf{87.47 \pm 0.023}$	$\mathbf{86.94 \pm 0.018}$
SAT	86.42 ± 0.019	84.59 ± 0.011
SAT**+FHL**	$\mathbf{87.71 \pm 0.015}$	$\mathbf{85.65 \pm 0.011}$
HiGCN	89.27 ± 0.021	85.73 ± 0.014
HiGCN**+FHL**	$\mathbf{89.33 \pm 0.017}$	$\mathbf{86.52 \pm 0.110}$

Table 6: FHL Versatility: Directed link prediction performance, where FHL is integrated into other SNN SOTAs.

Ablation Study To isolate the specific contribution of Lévy flight induced by FHL, we conduct an ablation study comparing the base block operator used in BscNet against two variants, where the regular Hodge Laplacian is replaced by (i) graph Laplacian and (ii) our proposed FHL. In Fig. 4 the results demonstrate that FHL consistently outperforms both baselines across all five datasets, including heterogeneous graphs. The standard graph Laplacian consistently has lower performance, degrading on an average by 3.32% compared to BScNet. FHL shows particularly strong gains on heterogeneous networks, such as Meetings and Wisconsin, where long-range dependencies are critical for link prediction. This validates our theoretical foundation that Lévy flights over simplices provide more efficient exploration of higher-order graph substructures compared to traditional spectral approaches. The consistent performance improvements across diverse networks highlight the fundamental advantage of fractional dynamics in capturing both local simplicial structure and non-local dependencies essential for link prediction tasks.

Operator choice and its impact The degree-2 condition is required to read the transition matrix as a probability, and the M-matrix property it rests on is currently known only in that setting (Mukherjee & Steenbergen, RSA 2016). The Hodge Laplacian is symmetric PSD on any complex, so its fractional power is always a valid spectral filter and always raises each eigenvalue to the power γ, widening the spectral gap with no structural assumption, and that is the only property FHL-SNN depends on. GCN likewise uses the graph Laplacian as a spectral filter without requiring regular degree.

So, from the theory point of view, theoretical results are valid only for the $\boldsymbol{L}_{\mathrm{down}}$ operator. Empirically the performance is comparable across $\boldsymbol{L}_{\mathrm{down}}^{\gamma}$, $\boldsymbol{L}_{\mathrm{up}}^{\gamma}$, and $\boldsymbol{L}_{N}^{\gamma}$ (full).

Further from table 7 we observe that empirically the architecture is robust to the component operator:

Choice of Fractional Parameter We observe that optimal values of the fractional power parameter γ typically lie in the range $[0.3, 0.7]$. This aligns with our theoretical analysis suggesting that Lévy

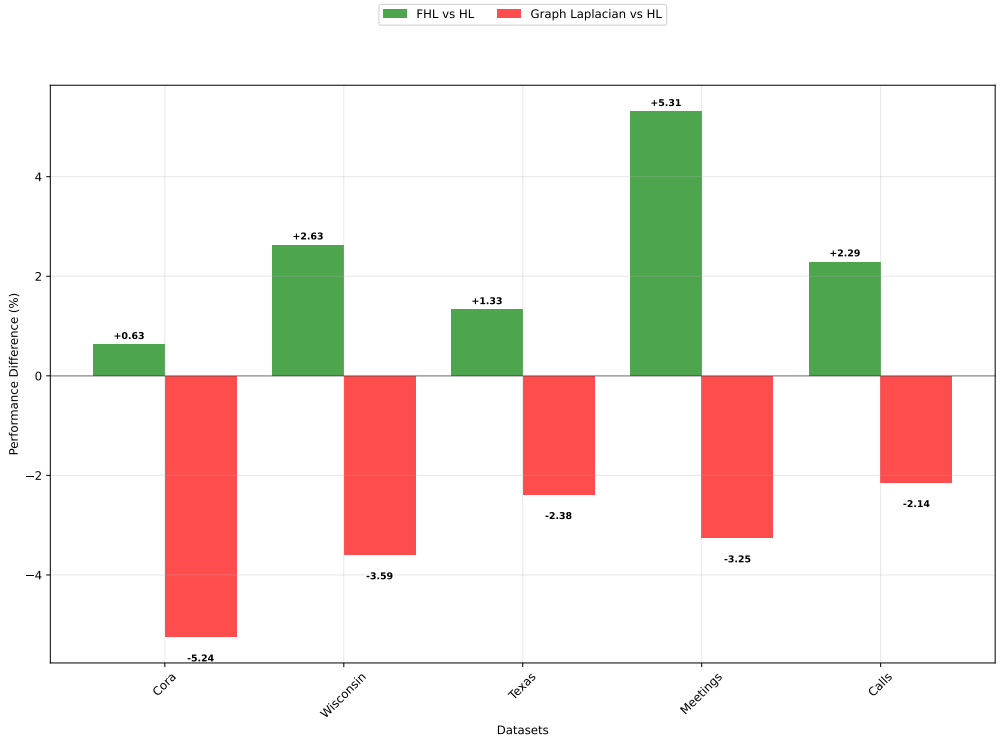


Figure 4: Ablation study comparing performance gains when fractional dynamics with Hodge Laplacian is considered relative to BscNet block operator, and BscNet with graph Laplacian.

Table 7: Impact of the component operator in FHL-SNN.

Operator	Cora	Citeseer	Meetings	Disease
$\boldsymbol{L}_{\text{down}}^{\gamma}$	92.77 ± 2.11	90.86 ± 1.59	88.83 ± 5.59	94.47 ± 2.61
$\boldsymbol{L}_{\text{up}}^{\gamma}$	90.42 ± 1.08	89.81 ± 1.72	86.93 ± 5.84	92.81 ± 1.45
$\boldsymbol{L}_{N}^{\gamma}$ (full)	94.42 ± 0.79	95.18 ± 1.33	90.21 ± 4.44	98.35 ± 0.56

flights provide more efficient exploration of directed graph structures. In our experiments, we use grid search to find optimal fractional parameter. Performance gains are particularly pronounced for graphs with more heterogeneous structure and sparse regions of the graphs, where traditional random-walk approaches often struggle to efficiently explore the network topology.

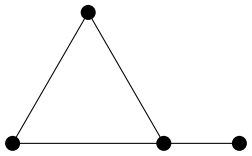
Versatility of the FHL Idea To demonstrate versatility of our FHL approach, we systematically integrate FHL into four existing SNN architectures. Table 6 shows consistent performance improvements across all models and datasets when they are augmented with FHL. Notably, the gains are most substantial for simpler architectures (SNN, SCNN, and SAT), suggesting that FHL provides complementary capabilities that are particularly valuable when the base model has fewer sophisticated mechanisms for capturing long-range dependencies. Even the already powerful HiGCN [34] architecture benefits from FHL integration, indicating that inclusion of fractional dynamics offers fundamental advantages in presence of high order structures.

Computational Costs The primary computational bottleneck in FHL-SNN is computing the fractional power $\mathbf{L}_d^{\gamma}$. While the Schur-Padé algorithm [32] requires $O(n^3)$ operations, we employ Chebyshev polynomial approximation for improved efficiency. Specifically, we approximate $\mathbf{L}_N^{\gamma} \approx \sum_{k=0}^{K} c_k T_k(\tilde{\mathbf{L}}_N)$, where T_k denotes the k-th Chebyshev polynomial, $\tilde{\mathbf{L}}_N = 2\mathbf{L}_N/\lambda_{\max} - \mathbf{I}$ is the rescaled Laplacian, and $\{c_k\}$ are the Chebyshev coefficients for t^{γ}. Leveraging the three-term recurrence $T_{k+1}(x) = 2xT_k(x) - T_{k-1}(x)$, which requires only K sparse matrix-vector multiplications, yielding $O(Km)$ complexity for a Hodge Laplacian with m non-zero entries. We present

a rigorous computational analysis of our approximation, run time analysis and sensitivity of the fractional γ parameter in figure 2 and present further details in section G.

E Toy Example of Hodge Laplacian Construction

Consider a simple triangle with an extra edge:



This graph has:

- 4 vertices (labeled 1,2,3,4)
- 4 edges (labeled a,b,c,d)
- 1 triangle

Step 1: Boundary Operator δ_0 (vertices to edges)

For oriented edges:

- a: $1 \to 2$
- b: $2 \to 3$
- c: $3 \to 1$
- d: $2 \to 4$

δ_0 is a 4×4 matrix:

$$\delta_0 = \begin{bmatrix} -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 1 & 0 & -1 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix}$$

Step 2: Boundary Operator δ_1 (edges to triangles)

For the clockwise-oriented triangle:

$$\delta_1 = \begin{bmatrix} 1 & 1 & -1 & 0 \end{bmatrix}$$

Step 3: Computing Hodge Laplacians

Hodge 0-Laplacian $\boldsymbol{L}_0 = \delta_0^* \delta_0$:

$$\boldsymbol{L}_0 = \begin{bmatrix} 2 & -1 & -1 & 0 \\ -1 & 3 & -1 & -1 \\ -1 & -1 & 2 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix}$$

Hodge 1-Laplacian $\boldsymbol{L}_1 = \delta_0 \delta_0^* + \delta_1^* \delta_1$: First compute $\delta_0 \delta_0^*$:

$$\delta_0 \delta_0^* = \begin{bmatrix} 2 & -1 & -1 & -1 \\ -1 & 2 & 0 & 0 \\ -1 & 0 & 2 & 0 \\ -1 & 0 & 0 & 2 \end{bmatrix}$$

Then compute $\delta_1^*\delta_1$:

$$\delta_1^*\delta_1 = \begin{bmatrix} 1 & 1 & -1 & 0 \\ 1 & 1 & -1 & 0 \\ -1 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Therefore:

$$\boldsymbol{L}_1 = \begin{bmatrix} 3 & 0 & -2 & -1 \\ 0 & 3 & -1 & 0 \\ -2 & -1 & 3 & 0 \\ -1 & 0 & 0 & 2 \end{bmatrix}$$

Hodge 2-Laplacian $\boldsymbol{L}_2 = \delta_1\delta_1^*$:

$$\boldsymbol{L}_2 = [3]$$

F Additional Experimental Details

Table 8 provides network summary statistics of the datasets used for our empirical evaluation of FHL-SNN.

Metric	Meetings	Calls	Disease	Cora	Citeseer
Nodes	101	100	2,665	2,708	3,327
Edges	256	124	2,663	5,278	4,552
Avg. Degree	5.07	2.48	2.00	3.90	2.74
Max Degree	24	25	12	168	99
Triangles	284	22	0	1,630	1,167
Clust. Coef.	0.656	0.105	0.000	0.241	0.141
Density	0.051	0.025	0.001	0.001	0.001
Diameter	7	7	8	19	28

Table 8: Network statistics across different datasets.

We implement our proposed FHL-SNN with Pytorch framework on one NVIDIA RTX 4080 GPU with 12 GiB RAM. We follow the standard data splitting procedure in the literature, for all datasets and randomly split edges into 85%/5%/10% for training, validation, and testing. Further, for all datasets, FHL-SNN is trained by the Adam optimizer with the Cross Entropy Loss function. For all methods, we run 25 times with the same partition and report the average and standard deviation of performance metrics used.

F.1 Additional Details on Competing Methods

In our results section, we used 33 baselines. For undirected link prediction, our experiments involved comparing against several models such as HNN [26], GCN [37], GAT [64], SAGE [30], SGC [67], SEAL [75], HGCN [14], PEGN [76], TLC-GNN [70], BscNets [17] and SNN [22]. For link prediction task over LRB PCQM-dataset our baseliness included GCN2 [15], GINE [68], GatedGCN [11], GatedGCN+RWSE, Transformer+LapPE [20], SAN+LapPE [39], SAN+RWSE. For link prediction experiments on ogbl-collab dataset we have 4 SOTA baselines; PNLP with Random walk [66], Refined-GAE [45], Hop-Rec [71] and TopoLink. For directed link prediction task, we include; HGNN [14], HGNN+ [27], DiGCN [61], DiGAE [38] and DHGNN [46]. For further showing versatility of our FHL model, we further compare it with existing SNN architectures such as SNN [22], SCNN [72] and SAT [28] adding fractional dynamics to them.

G Chebyshev Approximation

Lemma G.1. *Let ϕ_K be the degree-K Chebyshev polynomial approximation of $f(x) = x^\gamma$ on $[0, \lambda_{\max}]$, where $\lambda_{\max}$ is the largest eigenvalue of $\boldsymbol{L}_N^{down}$, the coefficients of ϕ_K are computed according to Algorithm 1, and $0 < \gamma \leq 1$. Let $\tilde{\boldsymbol{P}}$ be the transition matrix induced by the exact*

fractional Laplacian $(\boldsymbol{L}_N^{down})^\gamma$ *and* $\hat{\boldsymbol{P}}_{\boldsymbol{K}}$ *be its approximated counterpart induced by the Chebyshev approximation* $\phi_K\big[(\boldsymbol{L}_N^{down})\big]$*. Then*

$$||\tilde{\boldsymbol{P}} - \hat{\boldsymbol{P}}_{\boldsymbol{K}}||_2 \leq \varrho\varepsilon_K,$$

where $\varrho = \frac{2(1-p)}{(N+1)}$ *and* $\frac{1}{2} \leq p < 1$*. Moreover,* $\varepsilon_K \to 0$ *as* $K \to \infty$*.*

Proof: Note that since $\boldsymbol{L}_N^{down}$ is a real symmetric positive semi-definite matrix,

$$\phi_K\big[(\boldsymbol{L}_N^{down})\big] = U\phi_K(\Lambda)U^T,$$

where U is an orthonormal $?\times?$-matrix formed by eigenvectors of $\boldsymbol{L}_N^{down}$ and $\Lambda = diag\big\{\lambda_1, \ldots, \lambda_?\big\}$ is a diagonal matrix, where λ_i, $i = 1, \ldots, ?$ are eigenvalues of $\boldsymbol{L}_N^{down}$ (see Chapter 1 of [31] and Section 5 of [60]).

Hence,

$$\phi_K\big[(\boldsymbol{L}_N^{down})\big] - (\boldsymbol{L}_N^{down})^\gamma = Udiag\big\{\phi_K(\lambda_i) - \lambda_i^\gamma\big\}_{i=1}^{??}U^T,$$

and as such, we get

$$||\phi_K\big[(\boldsymbol{L}_N^{down})\big] - (\boldsymbol{L}_N^{down})^\gamma||_2 = \max_{\lambda_i \in \mathfrak{S}(\boldsymbol{L}_N^{down})} |\phi_K(\lambda_i) - \lambda_i^\gamma|,$$

where $\mathfrak{S}(\boldsymbol{L}_N^{down})$ is the spectrum of $\boldsymbol{L}_N^{down}$.

Setting $\varepsilon_K = \max_{x\in[0,\lambda_{\max}]} |\phi_K(x) - x^\gamma|$, we note that $\varepsilon_K \to 0$ as $K \to \infty$, since $f(x) = x^\gamma$, $x \in [0, \lambda_{\max}]$, is a continuous function bounded variation on $x \in [0, \lambda_{\max}]$, the corresponding sequence of Chebyshev approximations converges uniformly on a closed interval [62].

Now, recall that the transition matrix induced by the exact fractional Laplacian $(\boldsymbol{L}_N^{down})^\gamma$ is given by

$$\tilde{\boldsymbol{P}} = \boldsymbol{I} - \varrho(\boldsymbol{L}_N^{down})^\gamma.$$

Similarly, its counterpart $\hat{\boldsymbol{P}}_{\boldsymbol{K}}$ induced by approximating the fractional Laplacian $(\boldsymbol{L}_N^{down})^\gamma$ with Chebyshev polynomials of degree K is given by

$$\hat{\boldsymbol{P}}_{\boldsymbol{K}} = \boldsymbol{I} - \varrho\phi_K\big[(\boldsymbol{L}_N^{down})\big].$$

Then

$$||\tilde{\boldsymbol{P}} - \hat{\boldsymbol{P}}_{\boldsymbol{K}}||_2 = \varrho||(\boldsymbol{L}_N^{down})^\gamma - \phi_K\big[(\boldsymbol{L}_N^{down})\big]||_2 = \varrho \max_{\lambda_i \in \mathfrak{S}(\boldsymbol{L}_N^{down})} |\phi_K(\lambda_i) - \lambda_i^\gamma| \leq \varrho\varepsilon_K,$$

which concludes the proof.

Indeed, since $(\boldsymbol{L}_N^{down})^r\mathbb{1} = 0$, for $r \in \mathbb{Z}^+$, it immediately follows that $\phi_K\big[(\boldsymbol{L}_N^{down})\big]\mathbb{1} = \phi_K(0)\mathbb{1}$ and, hence, $\hat{\boldsymbol{P}}_{\boldsymbol{K}}\mathbb{1} = (1 - \varrho\phi_K\big[0\big])\mathbb{1}$. So, we can rescale $\hat{\boldsymbol{P}}_{\boldsymbol{K}}\mathbb{1}$ by $(1 - \varrho\phi_K\big[0\big])^{-1}$ to make $\hat{\boldsymbol{P}}_{\boldsymbol{K}}$ right stochastic. However, preservation of entrywise non-negativity is more tricky as it requires additional assumptions, such as a sufficiently small approximation error relative to the smallest positive transition probability, and is not guaranteed in general. Still, the Lemma above holds and ensures that so the approximate transition operator $\hat{\boldsymbol{P}}_{\boldsymbol{K}}$ converges to the exact one $\tilde{\boldsymbol{P}}$ in operator norm as $K \to \infty$.

Empirical study of K. We vary the polynomial order K on Cora at $\gamma = 0.5$. The operator-level row-sum deficit falls by $8\times$ across this range, halving with each doubling of K as predicted, and runtime grows roughly linearly. The approximation error, empirically, does not appear to propagate much to downstream predictions, and $K = 8$ matches $K = 32$ at half the cost.

We also provide a brief analysis on row and column sums. For the Chebyshev-approximated operator $\widetilde{\mathbf{P}}_K = \mathbf{I} - \rho_{,p_K}(\mathbf{L})$, row sums agree and all entries remain non-negative, e.g., the row sums are $0.9498, 0.9833, 0.9917$ at $K = 10, 30, 60$ for $\gamma = 0.5$. Similar results are observed for column sums.

K	Row-sum deficit	Accuracy	Wall-clock time / run (s)
4	0.1274	85.95 ± 1.82	22
8	0.0628	86.34 ± 1.06	28
16	0.0313	86.15 ± 1.40	40
32	0.0156	86.13 ± 1.27	63

Table 9

H Z-and M-matrices and property-c

M-matrices and Property *c*. A matrix $A \in \mathbb{R}^{n\times n}$ is called a *Z-matrix* if $a_{ij} \leq 0$ for all $i \neq j$. A Z-matrix is called an *M-matrix* if it admits the representation $A = sI - B$, where $B \geq 0$ and $s \geq \rho(B)$, with $\rho(B)$ denoting the spectral radius of B.

Following [4], a M-matrix is said to have *property c* if it can be represented as $A = sI - B$, $B \geq 0$, such that the matrix $\frac{1}{s}B$ is *semi-convergent*, i.e., $\lim_{k\to\infty}\left(\frac{B}{s}\right)^k$ exists. Property c is essential in the convergence and stability analysis of iterative numerical algorithms.

I. e., to study diffusion on these higher-order structures, we consider the vector spaces C^N of real-valued N-cochains, i.e., functions $f : S_N \to \mathbb{R}$ defined on the set S_N of oriented N-simplices. Each cochain is identified with a vector in $\mathbb{R}^{|S_N|}$, where $|S_N|$ denotes the number of N-simplices in the simplicial complex. The cochain spaces are connected through the coboundary operators $\delta_N : C^N \to C^{N+1}$ and their adjoints δ_N^*, where

$$\delta_N(f)([v_0,\ldots,v_{N+1}]) = \sum_{i=0}^{N+1}(-1)^i f([v_0,\ldots,\hat{v}_i,\ldots,v_{N+1}]).$$

On the orientation & the sign pattern of the specific Hodge Laplacian. Let $\sigma^N = v_0,\ldots,v_N \in \mathcal{K}_N$ be an N-simplex. An *orientation* of σ^N is an equivalence class of orderings of its vertices, where two orderings are equivalent if they differ by an even permutation. Hence, each simplex admits exactly two possible orientations: $[v_0,\ldots,v_N]$ and $-[v_0,\ldots,v_N]$. An *oriented simplex* is a simplex equipped with one of these two orientations. The choice of orientations induces the signed incidence (boundary) matrices. The Hodge Laplacian depends on the orientation only through the signs of the boundary operators, and reversing the orientation leaves the spectrum of the Hodge Laplacian unchanged. As shown by Roddenberry et al. (ICML 2021) and Yang et al. (2024), simplicial convolution is orientation equivariant. Hence, neither theory nor experiments for FHL-SNN are impacted by the orientation.

References

[1] V. L. Alev and L. C. Lau. Improved analysis of higher order random walks and applications. In *Proc. of STOC*, pages 1198–1211, 2020.

[2] A. Balakrishnan. Fractional powers of closed operators and the semigroups generated by them. *Pacific J. Math.*, 10(4):419–437, 1960.

[3] C. Battiloro, L. Testa, L. Giusti, S. Sardellitti, P. Di Lorenzo, and S. Barbarossa. Generalized simplicial attention neural networks. *IEEE Transactions on Signal and Information Processing over Networks*, 2024.

[4] A. Berman and R. J. Plemmons. *Nonnegative Matrices in the Mathematical Sciences*. SIAM, 1994.

[5] M. Besta, F. Scheidl, L. Gianinazzi, G. Kwasniewski, S. Klaiman, J. Müller, and T. Hoefler. Demystifying higher-order graph neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

[6] G. Bianconi. The topological dirac equation of networks and simplicial complexes. *Journal of Physics: Complexity*, 2(3):035022, 2021.

[7] M. Black, Z. Wan, A. Nayyeri, and Y. Wang. Understanding oversquashing in GNNs through the lens of effective resistance. In *Proc. of ICML*, 2023.

[8] C. Bodnar, F. Frasca, Y. Wang, N. Otter, G. F. Montufar, P. Lio, and M. Bronstein. Weisfeiler and Lehman go topological: Message passing simplicial networks. In *Proc. of ICML*, 2021.

[9] J. Böker, R. Levie, N. Huang, S. Villar, and C. Morris. Fine-grained expressivity of graph neural networks. In *Proc. of NeurIPS*, 2023.

[10] T. Bonis, L. Decreusefond, V. C. Tran, and Z. I. Zhang. Random walks on simplicial complexes. *arXiv preprint arXiv:2404.08803*, 2024.

[11] X. Bresson and T. Laurent. Residual gated graph convnets. *arXiv preprint arXiv:1711.07553*, 2017.

[12] L. Calmon, J. G. Restrepo, J. J. Torres, and G. Bianconi. Dirac synchronization is rhythmic and explosive. *Communications Physics*, 5(1):253, 2022.

[13] L. Calmon, M. T. Schaub, and G. Bianconi. Higher-order signal processing with the Dirac operator. In *Proc. of ACSSC*, 2022.

[14] I. Chami, Z. Ying, C. Ré, and J. Leskovec. Hyperbolic graph convolutional neural networks. In *Proc. of NeurIPS*, 2019.

[15] M. Chen, Z. Wei, Z. Huang, B. Ding, and Y. Li. Simple and deep graph convolutional networks. In *Proc. of ICML*, 2020.

[16] Y. Chen, Y. R. Gel, and K. Avrachenkov. LFGCN: Levitating over graphs with Levy flights. In *Proc. of ICDM*, 2020.

[17] Y. Chen, Y. R. Gel, and H. V. Poor. BScNets: Block simplicial complex neural networks. In *Proc. of AAAI*, 2022.

[18] F. Di Giovanni, T. K. Rusch, M. M. Bronstein, A. Deac, M. Lackenby, S. Mishra, and P. Veličković. How does over-squashing affect the power of GNNs? *TMLR*, 2024.

[19] P. Doukhan, G. Oppenheim, and M. Taqqu. *Theory and applications of long-range dependence*. Springer Science & Business Media, 2002.

[20] V. P. Dwivedi and X. Bresson. A generalization of transformer networks to graphs. In *Proc. of DLG-AAAI*, 2021.

[21] V. P. Dwivedi, L. Rampášek, M. Galkin, A. Parviz, G. Wolf, A. T. Luu, and D. Beaini. Long range graph benchmark. In *Proc. of NeurIPS*, 2022.

[22] S. Ebli, M. Defferrard, and G. Spreemann. Simplicial neural networks. In *Proc. of NeurIPS TDA Beyond 2020 Workshop*, 2020.

[23] M. Eidi and S. Mukherjee. Irreducibility of Markov chains on simplicial complexes, the spectrum of the discrete Hodge Laplacian and homology. *arXiv preprint arXiv:2310.07912*, 2023.

[24] F. Eijkelboom, R. Hesselink, and E. J. Bekkers. E (n) equivariant message passing simplicial networks. In *Proc. of ICML*, 2023.

[25] A. Einizade, D. Thanou, F. D. Malliaros, and J. H. Giraldo. Continuous Simplicial Neural Networks. In *Proc. of NeurIPS*, 2025.

[26] O. Ganea, G. Bécigneul, and T. Hofmann. Hyperbolic neural networks. In *Proc. of NeurIPS*, 2018.

[27] Y. Gao, Y. Feng, S. Ji, and R. Ji. HGNN+: General hypergraph neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3):3181–3199, 2022.

[28] C. W. J. Goh, C. Bodnar, and P. Lio. Simplicial attention networks. In *Proc. of ICLR GTRL*, 2022.

[29] S. Gurugubelli and S. P. Chepuri. Sann: Simple yet powerful simplicial-aware neural networks. In *Proc. of ICLR*, 2024.

[30] W. Hamilton, Z. Ying, and J. Leskovec. Inductive representation learning on large graphs. In *Proc. of NeurIPS*, 2017.

[31] N. J. Higham. *Functions of matrices: theory and computation*. SIAM, 2008.

[32] N. J. Higham and L. Lin. A schur–padé algorithm for fractional powers of a matrix. *SIAM Journal on Matrix Analysis and Applications*, 32(3):1056–1078, 2011.

[33] W. Hu, M. Fey, M. Zitnik, Y. Dong, H. Ren, B. Liu, M. Catasta, and J. Leskovec. Open graph benchmark: Datasets for machine learning on graphs. In *Proc. of NeurIPS*, 2020.

[34] Y. Huang, Y. Zeng, Q. Wu, and L. Lü. Higher-order graph convolutional network with flower-petals Laplacians on simplicial complexes. In *Proc. of AAAI*, 2024.

[35] J. L. Johnson and T. Goldring. Discrete Hodge theory on graphs: a tutorial. *Computing in Science & Engineering*, 15(5):42–55, 2012.

[36] A. Kalauch, S. Lavanya, and K. Sivakumar. Singular irreducible M-matrices revisited. *Linear Algebra and its Applications*, 565:47–64, 2019.

[37] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. In *Proc of ICLR*, 2017.

[38] G. Kollias, V. Kalantzis, T. Idé, A. Lozano, and N. Abe. Directed graph auto-encoders. In *Proc. of AAAI*, 2022.

[39] D. Kreuzer, D. Beaini, W. Hamilton, V. Létourneau, and P. Tossou. Rethinking graph transformers with spectral attention. In *Proc. of NeurIPS*, 2021.

[40] J. Leskovec and C. Faloutsos. Sampling from large graphs. In *Proc. of ACM SIGKDD*, 2006.

[41] P. Li and J. Leskovec. The expressive power of graph neural networks. *Graph Neural Networks: Foundations, Frontiers, and Applications*, pages 63–98, 2022.

[42] L.-H. Lim. Hodge Laplacians on graphs. *SIAM Review*, 62(3):685–715, 2020.

[43] S. Lloyd, S. Garnerone, and P. Zanardi. Quantum algorithms for topological and geometric analysis of data. *Nature communications*, 7(1):10138, 2016.

[44] C. López Amado, T. Schwarz, Y. Tian, and R. Lambiotte. Complex-weighted convolutional networks: provable expressiveness via complex diffusion. In *Proc. of LoG*, 2025.

[45] W. Ma, Y. Wang, X. Wang, and M. Zhang. Reconsidering the Performance of GAE in Link Prediction. *Proc. of CIKM*, 2025.

[46] Z. Ma, W. Zhao, and Z. Yang. Directed hypergraph representation learning for link prediction. In *Proc. of AISTATS*, 2024.

[47] H. Madhu, S. Gurugubelli, and S. P. Chepuri. Unsupervised parameter-free simplicial representation learning with scattering transforms. In *Proc. of ICML*, 2024.

[48] B. Mandelbrot. Long-run linearity, locally gaussian process, h-spectra and infinite variances. *International Economic Review*, 10(1):82–111, 1969.

[49] S. Maskey, R. Paolino, A. Bacho, and G. Kutyniok. A fractional graph Laplacian approach to oversmoothing. In *Proc. of NeurIPS*, 2024.

[50] C. Morris, G. Rattan, and P. Mutzel. Weisfeiler and Leman go sparse: Towards scalable higher-order graph embeddings. In *Proc. of NeurIPS*, 2020.

[51] S. Mukherjee and J. Steenbergen. Random walks on simplicial complexes and harmonics. *Random structures & algorithms*, 49(2):379–405, 2016.

[52] O. Parzanchevski and R. Rosenthal. Simplicial complexes: spectrum, homology and random walks. *Random Structures & Algorithms*, 50(2):225–261, 2017.

[53] R. J. Plemmons. M-matrices leading to semiconvergent splittings. *Linear Algebra and its Applications*, 15(3):243–252, 1976.

[54] E. Ribando-Gros, R. Wang, J. Chen, Y. Tong, and G.-W. Wei. Combinatorial and Hodge Laplacians: Similarities and Differences. *SIAM Review*, 66(3):575–601, 2024.

[55] T. M. Roddenberry, N. Glaze, and S. Segarra. Principled simplicial neural networks for trajectory prediction. In *Proc. of ICML*, 2021.

[56] G. Samorodnitsky. Long range dependence. *Foundations and Trends® in Stochastic Systems*, 1(3):163–257, 2007.

[57] M. T. Schaub, A. R. Benson, P. Horn, G. Lippner, and A. Jadbabaie. Random walks on simplicial complexes and the normalized Hodge 1-Laplacian. *SIAM Review*, 62(2):353–391, 2020.

[58] D. Taha, J. Chapman, M. Eidi, K. Devriendt, and G. Montufar. Demystifying topological message-passing with relational structures: A case study on oversquashing in simplicial message-passing. In *Proc. of ICLR*, 2025.

[59] C. Tang, Z. Ye, H. Zhao, L. Bai, and J. Lin. DeepSCNN: a simplicial convolutional neural network for deep learning. *Applied Intelligence*, 55(4):281, 2025.

[60] K.-C. Toh and L. N. Trefethen. The chebyshev polynomials of a matrix. *SIAM Journal on Matrix Analysis and Applications*, 20(2):400–419, 1998.

[61] Z. Tong, Y. Liang, C. Sun, X. Li, D. Rosenblum, and A. Lim. Digraph inception convolutional networks. In *Proc. of NeurIPS*, 2020.

[62] L. N. Trefethen. *Approximation theory and approximation practice, extended edition*. SIAM, 2019.

[63] Q. Truong and P. Chin. Weisfeiler and Lehman go paths: Learning topological features via path complexes. In *Proc. of AAAI*, 2024.

[64] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio. Graph attention networks. In *Proc. of ICLR*, 2018.

[65] Y. Verma, A. H. Souza, and V. Garg. Topological neural networks go persistent, equivariant, and continuous. *Proc. of ICML AI4Science*, 2024.

[66] Z. Wang, Y. Zhou, L. Hong, Y. Zou, H. Su, and S. Chen. Pairwise learning for neural link prediction. *arXiv preprint arXiv:2112.02936*, 2021.

[67] F. Wu, A. Souza, T. Zhang, C. Fifty, T. Yu, and K. Weinberger. Simplifying graph convolutional networks. In *Proc. of ICML*, 2019.

[68] K. Xu, W. Hu, J. Leskovec, and S. Jegelka. How powerful are graph neural networks? In *Proc. of ICLR*, 2019.

[69] Y. Yan and E. E. Kuruoglu. Binarized simplicial convolutional neural networks. *Neural Networks*, 183:106928, 2025.

[70] Z. Yan, T. Ma, L. Gao, Z. Tang, and C. Chen. Link prediction with persistent homology: An interactive view. In *Proc. of ICML*, 2021.

[71] J.-H. Yang, C.-M. Chen, C.-J. Wang, and M.-F. Tsai. Hop-rec: high-order proximity for implicit recommendation. In *Proc. of ACM RecSys*, 2018.

[72] M. Yang, E. Isufi, and G. Leus. Simplicial convolutional neural networks. In *Proc. of IEEE ICASSP*, 2022.

[73] B. Zhang, C. Fan, S. Liu, K. Huang, X. Zhao, J. Huang, and Z. Liu. The expressive power of graph neural networks: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 2024.

[74] H. Zhang, J. Xia, and M. Xu. Differential encoding for improved representation learning over graphs. *IEEE Transactions on Big Data*, 12(1):276–287, 2026.

[75] M. Zhang and Y. Chen. Link prediction based on graph neural networks. In *Proc. of NeurIPS*, 2018.

[76] Q. Zhao, Z. Ye, C. Chen, and Y. Wang. Persistence enhanced graph neural network. In *Proc. of AISTATS*, 2020.

[77] C. Zhou, X. Wang, and M. Zhang. Facilitating graph neural networks with random walk on simplicial complexes. In *Proc. of NeurIPS*, 2024.