Quest Journals
Journal of Research in Applied Mathematics
Volume 11 ~ Issue 12 (December 2025) pp: 100-108
ISSN (Online): 2394-0743 ISSN (Print): 2394-0735
www.questjournals.org

Research Paper

Integrating Discrete Topology with Graph Neural Networks A Framework for Directed Simplicial Complexes and Applications

Adel Ahmed Hassan Kubba[(1)]
Bastamy Mohamed Elmahadi [(2)]
[(1)] *Associate Professor - Nile Valley University - Faculty of Education*
[(2)] *Student*

Abstract: *This paper explores the synergistic integration of discrete topological spaces with graph theory, leading to the development of advanced neural network models for analyzing complex relational data. We introduce Discrete Topology and Graph Neural Networks (GNNs) as powerful tools for understanding graph structures. The core contribution is the novel Directed Simplicial Neural Network (Dir-SNN), a message-passing framework operating on directed simplicial complexes. Dir-SNNs leverage higher-order topological directionality defined via face maps, capturing rich structural patterns like directed flows and cycles that are lost in traditional undirected models. We demonstrate the superior expressivity of Dir-SNNs, proving they can distinguish non-isomorphic directed graphs that are indistinguishable by Directed Graph Neural Networks (Dir-GNNs). Numerical experiments on synthetic source localization and expressivity tasks validate the model's effectiveness and robustness, showing significant performance gains, particularly in noisy environments. This work bridges discrete topology and machine learning, offering a more nuanced approach to analyzing directed, higher-order interactions in complex systems.*

Keywords: *Discrete Topology, Graph Neural Networks, Directed Simplicial Complexes, Dir-SNN, Topological Directionality, Face Maps, Graph Discrimination, Source Localization, Learning on Structured Data.*

*Received 11 Dec., 2025; **R**evised 20 Dec., 2025; **A**ccepted 22 Dec., 2025 © The author(s) 2025.*
P**ublished with open access at **www.questjournas.org

I. Introduction

Topological spaces are fundamental concepts of pure mathematics, allowing them to be understood without relying on traditional geometric concepts such as measuring angles and distances. Among the most important and simplest of these spaces are discrete topological spaces, which are characterized by treating each point as an open set, which allows for the complete separation of each of its elements. Because of this fundamental importance, they serve as a powerful tool for analyzing systems, as they are clear in the arrangement and structure of points or nodes. Discrete topological spaces have many uses and are widely used as models to simplify and explore complex structures across different branches of mathematics, making them useful in fields such as set theory, algebra, and geometry. Discrete topological spaces are important in terms of understanding the relationship between elements and in a way. On the other hand, graph theory is another fundamental branch of mathematics that studies the relationship between entities represented by vertices (nodes) connected by edges (links). Graph theory has countless applications in the fields of networks, computer science, information systems, and cryptography. Discrete topological spaces enable clear distinction between vertices and the determination of the relationship between them in the graph, which facilitates the study and analysis of networks and communication patterns. Studies on topological spaces originated in the early twentieth century and were limited to solving problems related to geometry and differential equations.[5]

II. Graph Theory Basics

A graph is a mathematical abstraction that is widely useful in solving various types of problems. Fundamentally, a graph consists of a set of vertices (nodes) and a set of edges, where an edge connects two

vertices. Formally, a graph is defined as a pair G = (V, E), where VVV is a finite set of vertices, and E is a collection of edges, with each edge being a pair (u, v) where u, v∈V. This basic definition of a graph allows flexibility in its interpretation vertices could represent cities with roads as edges, or web pages with hyperlinks as connections.[2]

Several important types of graphs are frequently studied:

- Path, A simple graph in which the vertices can be arranged in such a way that two vertices are adjacent if and only if they are consecutive in the ordering.
- Undirected Graph**,** A graph in which each edge represents an unordered, transitive relationship between two nodes, typically depicted as a line or arc.
- Directed Graph (Digraph), A graph in which each edge represents an ordered, nontransitive relationship between two nodes, usually represented with an arrow indicating direction.

III. Discrete Topology and Graph Theory

With the fundamental definitions of discrete topology and graph theory established, it is essential to explore the body of research that has investigated the practical applications of these concepts. Numerous studies have explored how discrete topological spaces can enhance our understanding of graph structures, particularly in the context of complex networks and connectivity patterns. While discrete topology offers a clear way to distinguish between vertices in a graph, its application to more complex, multidimensional networks remain underexplored. Most previous studies have focused on simple, binary graph structures, often overlooking the potential of discrete topological spaces to provide deeper insights into dynamic and large-scale networks. In recent years, however, discrete topology has shown promise in analyzing complex systems in fields such as data science, network theory, and biological systems. The following sections will review key studies that have shaped the development of this interdisciplinary field, highlighted major findings and identified gaps in the current literature. These studies offer a foundation for future research, particularly in applying discrete topology to more sophisticated graph structures and networks. [3]

Definition of Learning on Graph Data (3.1) [2] We define the graph data as G = (V, E), with V representing the node set comprising |V | = N nodes. The edge set E ⊆ V × V represents the connections between nodes. The feature matrix $X = \{x_1, x_2, \ldots, x_N\}^T \in R^{N \times D}$ includes feature vectors x_i associated with node v_i, where D denotes the feature dimension. The adjacency matrix of G, denoted by $A \in R^{N \times N}$, sets $A_{ij} = 1$ for any existing edge $e_{i,j} \in E$ and $A_{ij} = 0$ otherwise. The normalized adjacency matrix is given by $\hat{A} = D^{-1/2}AD^{-1/2}$. The degree matrix D, being a diagonal matrix, is characterized by $D_{i,i} = \Sigma_j A_{i,j}$.

In the domain of graph learning, the node-level task stands out as a significant area of focus. The objective of this task is to forecast the properties (i.e., numerical value or probability) or class of the individual nodes. This process entails training a neural network model that utilizes a subset of nodes with known properties, denoted as V_L, to infer the properties of other unknown nodes. The essence of this training is encapsulated by optimizing the function:

$$\min_{\theta} L\left(f_\theta(G)\right) = \sum_{v_i \in VL} \ell\left(f_\theta(X, A)_i; y_i\right), \tag{3.1}$$

Here, the function $f_\theta(X, A)$ aims to forecast the property for each node, with y_i representing the actual state of node v_i. The discrepancy between the predicted and true properties is quantified using a loss function $\ell(\cdot, \cdot)$, such as RMSE (Root Mean Square Error).

IV. Graph Neural Networks

Over recent years, GNNs have garnered increasing interest and have been deployed across diverse fields, including bioinformatics, material science, chemistry, and neuroscience. Among them, Graph Convolutional Networks (GCN) and Graph Attention Networks (GAT), have advanced the frontier of research on graph-structured data with their sophisticated and effective designs. Typically, GNNs aim to learn graph representations, including node embeddings $h_i \in R^d$, by utilizing both the structural and feature information of a graph G. The process within a GNN involves two key operations: message passing and aggregation of neighborhood information. This involves each node in the graph repeatedly collecting and integrating information from its neighbors as well as its own attributes to enhance its representation. The operation of a GNN over L layers can be described by the following expression:

$$h_i^{(l+1)} = \sigma(h_i^{(l)}, AGG(h_j^{(l)}; j \in Ai)), \forall l \in [L], \tag{3.2}$$

where $h_i^{(l)}$ is the representation of node v_i at layer l, with $h_i^{(0)} = xi$ being the initial node features. Here, A_i represents the set of neighbors for node v_i, AGG(·) denotes a variant-specific aggregation function, and σ represents an activation function. Following the completion of L layers of message passing, the resultant node embedding h_i is passed through a projection function $F(h_i)$ to produce the output prediction $\hat{y}_i$. [1]

V. Mechanistic Models

Empirical models in epidemic forecasting utilize historical data to discern patterns and forecast the future spread of diseases. In contrast, mechanistic models provide a detailed framework that explores the biological and social complexities underlying the transmission of infectious diseases, thus exceeding the reliance on historical data inherent to empirical models. Among mechanistic approaches, classic compartmental models (e.g., SIR) are particularly notable. These models adeptly simplify the intricate dynamics of disease transmission into digestible components. This simplification facilitates a clearer understanding of infection spread, serving as a valuable tool for both researchers and policymakers.[1]

SIR Compartmental Model. In the domain of epidemiology, it is widely hypothesized that the rate at which networks evolve is significantly slower compared to the propagation speed of diseases. This fundamental assumption underpins the adoption of a SIR model, which is instrumental in accurately capturing the dynamics of epidemic spread. The SIR model categorizes the population into three distinct groups based on their disease status: susceptible (S) to infection, currently infectious (I), and recovered (R), with the latter group being immune to both contraction and transmission of the disease. The SIR model, formulated using ordinary differential equations (ODEs), are as follows:

$$\frac{dS(t)}{dt} = -\beta \frac{S(t)I(t)}{N}, \quad \frac{dI(t)}{dt} = \beta \frac{S(t)I(t)}{N} - \gamma I(t), \quad \frac{dR(t)}{dt} = \gamma I(t). \tag{3.3}$$

These equations distribute the total population N across the aforementioned categories, with the transitions between states regulated by two pivotal parameters: the transmission rate β ($S \rightarrow I$) and the recovery rate γ ($I \rightarrow R$). The model posits a quadratic relationship for disease transmission via interactions between susceptible and infectious individuals ($\beta S(t)I(t)$), alongside a linear recovery mechanism ($\gamma I(t)$). By fine-tuning the parameters of the SIR model, it is possible to compute the basic reproduction number
$R_0 = \beta/\gamma$, serving as a metric for the disease transmission potentials.

SIR Variants. While the SIR model provides a powerful framework for analyzing disease dynamics, its simplicity can sometimes neglect critical factors such as incubation periods, non-permanent immunity, and heterogeneous population interactions. This limitation has spurred the development of SIR variants, which offer a more comprehensive and nuanced understanding of the spread and control of infectious diseases. We briefly outline some of the most commonly used variants: i) SEIR: The SEIR model extends the basic SIR framework by incorporating an 'Exposed' (E) compartment. This compartment represents individuals who have been exposed to an infectious disease but are not yet infectious themselves. The model details the transition through the stages according to the sequence: [3]
$S \rightarrow E \rightarrow I \rightarrow R$. ii) SIRD: Enhancing the traditional SIR model, the SIRD variant includes a 'Dead' (D) compartment, thus adapting the progression to: $S \rightarrow I \rightarrow R \rightarrow D$. This modification accounts for individuals who succumb to the disease, providing a more accurate depiction of its mortality impact.

Table 3.1.: A brief description of epidemic tasks we categorized.

Tasks	Time Interval	Objective
Detection	History-Present	Incident Back-tracing
Surveillance	Present	Event Monitoring
Prediction	Future	Future Incident Prediction
Projection	Future	Change Simulation and Prediction

Directed Flag Complexes (5.1)[6] It is often useful to address (directed) graph-based problems by enriching the graph with higher-order relations, thereby mapping it to a higher-order simplicial complex while preserving its structure – nodes (0-simplices) and edges (1simplices) remain those of the underlying graph. When such a structure-preserving transformation maps distinguishable graphs to distinguishable simplicial complexes (formally, it preserves isomorphisms) and is injective, meaning it maps indistinguishable graphs to indistinguishable simplicial complexes, it is called a graph lifting. A prominent example is the flag complex lifting, which maps an undirected graph G to a flag complex $\widetilde{K}$ G. A flag complex is a simplicial complex where the k-simplices correspond to the (k +1)-cliques in G – subsets of k +1 vertices where each pair of distinct vertices is connected by an edge. Directed flag complex liftings extend this transformation to accommodate directed graphs (digraphs). In this case, a directed flag complex K_G is a directed simplicial complex where the k-simplices are

ordered (k+1)-cliques in G. In this context, an ordered k-clique is a totally ordered tuple $(v_1,...,v_k)$ such that (v_i,v_j) is a directed edge for $i < j$. A simple example showing a graph, its flag complex, a directed graph, and its directed flag complex is presented in Fig. 3.1.

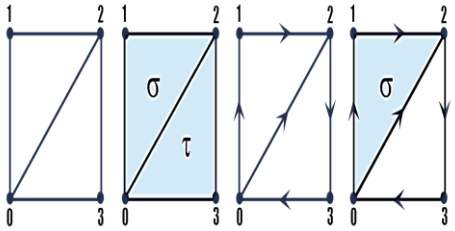

Fig. 3.1: From left to right: an undirected graph, its corresponding two-dimensional flag complex (nodes, edges, and triangles), a directed graph, and its corresponding directed flag complex (nodes, directed edges, and directed triangles). In the undirected flag complex, the triangles are $\sigma = \{0,1,2\}$ and $\tau = \{0,2,3\}$. In the directed flag complex, only the triangle $\sigma = (0,1,2)$ forms a directed triangle, as (0,2,3) lacks the required edge (0,3).

Face Maps (5.2) [6] In a directed simplicial complex K, if $\tau \subseteq \sigma$ and $\sigma \in K$, then τ is said to be a face of σ. Specifically, if $\dim(\tau) = \dim(\sigma)-1$, τ is called a facet of σ. A directed k-simplex has k +1 facets. Face maps are a formal tool used to identify the faces of a simplex by systematically removing one of its vertices. Let $K = (V, \Sigma)$ be a directed simplicial complex of dimension K. The face map d_i maps a k-simplex σ to the $(k-1)$-simplex τ_i obtained by omitting the i-th vertex from σ, thus it is a mapping $d_i : \Sigma \to \Sigma$ defined as:

$$d_i(\sigma) = \tau_i = \begin{cases} (v_0,..., \widetilde{v}_i,..., v_k) & \text{if } i < k, \\ (v_0,..., v_{k-1}, \widetilde{v}_k) & \text{if } i \geq k. \end{cases}, \tag{3.4}$$

where $\widetilde{v}_i$ means that the vertex in the i-th position has been removed. The resulting $(k-1)$-simplex τ_i preserves the original order of vertices, excluding the omitted vertex. For instance, applying the face maps to a directed triangle returns its directed edges. Face maps satisfy the simplicial identity $d_i \circ d_j = d_{j-1} \circ d_i$, where $\circ$ is the composition operator, for $i < j$, ensuring that the order in which vertices are removed does not affect the resulting face. Face maps are an essential tool for understanding and structuring the relationships between directed simplices within a complex.

Topological Signals (5.3) [6] Given a directed simplicial complex $K = (V, \Sigma)$, a topological signal over K is defined as a mapping $x : \Sigma \to R$ from the set of simplices Σ to real numbers. Therefore, the feature vectors $x_\sigma \in R^F$ and $x_\tau \in R^F$ of simplices σ and τ are a collection of F topological signals, i.e.,

$$x_\sigma = [x_1(\sigma),...,x_F(\sigma)]^\top \text{ and } x_\tau = [x_1(\tau),...,x_F(\tau)]^\top. \tag{3.5}$$

For example, in a directed simplicial complex of dimension two, there are F signals (features) associated with nodes, edges, and triangles.

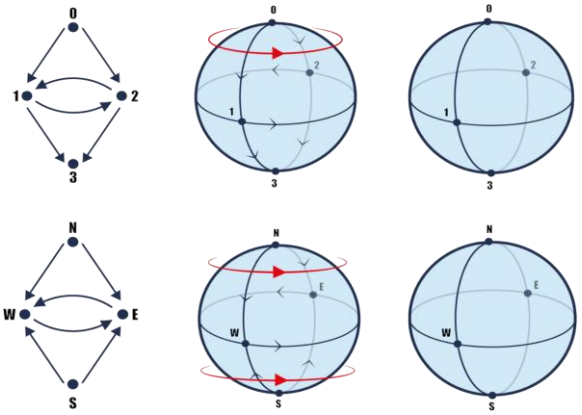

Fig. 3.2: (Left) A pair of non-isomorphic (distinguishable) digraphs, (Middle) their corresponding two-dimensional directed flag complexes, and (Right) their symmetrized undirected versions. Symmetrizing the directed flag complexes collapses the non-isomorphic digraphs into isomorphic flag complexes, while the directed flag complexes are non-isomorphic.

Direct Simplicial Neural Networks (5.4) [6] We introduce Directed Simplicial Neural Networks (Dir-SNNs), message-passing networks operating on directed simplicial complexes. To do so, we first formally motivate and then introduce a consistent notion of higher-order topological directionality.

Motivation (5.5) [6] A directed simplicial complex can be transformed into an undirected simplicial complex via symmetrization, wherein the order of the vertices in each directed simplex is disregarded. The symmetrization preserves isomorphisms, meaning that two isomorphic directed simplicial complexes will remain isomorphic as undirected complexes after symmetrization. Informally, this means that two indistinguishable directed simplicial complexes remain indistinguishable after the symmetrization. However, the process is not injective: non-isomorphic directed simplicial complexes may be mapped to isomorphic undirected simplicial complexes. Informally, this means that two distinguishable directed simplicial complexes can become indistinguishable after the symmetrization. For the same reason, composing symmetrization with lifting into (directed) flag complexes – whether by first symmetrizing a digraph and then lifting it into a flag complex, or by first lifting it into a directed flag complex and then symmetrizing it – can collapse distinct digraphs into the same undirected simplicial complex. This is just one of the possible formal hints showing that transitioning to higher-order undirected simplicial complexes is not always inherently beneficial, calling for a notion of higher-order directionality. Fig. 3.2 illustrates this fact. Higher-order Topological Directionality. We define directed relations among simplices in a directed simplicial complex K using face maps. Consider an ordered pair of simplices (σ, τ) with dim(σ) = dim(τ). Let (d_i, d_j) denote an ordered pair of the i-th and j-th face maps, as defined in (3.1).

We define (σ, τ) as being down (k, i, j)-adjacent if there exists a simplex κ such that dim(κ) = dim(σ) − k and $d_i(\sigma) \supseteq \kappa \subseteq d_j(\tau)$. To illustrate this, let σ = (0,1,2) and τ = (1,2,3) be 2-simplices (triangles) as in Fig. 3.3 Consider the ordered pair of face maps (d_0, d_2). Applying these maps, we obtain the directed edges $d_0((0,1,2)) = (1,2) = \kappa$ and $d_2((1,2,3)) = (1,2) = \kappa$, making the ordered pair of simplices (σ, τ) down (1,0,2)-adjacent. Similarly, we define (σ, τ) as being up (k, i, j)-adjacent if there exists a simplex κ such that dim(κ) = dim(σ) + k and $\sigma \subseteq d_i(\kappa)$ and $\tau \subseteq d_j(\kappa)$. Notably, for i ≠ j, if an ordered pair (σ, τ) is up/down (k, I ,j)-adjacent,

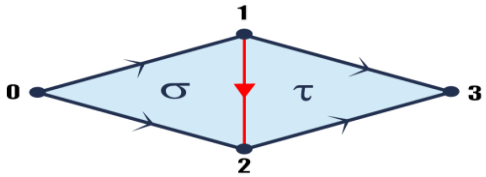

Fig. 3.3: Example of (1,0,2)-adjacent directed 2-simplices σ = (0,1,2) and τ = (1,2,3), sharing the edge κ = (1,2) (in red).

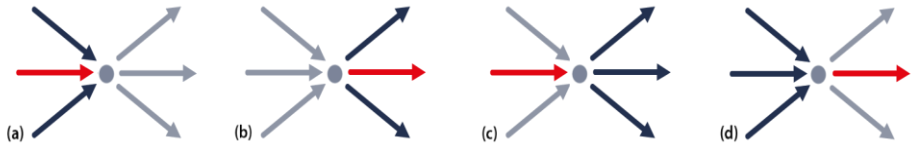

Fig. 3.4: Examples of edge down adjacencies. σ is the red edge in each subfigure. (a)/(b) $A^{0,0}_{\downarrow,1} / A^{1,1}_{\downarrow,1}$ connects σ with the edges with which it shares a target/source node; (c)/(d) $A^{0,1}_{\downarrow,1} / A^{1,0}_{\downarrow,1}$ connects σ with the edges whose source/target node is the target/source node of σ then (τ, σ) is up/down (k, j, i)-adjacent. Additionally, due to the symmetry in the face maps, if (σ, τ) is up/down (k, i, i)-adjacent, then (τ, σ) is also up/down (k, i, i)-adjacent, showing that not all the directed adjacencies are necessarily asymmetric. It is then natural to define notions of neighborhood among simplices of the same dimensions using the above adjacencies.

We define the lower (k, i, j)-adjacency $A^{i,j}_{\downarrow,k}$ of a simplex σ as

$$A^{i,j}_{\downarrow,k}(\sigma) = \left\{ \begin{array}{c} \dim(\sigma) = \dim(\tau), \\ \tau \in \Sigma \mid \exists \kappa : \dim(\kappa) = \dim(\sigma) - k, \\ d_i(\sigma) \supseteq \kappa \subseteq d_j(\tau) \end{array} \right\}, \qquad (3.7)$$

We define the upper (k, i, j)-adjacency $A^{i,j}_{\uparrow,k}$ of a simplex σ as

$$A^{i,j}_{\uparrow,k}(\sigma) = \left\{ \begin{array}{c} \dim(\sigma) = \dim(\tau), \\ \tau \in \Sigma \mid \exists \kappa : \dim(\kappa) = \dim(\sigma) + k, \\ \sigma \subseteq d_i(\sigma) \text{and } \tau \subseteq d_j(\kappa) \end{array} \right\}, \quad (3.8)$$

For the lower (k, i, j)-adjacency in (3.7) we assume dim(κ) = 0 if k > dim(σ), while for the upper (k, i, j)-adjacency in (3.8) we assume dim(κ) = dim(K) if
k > dim(K) − dim(σ). We show some examples of edge lower and upper adjacencies in Fig. 3.4 and Fig. 3.5, respectively. The upper (k, i, j)-adjacency offers a complementary perspective to the lower (k, i, j)-adjacency, because the former captures the directed interactions where simplices σ and τ are both included in some higher-order simplices, while the latter when they both contain some lower-order simplices. Finally, face maps can also be used to define notions of neighborhood among simplices of different dimensions. We define the boundary B and coboundary C of σ ∈ K as

$$B(\sigma) = \bigcup_{i=0}^{\dim(\sigma)} \{d_i(\sigma)\}, \quad C(\sigma) = \bigcup_{i=0}^{\dim(\sigma)+1} d_i^{-1}(\sigma), \quad (3.9)$$

where d_i^{-1} is the preimage of d_i. The boundary and the coboundary of σ are then its facets and the simplices it is a facet of, respectively.

Consistency of Higher-order Topological Directionality (5.6) [7] The way we define the directed adjacencies among simplices is grounded in the notion of simplicial directed path. In a digraph G = (V, E), a directed path is defined as a sequence of vertices $(v_0, v_1, ..., v_n)$ where each consecutive pair
$(v_i, v_{i+1}) \in E$ forms a directed edge. In a directed simplicial complex K,
a (k, i, j)-simplicial path between an ordered pair of simplices (σ, τ) in K is a sequence of simplices $\sigma = \alpha_0, \alpha_1, \alpha_2, ..., \alpha_n, \alpha_{n+1} = \tau$ such that each consecutive pair (α_k, α_{k+1}) is (k, i, j)-adjacent along the face maps (d_i, d_j). We show some examples of simplicial paths of triangles in Fig. 3.6. Higher order directionality reveals novel, discriminative structural properties, as demonstrated again in Fig. 3.2, where circular flows (in red) emerge considering of (1,0,2)-simplicial paths. Finally, simplicial paths can also traverse simplices of different orders. We decided to add the constraint
dim(σ) = dim(τ) in (3)-(4) to keep a distinction between same-dimension and different-dimension neighbors.

Fig. 3.5: Examples of edges upper adjacencies. σ is the red edge in
each subfigure. (a) $A^{2}{}_{\uparrow,}{}^{0}{}_{1}$ connects σ with the edge on its right; (a) $A^{2}{}_{\uparrow,}{}^{1}{}_{1}$ connects σ with the edge on its left.

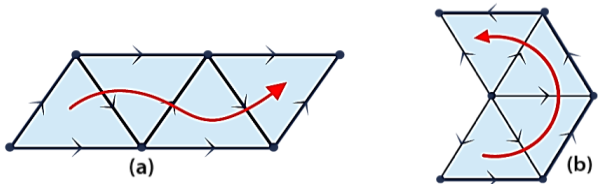

Fig. 3.6: Examples of simplicial paths (in red) of 2-simplices (triangles). (a) The (1,0,2) path, showing the simplices are Equi directed; (b) the (1,1,2) path, revealing a circular flow around a source node.

Directed Simplicial Neural Networks (5.7) [7] Directed Simplicial Neural Networks (Dir-SNNs) are message-passing networks leveraging the adjacencies from (3.3.3)-(3.3.4). Given (i) a directed simplicial complex K, and (ii) sets $\{A^{ij}_{\downarrow,k}\}_{i,j,k}$ and $\{A^{ij}_{\uparrow,k}\}_{i,j,k}$ collecting some of the lower and upper (k,i,j)-adjacencies as in (3)-(4), respectively, the l-th layer of a Dir-SNN updates the feature vector $\mathbf{x}^l_\sigma$ of a σ ∈ K as

$$m^{l+1}_{\sigma,\downarrow ijk} = \bigoplus_{\tau \in A^{ij}_{\downarrow,k}(\sigma)} \psi_{A^{ij}_{\downarrow,k}}\left(x^l_\sigma, x^l_\tau, x^l_\kappa\right), \quad (3.10)$$

$$m^{l+1}_{\sigma,\uparrow ijk} = \bigoplus_{\tau \in A^{ij}_{\uparrow,k}(\sigma)} \psi_{A^{ij}_{\uparrow,k}}\left(x^l_\sigma, x^l_\tau, x^l_\kappa\right), \quad (3.11)$$

$$m_{\sigma,\beta}^{l+1} = \bigoplus_{\tau \in \beta(\sigma)} \psi_{\beta}\left(x_{\sigma}^{l}, x_{\tau}^{l}\right), \tag{3.12}$$

$$m_{\sigma,\beta}^{l+1} C = \bigoplus_{\tau \in C(\sigma)} \psi_{\beta}\left(x_{\sigma}^{l}, x_{\tau}^{l}\right), \tag{3.13}$$

$$x_{\sigma}^{l+1} = \phi\left(x_{\sigma}^{l}, \left\{m_{\sigma,\downarrow ijk}^{l+1}\right\}_{ijk}, \left\{m_{\sigma,\uparrow ijk}^{l+1}\right\}_{ijk}, m_{\sigma,\beta}^{l+1}, m_{\sigma,C}^{l+1}\right). \tag{3.14}$$

with κ in (3.10) as in (3.7) and in (3.11) as in (4), $\oplus$ being a intra-neighborhood aggregator. The neighborhood-dependent message functions $\psi_{A_{\downarrow,k}^{ij}}$, $\psi_{A_{\uparrow k}^{ij}}$, ψ_B and ψ_C, and the update function ϕ are learnable functions. In other words, the feature vector of a simplex is updated in a learnable fashion through aggregated messages with its neighboring simplices. At the l-th layer, a simplex has collected information from simplices that are up to l steps away from it along the simplicial directed paths induced by the chosen adjacencies.

Remark (5.8)[7] In (3.12) and (3.14), single message functions ψ_B and ψ_C are used for computational efficiency. However, face map- and preimage dependent message functions, i.e. ψ_{di} and $\psi_{d^{-1}}$, could leverage i more fine-grained directed information, e.g. edges communicating with their source or target nodes using different sets of weights.

Expressiveness of Dir-SNNs (5.9)[8] The expressive power of topological neural networks (including GNNs) is usually measured by their capacity to distinguish non-isomorphic objects within their underlying domain. Here, we introduce the following result:

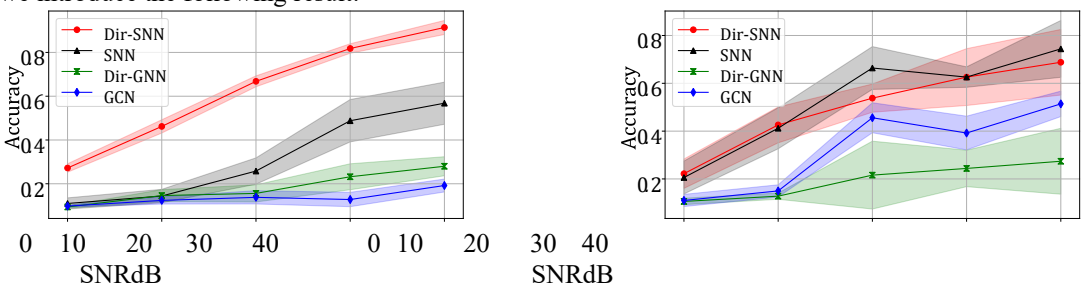

Fig. 3.7: SNR vs accuracy of directed and undirected TNNs and GNNs

on directed (left) and undirected (right) synthetic flag complexes

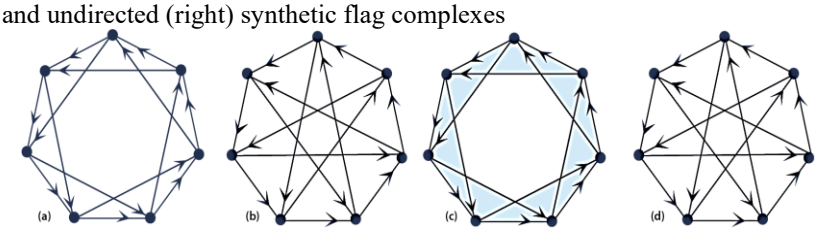

Fig. 3.8: A pair of non-isomorphic directed graphs (a) and (b),

along with their corresponding flag complexes (c) and (d). These digraphs can be distinguished by Dir-SNNs but not by Dir-GNNs

An example is shown in Fig. 3.3.8. Intuitively, our proof follows the same strategy, thus it relies on the definition of a directed simplicial isomorphism test that is proven to be (i) an upper bound on the expressiveness of Dir-SNNs, and (ii) more powerful than the directed graph isomorphism test, being an upper bound on the expressiveness of Dir-GNNs [20].

Remark (5.10) [8] Dir-SNNs generalize Dir-GNNs. A Dir-GNN is a Dir SNNs operating on a digraph, i.e., a directed simplicial complex of dimension one, which updates the node feature vectors using only the (1,0,1)- and the (1,1,0)-upper adjacencies in (3.8).

VI. Numerical Results

We first provide some preliminary results to validate the effectiveness of Dir-SNN on a synthetic source localization task at the edge level.

A. Source Localization

Dataset. We generate directed and undirected graphs following a Stochastic Block Model. Each graph has 70 nodes uniformly divided into 10 communities, with intra- and inter-community edge probabilities of 0.9 and 0.01, respectively. Intra-community edges are grouped into 10 edge communities, and the remaining intercommunity edges form an 11th partition. We generate 1000 edge signals from a zero-mean Gaussian distribution with variance $1/N_{edges}$, and then, for each signal, we introduce spikes to randomly selected source edges belonging to a single community with intensity
$\alpha \sim N(0,1)$. In the directed case, the graphs are lifted in their corresponding directed flag complexes, and the spikes are diffused over the graph following $x' = S^t x + n$, where $S \in R^{Nedges \times Nedges}$ is a non-symmetric binary matrix encoding $\mathcal{A}^{0,1}_{\downarrow,1}$ from (3) for each edge, t is the order of diffusion sampled from a Student-T distribution with 10 degrees of freedom and capped at 100, x is the original signal with the added spikes and n is additive white Gaussian noise inducing a specific SNR. In the undirected case, the graphs are lifted in their corresponding flag complexes, and S is the symmetric lower edge adjacency, analogously. The task is to identify the community originating the spikes, thus a classification problem with 11 classes. Experimental setup. Since the designed task considers only edge signals, we decided to employ a specific instance of Dir-SNN that TABLE I: Discrimination accuracies of Dir-GNN and Dir-SNN. operates on the four directed adjacency relations described by $A^0{}_{\downarrow'}{}^0{}_1$, $\mathcal{A}^{0,1}_{\downarrow,1}$, $\mathcal{A}^{1,0}_{\downarrow,1}$, $\mathcal{A}^{1,1}_{\downarrow,1}$, without considering boundary and coboundary. The aggregators, update, and message functions are chosen to have a convolutional architecture. Therefore, we compare with an undirected convolutional SNN using the undirected lower adjacency of the edges, and with two GNNs that operate on the projection of edge features on their endpoint nodes, one directed (Dir-GNN) and the other undirected (GCN [32]). We perform a grid search for all the models among the following values: {1,2,3} layers of size {16,32,64}. The embeddings are max-pooled and fed into an MLP to perform classification. We average over 5 seeds. Discussion. In Fig. 3.3.7, Dir-SNN consistently and largely outperforms all the baselines for all levels of SNR on directed flag complexes (Left). Moreover, Dir-SNN performs comparably to SNN on flag complexes (Right), showing their robustness to model mismatching.[8]

B. Expressivity Validation

Dataset. We employ a toy dataset containing only the two graphs in Fig. 8 (a)-(b), and we assign them two different classes. The task is then to correctly classify them.

Experimental Setup. We compare Dir-SNN and Dir-GNN. The parameters are the same of IV-A, but for Dir-SNN we also use $\mathcal{A}^{2,0}_{\uparrow,1}$(to leverage the directed triangles). The two graphs in Fig. 3.8 (a)-(b) are fed to the Dir-GNN with constant features on each node, while the two corresponding directed flag complexes in Fig. 3.8 (c)-(d) are fed to the Dir-SNN with constant features on each simplex. Discussion. The results in Table 6.1 numerically validate the improved expressivity of Dir-SNN. Indeed, Dir-GNN cannot discriminate the two graphs, while Dir-SNN can discriminate them thanks to the lifting into directed flag complexes.[9]

Table 6.1: Discrimination accuracies of Dir-GNN and Dir-SNN.

Model	Accuracy
Dir-GNN	50%
Dir-SNN	100%

VII. Conclusion:

This paper presents an advanced framework integrating discrete topology with graph neural networks by introducing the Directed Simplicial Neural Network (Dir-SNN), which operates by lifting directed graph data into directed simplicial complexes to capture higher-order interactions. The model leverages a novel notion of "topological directionality" defined via face maps to establish directed adjacency relations, enabling the detection of complex structural patterns such as circular flows. The model demonstrates superior theoretical and practical expressivity, successfully distinguishing non-isomorphic directed graphs that traditional models fail to discriminate, while also exhibiting robust and outstanding performance in practical tasks like source localization under noisy conditions, establishing it as a powerful tool for analyzing complex networks and directed interactions.

References

[1]. Adams H, Emerson T, Kirby M, Neville R, Peterson C, Shipman P, et al. Persistence images: A stable vector representation of persistent homology. Journal of Machine Learning Research. 2017;18(8):1-35.
[2]. Bauer U. Ripser: A lean C++ code for the computation of Vietoris-Rips persistence barcodes. SoftwareX. 2021;14:100730.
[3]. Bronstein MM, Bruna J, LeCun Y, Szlam A, Vandergheynst P. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. Nature Reviews Physics. 2021;3(6):422-440.

[4]. Bubenik P. Statistical topological data analysis using persistence landscapes. Journal of Machine Learning Research. 2015;16(1):77-102.
[5]. Carlsson G. Topology and data. Bulletin of the American Mathematical Society. 2009;46(2):255-308. 6.
[6]. Carlsson G, De Silva V. Zigzag persistence. Foundations of 2010;10(4):367-405. Computational Mathematics.
[7]. Carlsson G, Zomorodian A. The theory of multidimensional persistence. Discrete & Computational Geometry. 2009;42(1):71-93.
[8]. Choudhary A, Lin Y, Mischaikow K, Wang B. A GPU accelerated implementation of persistent homology. ACM Transactions on Mathematical Software. 2021;47(4):1-24.
[9]. Cohen-Steiner D, Edelsbrunner H, Harer J. Stability of persistence diagrams. Discrete & Computational Geometry. 2007;37(1):103-120.
[10]. Dey TK, Memoli F, Wang Y. Multiscale Mapper: Topological summarization via codomain covers. ACM Transactions on Graphics. 2020;39(4):61:1-61:14.