Unsupervised Parameter-free Simplicial Representation Learning with Scattering Transforms

Hiren Madhu [*1] **Sravanthi Gurugubelli** [*1] **Sundeep Prabhakar Chepuri** [1]

Abstract

Simplicial neural network models are becoming popular for processing and analyzing higher-order graph data, but they suffer from high training complexity and dependence on task-specific labels. To address these challenges, we propose simplicial scattering networks (SSNs), a parameter-free model inspired by scattering transforms designed to extract task-agnostic features from simplicial complex data without labels in a principled manner. Specifically, we propose a simplicial scattering transform based on random walk matrices for various adjacencies underlying a simplicial complex. We then use the simplicial scattering transform to construct a deep filter bank network that captures high-frequency information at multiple scales. The proposed simplicial scattering transform possesses properties such as permutation invariance, robustness to perturbations, and expressivity. We theoretically prove that including higher-order information improves the robustness of SSNs to perturbations. Empirical evaluations demonstrate that SSNs outperform existing simplicial or graph neural models in many tasks like node classification, simplicial closure, graph classification, trajectory prediction, and simplex prediction while being computationally efficient.

1. Introduction

Higher-order interactions between more than two entities occur naturally in various fields, such as biochemistry, social networks, and coauthorship networks (Ilya et al., 2020). A simplicial complex is a mathematical structure that captures not only pairwise interactions between entities but also higher-order interactions among groups of entities (Schaub et al., 2020). Such higher-order interactions occur naturally in social networks, email threads, and collaboration networks, to name a few. While a graph can only represent a pairwise friendship or a pairwise communication, simplicial complexes can model friend circles or group emails. A simplicial complex is a collection of simplices of different orders, namely, nodes, edges, triangles, and so on. Unlike in graphs, where an entity only has adjacency based on edges, a simplex in a simplicial complex has multiple adjacencies, allowing for a richer representation.

Building upon the success of graph neural networks (GNNs) (Kipf & Welling, 2017; Hamilton et al., 2017; Xu et al., 2018) for analyzing graph-structured data, simplicial neural networks (SNNs) are becoming popular for learning representation from higher order simplices (Bodnar et al., 2021; Yang et al., 2022; Roddenberry et al., 2021). These neural networks overcome the limitations of GNNs by processing higher-order interactions, thereby enriching the function space learnt by the neural model and, in turn, increasing the expressivity of representations (Bodnar et al., 2021). Bodnar et al. (2021) also shows that MPSN outperforms graph-based neural models by clique lifting graph data as this enables the model to increase the dimension of the data and learning representations that are more expressive than graph-based models. However, the requirement to aggregate information from various types of neighborhoods and different orders of simplices significantly increases the number of trainable parameters involved in SNNs, increasing its training complexity. Not only does this increase training time, but, as we show later on, it also requires substantial training data, without which the performance of SNNs is compromised. Another significant challenge in training the parameters of SNN model is the need for task-specific labels. It is hard to accurately label real-world, high-dimensional, and complex data because of its complicated structures, multiple true labeling schemes, and concerns about privacy.

Addressing these challenges and drawbacks of SNNs, we propose a parameter-free model (i.e., a model without any learnable parameters) to obtain topology preserving simplicial representations in a principled manner. To do so, we draw inspiration from scattering transforms (Mallat, 2012; Bruna & Mallat, 2013a;b). In particular, we propose a

[*]Equal contribution [1]Indian Institute of Science, India. Correspondence to: Hiren Madhu <hirenmadhu@iisc.ac.in>, Sravanthi Gurugubelli <sravanthig@iisc.ac.in>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

scattering transform-based model to efficiently extract task-agnostic features from simplicial complex data without labels. Scattering transforms extract universal features using a deep filter bank, which is designed, but not trained using any labels. Scattering transform-based models are known for their ability to extract representations with specific characteristics, such as invariance to predetermined operations (i.e., node permutation), nonexpansiveness, robustness (or, stability) to perturbations, and expressivity. While invariance, non-expansiveness, and stability of the scattering transform can be established theoretically, its expressivity is shown through empirical evidence (Bruna & Mallat, 2013a; Andén & Mallat, 2014; Bruna & Mallat, 2013b; Angles & Mallat, 2018). Although these methods are effective, they are developed for Euclidean data, are more suited to image and audio data, and do not account for relationships between entities in a dataset. Extending scattering transform to graph-structured data, Gama et al. (2019b); Gao et al. (2019); Zou & Lerman (2020) introduce geometric scattering based on diffusion transforms (Coifman & Maggioni, 2006) and spectral transforms (Hammond et al., 2009). Although geometric scattering accounts for the interactions between entities, similar to GNNs, it cannot capture the higher-order interactions that exist between a set of entities.

In this work, we propose a simplicial scattering network (SSN) to account for higher-order interactions that are present in simplicial complexes. We show that SSN preserves the characteristics of Euclidean and geometric scattering transform: invariance to permutations in simplicial complexes [cf. Theorem 5.1], non-expansiveness [cf. Theorem 5.2], robustness to perturbations [cf. Theorem 5.4], and expressivity through empirical evidence [cf. Section 6]. Furthermore, we also show SSN extracts these features while being computationally efficient [cf. Figure 2]. Our main contributions are as follows:

- We propose random walk matrices to capture different types of adjacency-based transition probabilities among simplices in a simplicial complex. Unlike the square random walk matrix used in geometric scattering, adjacency matrices of a simplicial complex are not square and cannot be raised to integer powers. To overcome this, we model the random walk matrices as diffusion transforms on simplicial data through an iterative process to formulate scattering transforms that can capture high-frequency information across multiple scales. By stacking multiple layers of simplicial scattering transforms, we construct a simplicial scattering network (SSN), which is interpretable as it follows a principled approach to extract simplex representations across varying frequency scales.
- Our approach to defining simplicial scattering transforms, as mentioned earlier, involves an iterative process for random walk-based diffusion of simplicial data from various types of neighborhoods, which differs from the traditional method of taking powers of random walk matrices of graphs in geometric scattering. Despite this difference, we theoretically demonstrate that the specific design of SSN retains all the characteristics of scattering transform, such as permutation equivariance, non-expansiveness, and robustness to structural perturbations. We theoretically prove the advantage of incorporating higher-order information in extracting representations of k-simplices. Specifically, we provide theoretical evidence showing that including higher-order information improves the robustness of the model to perturbations.

We evaluate the SSN model on various tasks such as node classification, graph classification, simplicial closure, trajectory prediction, and simplex prediction. We demonstrate that SSN performs better when incorporating boundary and coboundary information rather than solely using upper and lower adjacency information. We observe that SSN is superior to graph-based models by effectively capturing higher-order information. SSN consistently outperforms or performs on par with SNNs. Particularly, SSN excels in most tasks as real-world labeled data is often insufficient for training complex SNN models, and many tasks require a global view of the simplicial complex structure that SNNs fail to capture. Importantly, SSN achieves these results with much better time complexity, up to hundreds of times faster than SNNs [cf. Fig. 2].

2. Geometric Scattering

As SSN is inspired by the layered geometric scattering network, we briefly describe it in this section.

Diffusion-based geometric scattering transforms are based on diffusion wavelets (Hammond et al., 2009). The wavelet operators for graphs are constructed using the lazy random walk matrix $\mathbf{P} = \frac{1}{2}(\mathbf{I}_N + \mathbf{A}\mathbf{D}^{-1}) \in \mathbb{R}^{N\times N}$, where N is the number of nodes in the graph with $\mathbf{A}$ and $\mathbf{D}$ being its adjacency and degree matrices, respectively. When this random walk matrix is applied to a graph signal $\mathbf{x} \in \mathbb{R}^N$ as $\mathbf{P}\mathbf{x}$, the value $\mathbf{x}(v)$ for a node v is replaced with a localized average of the neighboring nodes of v.

Diffusion operators act as low-pass filters when applied to a graph signal $\mathbf{x} \in \mathbb{R}^N$, which may lead to information loss. To retain high-frequency information, wavelet matrices $\psi_j = \mathbf{P}^{2^{j-1}} - \mathbf{P}^{2^j}$ are used at different scales $j = 1, 2, \ldots, J$. These matrices are sensitive to high-frequency transitions of features at 2^j hops. A nonlinear modulus operator is then applied to the wavelet transformed signal, resulting in an operation represented as $|\psi_j \mathbf{x}|$. Finally, these transforms are applied in a layered manner, forming a deep hierarchical representation with each layer

capturing different features. The layers of the geometric scattering transform can be mathematically represented as

$$\begin{aligned} &S\mathbf{x} = f(\mathbf{x}) \\ &S|\psi_j \mathbf{x}| = f(|\psi_j \mathbf{x}|), \text{ for } 1 \leq j \leq J \\ &S|\psi_{j'}|\psi_j \mathbf{x}|| = f(|\psi_{j'}|\psi_j \mathbf{x}||), \text{ for } 1 \leq j \leq j' \leq J, \end{aligned} \quad (1)$$

where, S is the scattering transform, f aggregates transformed features, and j and j' denote scales. The function f is defined as a weighted sum or a simple summation operator in Gama et al. (2019b) and Gao et al. (2019), respectively. The resulting features $S|\psi_j \mathbf{x}|$ and $S|\psi_{j'}|\psi_j \mathbf{x}||$ are then concatenated to form an embedding vector for the graph.

Theoretical results from Zou & Lerman (2020) and Gama et al. (2019a) indicate the robustness of geometric scattering transforms to perturbations in the graph structure. However, these transforms are computed on graphs and only capture pairwise interactions.

3. Simplicial Complexes

In this section, we describe a simplicial complex and the types of adjacency matrices that represent a simplicial complex.

Consider a set of nodes denoted by $\mathcal{V}$. A k-simplex, represented as σ_k, refers to a simplex of order k and is a subset of $\mathcal{V}$ consisting of $k+1$ elements. A simplicial complex $\mathcal{X}$, comprises a finite set of simplices. Simplicial complexes satisfy the principle of inclusion, i.e., if σ_k is a part of $\mathcal{X}$, then all non-empty subsets of σ_k are also included in $\mathcal{X}$. The order, K, of $\mathcal{X}$ is determined by the highest-ordered simplex that it contains.

A k-simplex σ_k has four distinct types of neighbors in a simplicial complex: lower, upper, boundary, and coboundary adjacent neighbors. The lower adjacent neighbors represent simplices of the same order as σ_k that share a common $(k-1)$-simplex. Upper adjacent neighbors are simplices of the same order as σ_k sharing a common $(k+1)$-simplex. The boundary adjacent neighbors represent all the $(k-1)$-simplices that are faces of σ_k. The coboundary adjacent neighbors are all the $(k+1)$-simplices for which σ_k is a face. The incidence relationships between k-simplices and $(k-1)$-simplices are captured by the boundary matrix of order k and is denoted by $\mathbf{B}_k \in \mathbb{R}^{N_{k-1} \times N_k}$, where N_k is the number of simplices of order k. Specifically, the (i,j)th entry of the boundary matrix $\mathbf{B}_k$ is non-zero if the ith $(k-1)$-simplex forms a boundary of the jth k-simplex. To model the orientation of simplices in $\mathbf{B}_k$, the non-zero entries in $\mathbf{B}_k$ can take values of either $+1$ or -1, reflecting the relative orientations between the ith $(k-1)$-simplex and the jth k-simplex. However, for unoriented simplices, unoriented incidence matrices are used where the (i,j)th entry is 1 if the ith $(k-1)$-simplex is a boundary of the jth k-simplex and 0, otherwise. The upper and lower adjacencies between k-simplices can be represented using the incidence matrices as $\mathbf{B}_{k+1}\mathbf{B}_{k+1}^{\mathrm{T}}$ and $\mathbf{B}_k^{\mathrm{T}}\mathbf{B}_k$, respectively.

We next introduce the proposed simplicial scattering transforms in the following section.

4. Simplicial Scattering Network

Consider a simplicial complex of order K, denoted as $\mathcal{X}$. Let us collect the feature set of all simplices of all orders in $\mathbf{X} = \{\mathbf{X}_0, \mathbf{X}_1, \ldots, \mathbf{X}_K\}$, where $\mathbf{X}_k \in \mathbb{R}^{N_k \times D_k}$, is the matrix that is formed by stacking the features of all k-simplices. Here, D_k denotes the dimension of the feature space for $k = 0, \ldots, K$. Furthermore, the set of all the boundary matrices of every order is collected as $\mathbf{B} = \{\mathbf{B}_1, \ldots, \mathbf{B}_K\}$.

In what follows, we discuss the different components in a simplicial scattering transform pipeline. Figure 1 illustrates these different components.

Simplicial random walks. Let us introduce the concept of random walk matrices, which are crucial in understanding the interactions between the different elements in $\mathcal{X}$. These matrices represent the probability of transition between different simplices based on their adjacency relationships. Specifically, we define four types of random walk matrices, denoted as $\mathbf{P}_k^{\mathcal{C}} \in \mathbb{R}^{N_k \times N_{k+1}}$, $\mathbf{P}_k^{\mathcal{B}} \in \mathbb{R}^{N_k \times N_{k-1}}$, $\mathbf{P}_k^{\mathcal{L}} \in \mathbb{R}^{N_k \times N_k}$, and $\mathbf{P}_k^{\mathcal{U}} \in \mathbb{R}^{N_k \times N_k}$, corresponding to the coboundary, boundary, lower, and upper adjacencies of k-simplices, respectively. These matrices are defined as

$$\mathbf{P}_{k+1}^{\mathcal{C}} = \frac{1}{5}(\mathbf{D}_{k+1}^{\mathcal{B}})^{-1}\mathbf{B}_{k+1}, \;\; \mathbf{P}_k^{\mathcal{B}} = \frac{1}{5}\mathbf{B}_k^{\mathrm{T}}(\mathbf{D}_k^{\mathcal{B}})^{-1},$$

$$\mathbf{P}_k^{\mathcal{L}} = \frac{1}{5}\mathbf{B}_k^T\mathbf{B}_k(\mathbf{D}_k^{\mathcal{L}})^{-1}, \text{ and } \mathbf{P}_k^{\mathcal{U}} = \frac{1}{5}\mathbf{B}_{k+1}\mathbf{B}_{k+1}^T(\mathbf{D}_k^{\mathcal{U}})^{-1}.$$

Here, $\mathbf{D}_k^{\mathcal{B}} = \mathrm{diag}(\mathbf{B}_k\mathbf{1})$, $\mathbf{D}_k^{\mathcal{L}} = \mathrm{diag}(\mathbf{B}_k^T\mathbf{B}_k\mathbf{1})$, and $\mathbf{D}_k^{\mathcal{U}} = \mathrm{diag}(\mathbf{B}_{k+1}\mathbf{B}_{k+1}^T\mathbf{1})$. The normalization of the kth order boundary matrices by $(\mathbf{D}_k^{\mathcal{B}})^{-1}$ ensures that the transition probability from a k-simplex to a $(k-1)$-simplex is adjusted based on the number of k-simplices that each $(k-1)$-simplex is connected to. This prevents $(k-1)$-simplices connected to many k-simplices from dominating the random walk while ensuring that those with fewer connections are adequately represented. Similarly, the normalization of the kth order lower (or upper, respectively) adjacency matrices by $(\mathbf{D}_k^{\mathcal{L}})^{-1}$ ensures a balanced transition between k-simplices that share a $(k-1)$-simplex (or $(k-1)$-simplex). This normalization prevents k-simplices that share many $(k-1)$-simplices (or $(k-1)$-simplices) with other k-simplices from dominating the random walk. In addition to the previously defined random walk matrices, we also introduce a random walk matrix $\mathbf{P}_k^{\mathcal{S}} = \frac{1}{5}\mathbf{I}_{N_k} \in \mathbb{R}^{N_k \times N_k}$ that corresponds to staying at the same simplex, where $\mathbf{I}$ is the identity matrix. Given that we have four types of adja-

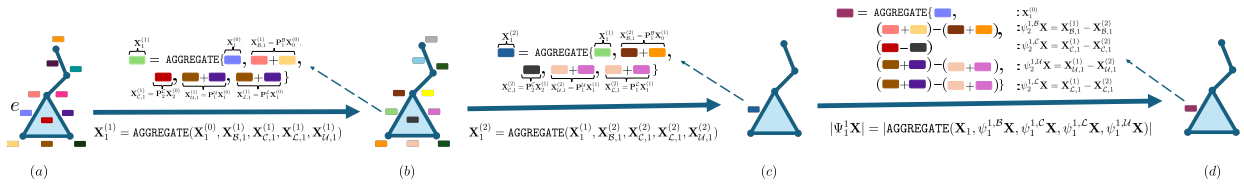

Figure 1: Pipeline for embedding generation using simplicial scattering transform: (a) Initial features over nodes, edges, and triangles in the complex. (b) Message propagation for all simplices from 1-hop neighbors. (c) Message propagation for edge e from 2-hop neighbors. (d) Generation of simplicial scattering-based embedding for edge e at scale $j = 1$.

cencies and an additional possibility of staying at the same simplex, we distribute the total probability equally among these five possibilities. Therefore, we scale all the random walk matrices by a factor of $\frac{1}{5}$, ensuring that the sum of the probabilities across all possible movements (including staying put) equals 1. This scaling maintains the probabilistic interpretation of the random walk matrices.

The boundary random walk matrices, denoted as $\mathbf{P}_k^{\mathcal{B}}$, are not square matrices, which inhibits the calculation of their powers, as opposed to geometric scattering where the lazy random walk matrices are square matrices. To address this, we propose a series of transformations:

$$\mathbf{X}_{\mathcal{B},k}^{(j)} = \mathbf{P}_k^{\mathcal{B}}\mathbf{X}_{k-1}^{(j-1)}, \qquad \mathbf{X}_{\mathcal{C},k}^{(j)} = \mathbf{P}_{k+1}^{\mathcal{C}}\mathbf{X}_{k+1}^{(j-1)},$$
$$\mathbf{X}_{\mathcal{L},k}^{(j)} = \mathbf{P}_k^{\mathcal{L}}\mathbf{X}_k^{(j-1)}, \text{ and } \mathbf{X}_{\mathcal{U},k}^{(j)} = \mathbf{P}_k^{\mathcal{U}}\mathbf{X}_k^{(j-1)}, \quad (2)$$

to propagate information from the boundary, coboundary, lower, and upper adjacent neighbors, respectively. In these equations, the integer j signifies the scale of message passing. The final transformation aggregates self-information with the information received from neighboring nodes as

$$\mathbf{X}_k^{(j)} = \texttt{AGGREGATE}(\mathbf{P}_k^{\mathcal{S}}\mathbf{X}_k^{(j-1)}, \mathbf{X}_{\mathcal{B},k}^{(j)}, \mathbf{X}_{\mathcal{C},k}^{(j)}, \mathbf{X}_{\mathcal{L},k}^{(j)}, \mathbf{X}_{\mathcal{U},k}^{(j)}). \quad (3)$$

In our experiments, we choose the AGGREGATE function to be summation. However, any aggregation function that does not change the dimensions after aggregation, such as average or weighted average, could be used. Fig. 1(a) demonstrates message propagation for an edge e in a simplicial complex. The process of aggregating features from the self, boundary, coboundary, upper, and lower edge adjacencies is demonstrated over the arrow from Fig. 1(a) to Fig. 1(b). Next, the process is repeated to accumulate information from the higher-hop neighbors. For instance, 2-hop neighborhood aggregation, again corresponding to edge e, is depicted over the arrow from Fig. 1(b) to Fig. 1(c).

Simplicial scattering transforms. With information gathered from all four neighborhoods and across different scales, the simplicial scattering transform is defined as

$$\psi_j^{k,\mathcal{B}}\mathbf{X} = \mathbf{X}_{\mathcal{B},k}^{(j-1)} - \mathbf{X}_{\mathcal{B},k}^{(j)}, \qquad \psi_j^{k,\mathcal{C}}\mathbf{X} = \mathbf{X}_{\mathcal{C},k}^{(j-1)} - \mathbf{X}_{\mathcal{C},k}^{(j)},$$
$$\psi_j^{k,\mathcal{L}}\mathbf{X} = \mathbf{X}_{\mathcal{L},k}^{(j-1)} - \mathbf{X}_{\mathcal{L},k}^{(j)}, \text{ and } \psi_j^{k,\mathcal{U}}\mathbf{X} = \mathbf{X}_{\mathcal{U},k}^{(j-1)} - \mathbf{X}_{\mathcal{U},k}^{(j)}. \quad (4)$$

Taking differences between aggregated features from consecutive scales, as shown in (3), allows us to preserve high-frequency information among different neighbors, resulting in discriminative features. The simplicial scattering transforms at scale $j = 1$ involving computing the difference between the aggregated features from the 1-hop and 2-hop boundary, coboundary, upper, and lower adjacent simplices, respectively, and aggregating these differences, specifically for edge e, is shown in Fig. 1 over the arrow from Fig. 1(c) to Fig. 1(d). This approach mitigates the oversmoothing commonly encountered in message-passing neural networks, where information from neighborhoods is averaged, potentially leading to a loss of discriminative power. By leveraging the differences between features from different scales, our method thus ensures the retention of valuable information while avoiding oversmoothing. Here, $\psi_j^{k,\mathcal{N}}\mathbf{X}$ denotes the high-frequency information between the j^{th} and $j-1^{th}$ neighborhood of type $\mathcal{N} \in \{\mathcal{B}, \mathcal{C}, \mathcal{L}, \mathcal{U}\}$. The features of k-simplices after the first order of scattering transform at the scale j, denoted by $|\Psi_j^k\mathbf{X}|$, are given by

$$|\Psi_j^k\mathbf{X}| = |\texttt{AGGREGATE}(\mathbf{X}_k, \psi_j^{k,\mathcal{B}}\mathbf{X}, \psi_j^{k,\mathcal{C}}\mathbf{X}, \psi_j^{k,\mathcal{L}}\mathbf{X}, \psi_j^{k,\mathcal{U}}\mathbf{X})|$$

where $|\cdot|$ represents the nonlinear modulus operator. The modulus operator is chosen due to its proven properties of being equivariant to permutations, nonexpansiveness, and stability (Zou & Lerman, 2020; Gama et al., 2019b), which generalizes well to simplicial complexes, as we theoretically show later on. Let J be a hyperparameter that represents the maximum scale. At the final scale J, we assign the embeddings as $|\Psi_J\mathbf{X}| = \{\mathbf{X}_0^{(J)}, \mathbf{X}_1^{(J)}, \ldots, \mathbf{X}_K^{(J)}\}$, as calculated by (3). We denote the set of embeddings of all orders by

$$|\Psi_j\mathbf{X}| = \{|\Psi_j^0\mathbf{X}|, |\Psi_j^1\mathbf{X}|, \ldots, |\Psi_j^k\mathbf{X}|\}.$$

Deep simplicial scattering network. The computations in (4) can be repeated multiple times, with the output of one layer serving as the input to the next. This results in a network for assigning representations for simplices, with each layer capturing different types of features. We call this network as the simplicial scattering network (SSN). For example, in the second layer of SSN, we repeat the computations from Equation (4), but this time we use $|\Psi_j\mathbf{X}|$

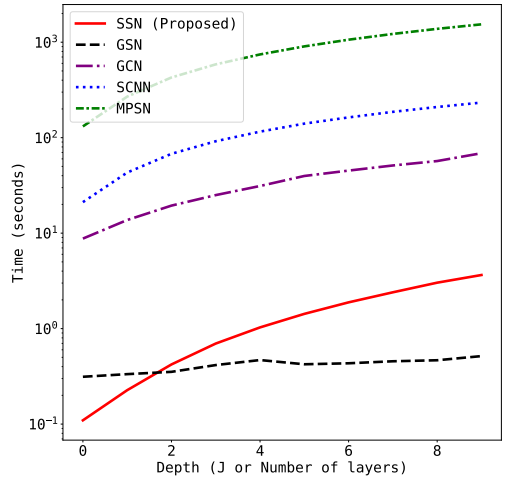

Figure 2: Runtime comparison of SSN with the geometric scattering network (GSN) (Gao et al., 2019) and existing graph (Kipf & Welling, 2017) and simplicial neural network models (Bodnar et al., 2021; Yang et al., 2022).

as the input. Thus, we obtain

$$|\Psi_{j'}^{k}|\Psi_j \mathbf{X}|| = \Big|\Big[|\Psi_j^k \mathbf{X}| + \psi_{j'}^{k,\mathcal{B}}|\Psi_j^k \mathbf{X}| + \psi_{j'}^{k,\mathcal{C}}|\Psi_j^k \mathbf{X}| + \psi_{j'}^{k,\mathcal{L}}|\Psi_j^k \mathbf{X}| + \psi_{j'}^{k,\mathcal{U}}|\Psi_j^k \mathbf{X}|\Big]\Big|,$$

where $1 \leq j, j' \leq J$. The set of embeddings for simplices of all orders after the second layer of the SSN is denoted as

$$|\Psi_{j'}|\Psi_j \mathbf{X}|| = \{|\Psi_{j'}^0|\Psi_j \mathbf{X}||, \ldots, |\Psi_{j'}^K|\Psi_j \mathbf{X}||\}.$$

We sequentially concatenate M layers in this manner to obtain the SSN embeddings. Finally, the embeddings of k-simplices from these different layers can be pooled as

$$S_k(\mathbf{X}, \mathbf{B}) = \texttt{COMBINE}(\mathbf{X}_k, |\Psi_j^k \mathbf{X}|, \Psi_{j'}^k|\Psi_j \mathbf{X}|, \ldots),$$

for $1 \leq j, j' \leq J$. In our experiments, we choose concatenation for the COMBINE function. Alternatively, any aggregation function such as sum or average can be used. The set containing embeddings of simplices of different orders is denoted by $\phi(\mathcal{X}) = \{S_k(\mathbf{X}, \mathbf{B})\}_{k=0}^{K}$.

In essence, SSN does not contain any learnable parameters, enabling feature extraction without needing any training labels while being simple and computationally efficient. We illustrate this in Fig. 2, wherein we compare the runtimes required to compute the embeddings using SSN with geometric scattering network (GSN) (Gao et al., 2019) and existing neural network-based models such as the graph convolutional network (GCN) (Kipf & Welling, 2017), the message passing simplicial network (MPSN) (Bodnar et al., 2021), and the simplicial convolutional neural network (SCNN) (Yang et al., 2022), specifically for the task of node classification on senate-bills dataset (Philip et al., 2021). For the neural network-based models, we measure the time required to train a network with J layers. For SSN and GSN, we compare the time taken to compute representations using a maximum scale of J. We observe that the runtime of the neural network models increases linearly with the number of layers. However, since GSN and SSN involve no training components, their runtime remains significantly small for all J, with SSN running slightly slower compared to GSN due to the inclusion of multiple adjacencies.

Next, we discuss the theoretical properties of the proposed computationally efficient SNN model for learning simplicial embeddings. Specifically, we present a theoretical demonstration that SSN satisfies all the fundamental properties of scattering transforms, such as permutation equivariance, nonexpansiveness, and robustness to perturbations. This is achieved despite the fact that simplicial scattering transforms do not involve powers of lazy random walk matrices, but instead adopt a sequential approach, which is distinct from the methods used in Euclidean and geometric scattering transforms. Additionally, we study the impact of higher-order interactions on the robustness of SSN and theoretically prove that incorporating higher-order information enhances the robustness of SSN to perturbations.

5. Theoretical Characterization

Permutation equivariance. Permutation equivariance is an essential property for SSN because the incidence matrix of a simplicial complex can be represented with various permutations. But, a permuted simplicial complex should have the same representations as the original one. This property of the SSN is encapsulated in the following theorem.

Theorem 5.1. *Let* $\{\Pi_k \in \mathbb{R}^{n_k \times n_k}\}_{k=0}^{K}$ *represent permutation matrices that act on the input feature matrices for each order* $k = 1, \ldots, K$. *Define* $\Pi\mathbf{X} = \{\Pi_k \mathbf{X}_k\}_{k=0}^{K}$ *and* $\Pi\mathbf{B}\Pi^{\mathrm{T}} = \{\Pi_{k-1}\mathbf{B}_k\Pi_k^{\mathrm{T}}\}_{k=0}^{K}$. *Then, we have*

$$S_k(\Pi\mathbf{X}, \Pi\mathbf{B}\Pi^{\mathrm{T}}) = \Pi S_k(\mathbf{X}, \mathbf{B}).$$

The proof of this theorem is available in Appendix A. Initially, wherein we show that the computation of random walk matrices and the proposed scattering transforms are permutation equivariant. Given that the nonlinearity function (i.e., the modulus operator) and the COMBINE (Concatenation) function, which aggregates embeddings from different scales, are permutation equivariant, it follows that SSN, as a composition of these functions, is also permutation equivariant.

Non-expansiveness. The non-expansiveness property of the scattering transform guarantees the conservation of signal energy, which is important to preserve expressivity and stability by bounding the scattering transform output. SSN also exhibits non-expansiveness, i.e., energy conservation

is guaranteed both before and after applying the scattering transform. We formalize this in the following theorem.

Theorem 5.2. *There exists a constant $C_k > 0$ for every order $k = 0, \dots, K$ such that*

$$C_k \|\mathbf{X}_k\| \leq \sum_{j=0}^{J} \|\Psi_j^k \mathbf{X}_k\| \leq \|\mathbf{X}_k\|.$$

The fact that C_k is greater than 0 implies that the information in the signal is not entirely lost; hence, the transformed features retain at least as much expressiveness as the initial features of the simplices. Furthermore, the upper bound suggests that the signal is not amplified, which indicates stability of the simplicial scattering transform.

The proof of this theorem is provided in Appendix B. We demonstrate non-expansiveness by showing that the proposed sequential message passing-based transforms can be represented in a form similar to that of scattering transforms in existing geometric scattering methods. We show that this representation holds even though the notion of adjacency in our context differs, and our transforms do not involve integer powers of lazy random walk matrices but instead follow a sequential approach.

Stability. The stability of the scattering transform is measured by its ability to handle perturbations in the input data, ensuring that similar inputs result in similar features in the feature space. We analyze the stability of the simplicial representations extracted using SSN. To characterize the stability of the representations computed using SSN, we define simplicial diffusion distance, extending the diffusion distance used in geometric scattering to characterize stability (Perlmutter et al., 2019).

Definition 5.3. Simplicial diffusion distance. Let $\mathcal{X} = \{\mathcal{X}_k\}_{k=0}^K$ and $\widetilde{\mathcal{X}} = \{\widetilde{\mathcal{X}}_k\}_{k=0}^K$ be two simplicial complexes, where $\mathcal{X}_k$ denotes the set of k-simplices. We have $|\mathcal{X}_k| = |\tilde{\mathcal{X}}_k|$ for $k = 0, \dots, K$, where $|\mathcal{S}|$ denotes the cardinality of the set $\mathcal{S}$. The random matrices corresponding to $\mathcal{X}$ are denoted by $\mathbf{P}_k^{\mathcal{B}}$, $\mathbf{P}_k^{\mathcal{L}}$, and $\mathbf{P}_k^{\mathcal{U}}$, and those corresponding to $\widetilde{\mathcal{X}}$ are denoted by $\widetilde{\mathbf{P}}_k^{\mathcal{B}}$, $\widetilde{\mathbf{P}}_k^{\mathcal{L}}$, and $\widetilde{\mathbf{P}}_k^{\mathcal{U}}$. The diffusion distance between the two simplicial complexes is defined as

$$\begin{aligned}\operatorname{dist}(\mathcal{X}, \widetilde{\mathcal{X}}) = \min_{\{\Pi_k \in \mathbb{P}_{|\mathcal{X}|_k}\}_{k=0}^K} \sum_{k=0}^{K} \|\mathbf{P}_k^{\mathcal{B}} - \Pi_{k-1} \widetilde{\mathbf{P}}_k^{\mathcal{B}} \Pi_k^T\|_2 \\ + \|\mathbf{P}_k^{\mathcal{L}} - \Pi_k \widetilde{\mathbf{P}}_k^{\mathcal{L}} \Pi_k^T\|_2 + \|\mathbf{P}_k^{\mathcal{U}} - \Pi_{k-1} \mathbf{P}_k^{\mathcal{U}} \Pi_{k-1}^T\|_2. \quad (5)\end{aligned}$$

The following theorem formally states the stability of the embeddings of SSN in terms of the simplicial diffusion distance.

Theorem 5.4. *Consider two simplicial complexes $\mathcal{X} = \{\mathcal{X}_k\}_{k=0}^K$ and $\widetilde{\mathcal{X}} = \{\widetilde{\mathcal{X}}_k\}_{k=0}^K$ with $|\mathcal{X}_k| = |\tilde{\mathcal{X}}_k|$ for $k = 0, \dots, K$. Let $\phi(\mathcal{X})$ and $\phi(\tilde{\mathcal{X}})$ represent the SSN embeddings of the simplicial complexes $\mathcal{X}$ and $\widetilde{\mathcal{X}}$, respectively. Then, the following inequality holds*

$$\left\| \phi(\mathcal{X}) - \phi(\tilde{\mathcal{X}}) \right\|_2^2 \leq L \operatorname{dist}(\mathcal{X}, \widetilde{\mathcal{X}})$$

for $L > 0$.

This theorem suggests that SSN embeddings are robust to small perturbations in the simplicial structure. In other words, minor changes in the structure of the simplicial complex will not significantly affect the resulting SSN embeddings. The proof of this theorem, which can be found in Appendix C, utilizes a similar approach to that used in geometric scattering. Despite the difference in the underlying transforms, with SSN using sequential message passing-based transforms as opposed to the powers of random walk matrices used in geometric scattering, we demonstrate an equivalence between the two, thereby establishing the stability of SSN embeddings with respect to perturbations in the simplicial structure. Furthermore, we perform a numerical experiment on the small-world graph (Watts & Strogatz, 1998) consisting of 200 nodes to illustrate the stability of SSN [cf. Fig. 3 in Appendix 3]. We observe that as the rewiring probability increases (i.e., more the structure changes), the simplicial diffusion distance increases, resulting in less stable representations.

Do we need aggregations from all the four adjacencies? Next, we theoretically examine the impact of including higher-order simplices on the stability of representations produced by the SSN. Specifically, we prove that the representations from SSN are more stable if $(k+1)$-simplices are used for feature aggregation in the message passing-based transformation stage of k-simplices, as opposed to using only $(k-1)$ and k simplices. This is formally stated in the following theorem.

Theorem 5.5. *Consider two simplicial complexes $\mathcal{X} = \{\mathcal{X}_k\}_{k=0}^K$ and $\widetilde{\mathcal{X}} = \{\widetilde{\mathcal{X}}_k\}_{k=0}^K$ with $|\mathcal{X}_k| = |\tilde{\mathcal{X}}_k|$ for $k = 0, \dots, K$. Denote the simplicial scattering embeddings of $\{\mathcal{X}_k\}_{k=0}^K$ in $\mathcal{X}$ (or $\{\tilde{\mathcal{X}}_k\}_{k=0}^K$ in $\tilde{\mathcal{X}}$, respectively) as $\phi(\mathcal{X})$ (or $\phi(\tilde{\mathcal{X}})$). Consider a minor modification to the SSN model, where the embeddings of simplices of a particular order, say k, are now computed without propagating information from $(k+1)$-simplices. Let the output of the simplicial scattering model with such a modification be denoted by $\phi^{\setminus k+1}(\mathcal{X})$ (or $\phi^{\setminus k+1}(\tilde{\mathcal{X}})$). Then, there exists constants L and $L_{\setminus k+1}$ such that*

$$\left\| \phi(\mathcal{X}) - \phi(\tilde{\mathcal{X}}) \right\|_2^2 \leq L \operatorname{dist}(\mathcal{X}, \tilde{\mathcal{X}})$$

and

$$\left\| \phi^{\setminus k+1}(\mathcal{X}) - \phi^{\setminus k+1}(\tilde{\mathcal{X}}) \right\|_2^2 \leq L_{\setminus k+1} \operatorname{dist}(\mathcal{X}, \tilde{\mathcal{X}}),$$

with $L \leq L_{\setminus k+1}$.

This theorem suggests that for simplices of any order, utilizing information from higher-order simplices enhances the stability of their representations.

While the computational efficiency, nonexpansivenss, and stability can be demonstrated through theoretical analysis and visual illustrations, the expressivity of SSN requires validation through comprehensive numerical experiments.

6. Numerical Experiments

In this section, we empirically evaluate SSN, assessing its performance across five downstream tasks: node classification, graph classification, simplicial closure prediction, trajectory prediction, and simplex prediction. [1] The embeddings computed using the SSN method can be utilized to perform downstream tasks by feeding them into a simple classifier. We employ a ridge classifier to classify nodes from their embeddings, predict the closure of open simplices from the embeddings of their constituent simplices, predict trajectories from edge embeddings, and classify simplices from their embeddings. However, for graph classification, we use a multi-layer perception. The code to run the experiments is available at https://github.com/HirenMadhu/SSN.

6.1. Baselines

We evaluate our approach by comparing it against various supervised and unsupervised techniques, denoted as S and US, respectively. More details about these methods are detailed in the following subsection. We compare the performance of our unsupervised SSN model against a variety of both supervised and unsupervised models. This includes simplicial neural network-based supervised models such as simplicial convolutional neural networks (SCNN) (Yang et al., 2022) and message passing neural networks (MPSN) (Bodnar et al., 2021). We also benchmark against graph neural network-based supervised models, including graph convolutional network (GCN) (Kipf & Welling, 2017), graph attention networks (GAT) (Veličković et al., 2018), graph isomorphism networks (GIN) (Xu et al., 2018), and GraphSage (Hamilton et al., 2017). In addition, we compare our SSN model with the unsupervised geometric scattering network (GSN) (Gao et al., 2019) and other supervised geometric scattering models such as scattering GCN (ScGCN) (Min et al., 2020b), and geometric scattering attention network (GSAN) (Min et al., 2020a). We also introduce a variant of our SSN model, denoted by SSN-WOB, in which the aggregation is performed using only the Hodge Laplacians and ignoring boundary and coboundary adjacencies. Additionally, we introduce a method named Diffusion. In this approach, we only concatenate $\mathbf{X}^{(j)}$ across scales without computing the difference. Lastly, we include two additional baseline methods for comparison. For simplicial closure prediction, we consider logistic regression (Log. Reg.) as described in (Benson et al., 2018). For trajectory prediction, we use a projection-based method (denoted as Projection) from (Roddenberry et al., 2021), which is based on projecting the input edge features onto the Hodge Laplacian kernel. For all the graph-based baselines, we utilize the first-order simplicial complexes as inputs wherever simplicial complex structures are available, such as in all tasks other than graph classification. When only graph structures are available, as in the case of graph classification, the graph-based models take the graph as input, while the simplicial complex-based models take clique-lifted graphs as input.

[1] Details related to the datasets, feature initialization, and experimental setup can be found in Appendix E.

6.2. Downstream Tasks

We discuss the details of these tasks next.

Node classification. Node classification involves predicting labels for 0-simplices, within a given simplicial complex. We carry out node classification on publicly available datasets, namely the primary-school, high-school, and senate-bills datasets (Philip et al., 2021). The performance metric utilized to evaluate node classification is the accuracy of the classification.

Graph classification. Graph classification involves categorizing graphs (specifically, clique-lifted graphs) as belonging to different classes. Since this task requires the extraction of representations for the entire graph, we concatenate embeddings of different simplices. Our experiments are conducted on the PROTEINS, NCI1, REDDIT-B, REDDIT-M, and IMDB-B datasets from the TUDatasets repository (Morris et al., 2020). We employ mean accuracy calculated after training over ten folds as the metric.

Simplicial closure prediction. For simplicial closure prediction, we focus on predicting the closure of open triangles within a time series of simplicial complex data. To obtain a representation for an open triangle, we concatenate the representations of its constituent nodes and edges. This representation is then used for classification purposes. We carry out this task on the email-Enron, primary-school, and high-school datasets. The data is temporally split, with the first 80% used for training the encoder and the remaining 20% for inference. Given the skewed nature of the dataset, we use the area under the precision-recall curve (AUC-PR) as the performance metric.

Trajectory prediction. Trajectory prediction is a task that involves predicting the next node in a sequence, given a sequence of nodes. We assess the performance of trajectory prediction on the Ocean (Roddenberry et al., 2021),

Method	Type	high-school	primary-school	senate-bills
GCN	S	98.2 ± 1.3	76.12 ± 2.89	80.34 ± 2.65
GraphSage	S	95.2 ± 2.1	76.94 ± 3.54	78.47 ± 9.77
GAT	S	93.9 ± 4.3	71.02 ± 5.38	80.51 ± 2.96
GIN	S	94.7 ± 1.8	80.00 ± 3.63	68.31 ± 17.48
ScGCN	S	42.4 ± 4.4	29.2 ± 2.2	66.4 ± 3.9
GSAN	S	42.3 ± 5.0	28.4 ± 2.1	67.1 ± 3.1
GSN	US	27 ± 2	31 ± 3.8	58.3 ± 5.4
SCNN	S	86 ± 5	68.14 ± 3.18	77 ± 4
MPSN	S	91 ± 2	72.65 ± 4.58	77.1 ± 4.1
Diffusion	US	85 ± 3.13	76.3 ± 4.2	78.47 ± 4.59
SSN-WOB	US	95.91 ± 3.32	81.63 ± 3.29	80.85 ± 1.86
SSN	US	**99.39 ± 0.74**	**91.43 ± 2.20**	**83.22 ± 3.9**

Table 1: Accuracies of node classification on simplicial complex datasets.

Method	Type	PROTEINS	NCI1	IMDB-B	REDDIT-B	REDDIT-M
DGCNN	S	72.9 ± 3.5	76.4 ± 1.7	69.2 ± 3.0	87.8 ± 2.5	49.2 ± 1.2
GIN	S	73.3 ± 4.5	80.0 ± 1.4	71.2 ± 3.9	89.9 ± 1.9	56.1 ± 1.7
GraphSage	S	73.0 ± 4.5	76.0 ± 1.8	68.8 ± 4.5	84.3 ± 1.9	50.0 ± 1.3
GSN	US	74.1 ± 4.0	79.1 ± 1.2	71.2 ± 3.2	89.6 ± 2	53.3 ± 1.4
MPSN	S	76.5 ± 3.4	**82.8 ± 2.2**	**75.6 ± 3.2**	**92.2 ± 1.0**	**57.3 ± 1.6**
SSN	US	**78.2 ± 3.4**	77.7 ± 2.6	74.8 ± 3.6	88.4 ± 3.2	52.2 ± 1.4

Table 2: Graph classification accuracies on TUDatasets.

Synthetic (Roddenberry et al., 2021), and Mesh (Cordonnier & Loukas, 2018) datasets. The metric used to analyze the performance of our model is classification accuracy.

Simplex prediction: Simplex prediction involves classifying the type of a simplex within a given simplicial complex. If a higher-order representation is available, we use it to classify the simplex type. Otherwise, we average the node representations for classification. We report the classification accuracy for the madison-reviews, algebra-questions, and geometry-questions datasets (Ilya et al., 2020).

In all the results presented as tables in the paper, bold indicates the best and underline means the second best.

6.3. Discussion

Performance. The empirical results demonstrate that the proposed unsupervised SSN method surpasses all graph-based methods, including supervised graph neural network methods and both supervised and unsupervised geometric scattering methods. Notably, SSN also consistently outperforms supervised simplicial complex-based baselines, except for some graph classification experiments, wherein MPSN slightly outperforms SSN on certain datasets.

In the case of node classification (Table 1), SSN exceeds all the supervised graph methods and simplicial methods, with an average increase of 5.23% compared to the best-performing graph-based method and an average increase of 11.1% compared to the best-performing simplicial method. For simplicial closure prediction (Table 3), SSN outperforms the non-neural network-based logistic regression model by a substantial margin. It also surpasses the simplicial neural network models MPSN and SCNN on all three datasets. In the trajectory prediction task (Table 4), SSN significantly outperforms existing methods on the ocean drifters dataset and the synthetic dataset, with an average

Method	Type	email-Enron	high-school	primary-school
Random baseline (RB)	Untrained	0.0537	0.0112	0.0105
Log. Reg. / RB	S	0.55 ± 0.0	0.59 ± 0.0	1.79 ± 0.0
SCNN/RB	S	14.17±0.1	9.10±0.0	20.19±0.0
MPSN/RB	S	14.51±0.0	84.34±0.00	41.05±0.20
SSN-WOB/RB	US	17.20 ± 0.97	68.48 ± 1.39	**47.62±0.0**
SSN/RB	US	**17.26 ± 0.87**	**85.36 ± 1.17**	**47.62±0.0**

Table 3: AUC-PR scores for simplicial closure prediction on various datasets.

Method	Type	Ocean	Synthetic	Mesh
Projection	S	27.15±0.0	52.0±0.0	30.9±0.0
SCoNe	S	30.0±0.6	55.4±1.1	**98.5±1.9**
SCNN	S	28.5±0.6	50.5±1.0	98.1±2.0
SSN	US	**49±2**	**61±0**	97.7±0.05

Table 4: Accuracy results from trajectory prediction experiments.

increase of 11.9%. On the Mesh dataset, although SSN does not achieve the best performance, it exhibits a lower standard deviation than the other two methods. In the graph classification task (Table 2), SSN outperforms graph methods on the PROTEINS and IMDB-B datasets. Finally, in the simplex classification task (Table 5), SSN outperforms all the graph-based models on all the datasets and simplicial neural network-based models on the majority of the datasets.

Taking differences in SSN preserves the discriminative power of the embeddings. We observe that using Diffusion embeddings instead of the difference scattering embeddings results in significantly poorer performance. This highlights the advantage of computing differences between aggregated features at consecutive scales, which enhances feature expressiveness and minimizes information loss from averaging. Furthermore, we also notice that Diffusion does outperform MPSN and requires the learnable parameters from MPSN to compute discriminative features for simplices.

SNNs require abundant data in order to achieve optimal performance. An interesting observation from the node classification results (Table 1) is that GNNs tend to perform better than SNNs. This could be attributed to the increased number of parameters in the simplicial models, which cannot converge as effectively as simpler graph models with fewer parameters. The increased parameter complexity of the simplicial neural network-based models necessitates more data to learn the optimal set of parameters. Thus, simplicial-based methods do not perform as well compared to graph-based models. However, because SSN does not require training and considers the simplicial structure, the extracted features are expressive and simplicial-aware, leading to superior performance. In contrast, in Table 2 and 5, we observe that simplicial models perform well because substantial higher-order data is available.

Method	Type	madison-reviews	algebra-questions	geometry-questions
GCN	S	95.5±1.7	66.1±2.6	72.1 ± 2.5
GSN	US	55.6 ± 2.1	37 ± 2.6	43.2 ± 2.3
SCNN	S	89.9 ± 2.9	65 ± 2	79.1 ± 1.8
MPSN	S	94 ± 2.4	66.9 ± 2.5	**82.3 ± 2.1**
Diffusion	US	91.7 ± 2.5	61.5 ± 3.5	69.1 ± 2.2
SSN-WOB	US	94.4 ± 1.4	55.20 ± 3.6	64.81 ± 3.49
SSN	US	**96.3 ± 1**	**73.2 ± 2.8**	79.9 ± 2.1

Table 5: Simplex prediction accuracies on various datasets.

SSN captures global information of simplicial complexes. Another observation from the simplicial closure experiments (Table 3) is that the ideal representation for simplicial closure should embed both local information of the simplex of interest and global information in the simplicial complex. This is achievable with SSN because increasing the scale J enhances the information aggregation range and scattering features take a global view of the simplicial complex.

SNNs outperform existing graph baselines on large datasets with small graphs. For graph classification (Table 2), we observe that MPSN outperforms all the methods. Since many graphs are available as training data, the MPSN can learn an optimal set of parameters in a supervised training setting. Further, MPSN is as powerful as the SWL test (Bodnar et al., 2021), i.e., it computes unique representations of two simplicial complexes if they are not isomorphic by the SWL test. This also contributes to the superior performance of MPSN in graph classification tasks.

SSN performs on par with graph-based methods when no higher-order data is available. From Table 2, we note that SSN performs on par with certain graph-based methods, such as GIN, on NCI1. This is due to the absence of cliques in these datasets, resulting in simplicial complexes being identical to graphs. As a result, these simplicial complexes fail to provide any supplementary information from which SSN could benefit. In other datasets, SSN either outperforms graph methods or yields similar performance compared to graph methods, as ample cliques exist.

Summary. When sufficient higher-order information is present, SSN performs better than graph-based models and outperforms SNNs in tasks requiring a global view of the data and in situations where sufficient training data is unavailable. Another common observation across all tasks is that SSN benefits from including boundary and coboundary information over only upper and lower adjacency data, aligning with the theory in (Bodnar et al., 2021) that such inclusion enhances embedding expressivity.

Ablation study. To illustrate the significance of the maximum scale J, we test SSN with varying values of J. The results are presented in Appendix F (Table 8), where we show that the performance of SSN increases with J until it saturates. Additionally, we assessed the effectiveness of SSN in low-resource settings by training it and the baselines with different levels of training data. SSN is observed to outperform MPSN, especially when less labeled data was available, further underscoring the suitability of SSN in data-limited scenarios.

7. Related works

The simplicial scattering model draws inspiration from scattering methods initially developed for Euclidean data, such as images and audio (Mallat, 2012). This approach was later adapted into a layered structure, similar to convolutional neural networks (CNNs), and has proven effective in various audio and image processing applications, particularly in low-resource settings (Bruna & Mallat, 2013a;b; Andén & Mallat, 2014; Angles & Mallat, 2018). However, these methods primarily cater to Euclidean data and do not account for relations between dataset samples.

Extending scattering transforms to graph data, Gama et al. (2019b); Gao et al. (2019) employs diffusion wavelets introduced in Coifman & Maggioni (2006) to extract graph features. Following this work, methods that fuse scattering features with some learning components have been introduced (Tong et al., 2020; Castro et al., 2020; Angles & Mallat, 2018; Min et al., 2020b). Note we can introduce learning components in SSN by making the AGGREGATE and COMBINE functions learnable. However, due to the learnable components, they require labels to train the parameters. In a different approach, Zou & Lerman (2020) introduces geometric scattering that utilizes spectral wavelets introduced in Hammond et al. (2009). However, this method is computationally expensive due to spectral decomposition. Furthermore, similar to GNNs, all the above geometric scattering methods are constrained to pairwise interactions.

The Hodgelets method (Roddenberry et al., 2022) extends (Zou & Lerman, 2020) to simplicial complexes using spectral decomposition of the Hodge Laplacian matrices. However, the Hodgelets method does not consider boundary and coboundary adjacencies, which are known to enhance the expressivity of simplicial models Bodnar et al. (2021). Further, Hodgelets require spectral decomposition, which is computationally expensive, hindering the scalability of the method.

8. Conclusion

We introduced SSN, a computationally efficient, training-free model for computing embeddings in a simplicial complex. SSN uses lazy random walk matrices and a sequentially designed filter bank to capture frequency components in the input data, producing task-agnostic, simplicial-aware representations. We showed that SSN exhibits permutation equivariance, nonexpansiveness, and robustness to minor simplicial structure perturbations. We also demonstrated that using higher-order simplices for feature aggregation in the transformation stage improves the stability of the output representations. Numerical experiments also showed that SSN performs better when it includes boundary and coboundary information. SSN also outperformed graph-based models when sufficient higher-order information is available and surpasses other simplicial neural network models in tasks requiring a global view of the data or when training data is limited.

Acknowledgements

This research was partially supported by the Kotak IISc AI-ML Centre (KIAC) and by the DRDO grant DFTM/02/3125/M/02/AIR-02.

Impact statement

This paper presents a theoretical framework for simplicial representation learning work that aims to advance the field of Machine Learning. Societal consequences of our work are unclear or none of which we feel must be specifically highlighted here.

References

Andén, J. and Mallat, S. Deep scattering spectrum. *IEEE Transactions on Signal Processing*, 62(16):4114–4128, 2014.

Angles, T. and Mallat, S. Generative networks as inverse problems with scattering transforms. *arXiv preprint arXiv:1805.06621*, 2018.

Benson, A. R., Abebe, R., Schaub, M. T., Jadbabaie, A., and Kleinberg, J. Simplicial closure and higher-order link prediction. *IEEE Transactions on Neural Networks and Learning systems*, 115(48):E11221–30, 2018.

Bodnar, C., Frasca, F., Wang, Y., Otter, N., Montufar, G. F., Lio, P., and Bronstein, M. Weisfeiler and lehman go topological: Message passing simplicial networks. In *International Conference on Machine Learning*, pp. 1026–1037. PMLR, 2021.

Bruna, J. and Mallat, S. Audio texture synthesis with scattering moments. *arXiv preprint arXiv:1311.0407*, 2013a.

Bruna, J. and Mallat, S. Invariant scattering convolution networks. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1872–1886, 2013b.

Castro, E., Benz, A., Tong, A., Wolf, G., and Krishnaswamy, S. Uncovering the folding landscape of rna secondary structure with deep graph embeddings. *arXiv preprint arXiv:2006.06885*, 2020.

Coifman, R. R. and Maggioni, M. Diffusion wavelets. *Applied and Computational Harmonic Analysis*, 21(1):53–94, 2006. ISSN 1063-5203. doi: https://doi.org/10.1016/j.acha.2006.04.004. URL https://www.sciencedirect.com/science/article/pii/S106352030600056X. Special Issue: Diffusion Maps and Wavelets.

Cordonnier, J. B. and Loukas, A. Extrapolating paths with graph neural networks. *arXiv preprint 1903.07518*, 2018.

Gama, F., Ribeiro, A., and Bruna, J. Stability of graph scattering transforms. *Advances in Neural Information Processing Systems*, 32, 2019a.

Gama, F., Ribeiro, A., and Bruna, J. Diffusion scattering transforms on graphs. In *International Conference on Learning Representations*, 2019b. URL https://openreview.net/forum?id=BygqBiRcFQ.

Gao, F., Wolf, G., and Hirn, M. Geometric scattering for graph data analysis. In *International Conference on Machine Learning*, pp. 2122–2131. PMLR, 2019.

Hamilton, W. L., Ying, R., and Leskovec, J. Inductive representation learning on large graphs. *CoRR*, abs/1706.02216, 2017. URL http://arxiv.org/abs/1706.02216.

Hammond, D. K., Vandergheynst, P., and Gribonval, R. Wavelets on graphs via spectral graph theory, 2009.

Ilya, A., Nate, V., and Austin, R. B. Clustering in graphs and hypergraphs with categorical edge labels. *Proceedings of the Web Conference (WWW)*, 2020. URL https://www.cs.cornell.edu/~arb/data/.

Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=SJU4ayYgl.

Madhu, H. and Chepuri, S. P. TopoSRL: Topology preserving self-supervised simplicial representation learning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=INS3ltgjg7.

Mallat, S. Group invariant scattering. *Communications on Pure and Applied Mathematics*, 65(10):1331–1398, 2012.

Min, Y., Wenkel, F., and Wolf, G. Geometric scattering attention networks. *CoRR*, abs/2010.15010, 2020a. URL https://arxiv.org/abs/2010.15010.

Min, Y., Wenkel, F., and Wolf, G. Scattering gcn: Overcoming oversmoothness in graph convolutional networks. *Advances in neural information processing systems*, 33: 14498–14508, 2020b.

Morris, C., Kriege, N. M., Bause, F., Kersting, K., Mutzel, P., and Neumann, M. Tudataset: A collection of benchmark datasets for learning with graphs. *arXiv preprint arXiv:2007.08663*, 2020.

Perlmutter, M., Gao, F., Wolf, G., and Hirn, M. Understanding graph neural networks with asymmetric geometric scattering transforms. *arXiv preprint arXiv:1911.06253*, 2019.

Philip, S. C., Nate, V., and Austin, R. B. Generative hypergraph clustering: from blockmodels to modularity. *Science Advances*, 2021. URL https://www.cs.cornell.edu/~arb/data/.

Roddenberry, T. M., Glaze, N., and Segarra, S. Principled simplicial neural networks for trajectory prediction. In *International Conference on Machine Learning*, pp. 9020–9029. PMLR, 2021.

Roddenberry, T. M., Frantzen, F., Schaub, M. T., and Segarra, S. Hodgelets: Localized spectral representations of flows on simplicial complexes. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5922–5926. IEEE, 2022.

Schaub, M. T., Benson, A. R., Horn, P., Lippner, G., and Jadbabaie, A. Random walks on simplicial complexes and the normalized hodge 1-laplacian. *SIAM Review*, 62 (2):353–91, 2020.

Tong, A., Wenkel, F., MacDonald, K., Krishnaswamy, S., and Wolf, G. Data-driven learning of geometric scattering networks. *arXiv preprint arXiv:2010.02415*, 2020.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., and Bengio, Y. Graph attention networks, 2018.

Watts, D. J. and Strogatz, S. H. Collective dynamics of 'small-world'networks. *nature*, 393(6684):440–442, 1998.

Xu, K., Hu, W., Leskovec, J., and Jegelka, S. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.

Yang, M., Isufi, E., and Leus, G. Simplicial convolutional neural networks. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8847–8851. IEEE, 2022.

Zou, D. and Lerman, G. Graph convolutional neural networks via scattering. *Applied and Computational Harmonic Analysis*, 49(3):1046–1074, 2020.

A. Proof of Theorem 5.1

With boundary matrices permuted as $\Pi\mathbf{B}\Pi^{\mathrm{T}} = \{\Pi_{k-1}\mathbf{B}_k\Pi_k^{\mathrm{T}}\}_{k=0}^{K}$, the random walk matrices corresponding to boundary, upper, and lower adjacencies, are altered to

$$\begin{aligned}
\bar{\mathbf{P}}_{k+1}^{\mathcal{C}} &= \Pi_k(\mathbf{D}_{k+1}^{\mathcal{C}})^{-1}\Pi_k^{\mathrm{T}}\Pi_k\mathbf{B}_{k+1}\Pi_{k+1}^{\mathrm{T}} = \Pi_k(\mathbf{D}_{k+1}^{\mathcal{C}})^{-1}\mathbf{B}_{k+1}\Pi_{k+1}^{\mathrm{T}} = \Pi_k\mathbf{P}_{k+1}^{\mathcal{C}}\Pi_{k+1}^{\mathrm{T}},\\
\bar{\mathbf{P}}_{k}^{\mathcal{B}} &= \Pi_k\mathbf{B}_k^{\mathrm{T}}\Pi_{k-1}^{\mathrm{T}}\Pi_{k-1}(\mathbf{D}_{k}^{\mathcal{B}})^{-1}\Pi_{k-1}^{\mathrm{T}} = \Pi_k\mathbf{B}_k^{\mathrm{T}}(\mathbf{D}_{k}^{\mathcal{B}})^{-1}\Pi_{k-1}^{\mathrm{T}} = \Pi_k\mathbf{P}_{k}^{\mathcal{B}}\Pi_{k-1}^{\mathrm{T}},\\
\bar{\mathbf{P}}_{k}^{\mathcal{L}} &= \Pi_k\mathbf{B}_k^{T}\Pi_{k-1}^{\mathrm{T}}\Pi_{k-1}\mathbf{B}_k\Pi_k^{\mathrm{T}}\Pi_k(\mathbf{D}_{k}^{\mathcal{L}})^{-1}\Pi_k^{\mathrm{T}} = \Pi_k\mathbf{B}_k^{T}\mathbf{B}_k(\mathbf{D}_{k}^{\mathcal{L}})^{-1}\Pi_k^{\mathrm{T}} = \Pi_k\mathbf{P}_{k}^{\mathcal{L}}\Pi_k^{\mathrm{T}},\\
\bar{\mathbf{P}}_{k}^{\mathcal{U}} &= \Pi_k\mathbf{B}_{k+1}\Pi_{k+1}^{\mathrm{T}}\Pi_{k+1}\mathbf{B}_{k+1}^{T}\Pi_k^{\mathrm{T}}\Pi_k(\mathbf{D}_{k}^{\mathcal{U}})^{-1}\Pi_k^{\mathrm{T}} = \Pi_k\mathbf{B}_{k+1}\mathbf{B}_{k+1}^{T}(\mathbf{D}_{k}^{\mathcal{U}})^{-1}\Pi_k^{\mathrm{T}} = \Pi_k\mathbf{P}_{k}^{\mathcal{U}}\Pi_k^{\mathrm{T}},
\end{aligned} \tag{6}$$

respectively. Here, we have used the fact that $\Pi_k^{\mathrm{T}}\Pi_k = \mathbf{I}$ to simplify the expressions. The message passing-based transforms in (2) corresponding to information propagation from boundary, coboundary, upper, and lower adjacent neighbors are

$$\begin{aligned}
\bar{\mathbf{X}}_{\mathcal{B},k}^{(j)} &= \bar{\mathbf{P}}_{k}^{\mathcal{B}}\Pi_{k-1}\mathbf{X}_{k-1}^{(j-1)} = \Pi_k\mathbf{P}_{k}^{\mathcal{B}}\mathbf{X}_{k-1}^{(j-1)} = \Pi_k\mathbf{X}_{\mathcal{B},k}^{(j)}\\
\bar{\mathbf{X}}_{\mathcal{C},k}^{(j)} &= \bar{\mathbf{P}}_{k+1}^{\mathcal{C}}\Pi_{k+1}\mathbf{X}_{k+1}^{(j-1)} = \Pi_k\mathbf{P}_{k+1}^{\mathcal{C}}\mathbf{X}_{k+1}^{(j-1)} = \Pi_k\mathbf{X}_{\mathcal{C},k}^{(j)}\\
\bar{\mathbf{X}}_{\mathcal{L},k}^{(j)} &= \bar{\mathbf{P}}_{k}^{\mathcal{L}}\Pi_k\mathbf{X}_{k}^{(j-1)} = \Pi_k\mathbf{P}_{k}^{\mathcal{L}}\mathbf{X}_{k}^{(j-1)} = \Pi_k\mathbf{X}_{\mathcal{L},k}^{(j)}\\
\bar{\mathbf{X}}_{\mathcal{U},k}^{(j)} &= \bar{\mathbf{P}}_{k}^{\mathcal{U}}\Pi_k\mathbf{X}_{k}^{(j-1)} = \Pi_k\mathbf{P}_{k}^{\mathcal{U}}\mathbf{X}_{k}^{(j-1)} = \Pi_k\mathbf{X}_{\mathcal{U},k}^{(j)},
\end{aligned} \tag{7}$$

respectively. The final update aggregation transform for the permuted data is therefore given by

$$\begin{aligned}
\bar{\mathbf{X}}_{k}^{(j)} &= \bar{\mathbf{X}}_{k}^{(j-1)} + \bar{\mathbf{X}}_{\mathcal{B},k}^{(j)} + \bar{\mathbf{X}}_{\mathcal{C},k}^{(j)} + \bar{\mathbf{X}}_{\mathcal{L},k}^{(j)} + \bar{\mathbf{X}}_{\mathcal{U},k}^{(j)}\\
&= \Pi_k\left(\mathbf{X}_{k}^{(j-1)} + \mathbf{X}_{\mathcal{B},k}^{(j)} + \mathbf{X}_{\mathcal{C},k}^{(j)} + \mathbf{X}_{\mathcal{L},k}^{(j)} + \mathbf{X}_{\mathcal{U},k}^{(j)}\right)\\
&= \Pi_k\mathbf{X}_{k}^{(j)}.
\end{aligned} \tag{8}$$

This leads to the simplicial scattering transform being $\Pi_k(\mathbf{X}_k^{(j-1)} - \mathbf{X}_k^{j})$, which is $\Pi_k\Psi_j^k\mathbf{X}$, i.e., when the data $\mathbf{X}_k$ and the boundary matrices $\mathbf{B}_k$ for $k = 0, 1, \ldots, K$ are permuted using set of permutation matrices $\Pi = \{\Pi_0, \ldots, \Pi_K\}$, the output of the simplicial scattering transform after one layer, denoted by $\Psi_j^k\mathbf{X}$, is permuted by the same permutation matrix. Hence, the operations involved in simplicial scattering are proved to be permutation equivariant. As the subsequent layer in the model involves the same operations but with $\Pi_k\Psi_j\mathbf{X}$ as input, the model continues to be permutation equivariant. The same reasoning could to applied to the subsequent J layers that follow. Since we choose the final `COMBINE` function to be concatenation, which preserves permutation equivariance, the permutation equivariance `SSN` is proved.

B. Proof of Theorem 5.2

Despite the fact that `SSN` incorporates multiple aggregations from various types of adjacent simplices of different orders, the complete set of random walk-based diffusion transforms as given in (2) can be compactly represented using a single diffusion matrix. Specifically, consider the following matrix $\mathbf{T} \in \mathbb{R}^{N\times N}$

$$\mathbf{T} = \begin{bmatrix}
\mathbf{I}_{n_0} + \mathbf{P}_0^{\mathcal{U}} & \mathbf{P}_1^{\mathcal{C}} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0}\\
\mathbf{P}_1^{\mathcal{B}} & \mathbf{I}_{n_1} + \mathbf{P}_1^{\mathcal{U}} + \mathbf{P}_1^{\mathcal{L}} & \mathbf{P}_2^{\mathcal{C}} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0}\\
\mathbf{0} & \mathbf{P}_2^{\mathcal{B}} & \mathbf{I}_{n_2} + \mathbf{P}_2^{\mathcal{U}} + \mathbf{P}_2^{\mathcal{L}} & \mathbf{P}_3^{\mathcal{C}} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0}\\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots\\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{P}_K^{\mathcal{B}} & \mathbf{I}_{n_K} + \mathbf{P}_K^{\mathcal{L}}
\end{bmatrix} \tag{9}$$

where $N = \sum_{i=0}^{k} N_k$. Consider a matrix $\bar{\mathbf{X}}$ that stacks the features of simplices of different orders, denoted as $\bar{\mathbf{X}} = [{\mathbf{X}_0^{(j)}}^{\mathrm{T}}, {\mathbf{X}_1^{(j)}}^{\mathrm{T}}, \cdots, {\mathbf{X}_K^{(j)}}^{\mathrm{T}}]^{\mathrm{T}}$. When we multiply the diffusion matrix $\mathbf{T}$ with $\bar{\mathbf{X}}$, it equates to executing the random walk-based diffusion transforms for simplices of all orders. More specifically, the operations that correspond to the multiplication of the kth row in $\mathbf{T}$ with the kth entry in $\bar{X}$ yield results equivalent to those in equation (2). Let $\bar{\mathbf{X}}^{j+1} = \mathbf{T}^{(j)}\bar{\mathbf{X}}$. The `SSN` embeddings of all orders after the first layer, denoted as $|\Psi_j\bar{\mathbf{X}}|$, can now be computed by $|\Psi_j\bar{\mathbf{X}}| = |\bar{\mathbf{X}}^{(j-1)} - \bar{\mathbf{X}}^{(j)}|$, for $j < J$ and as $|\Psi_J\bar{\mathbf{X}}| = \bar{\mathbf{X}}^{(J)}$ for $j = J$. By appropriately indexing the entries of $|\Psi_j\bar{\mathbf{X}}|$, first order simplicial scattering features for simplices of order k can be obtained. This procedure can be extended to the further layers in the `SSN` model.

While the simplicial scattering transform differs from geometric scattering transforms due to its sequential application of random walk-based transforms from various neighbor types, we find a parallel in the way the simplicial complex is represented. This representation is similar to the form of geometric scattering as detailed in the work by Gama et al. (2019b). Building on the nonexpansiveness proof of geometric scattering found in Gama et al. (2019b), we can similarly establish the nonexpansiveness of SSN.

C. Proof of Theorem 5.4

Let the diffusion operators of the simplicial complexes $\mathcal{X}$ and $\tilde{\mathcal{X}}$, denoted by $\mathbf{T}$ and $\tilde{\mathbf{T}}$, respectively, be defined as in (9). These diffusion matrices enable us to represent the simplicial scattering transform in a manner analogous to the presentation of geometric scattering. This is achieved despite not taking the powers of lazy random walk matrices, as is done in geometric scattering transforms, as detailed in Appendix B. This representation also enables us to extend the stability properties inherent to geometric scattering models to simplicial scattering models. It has been proved in Gama et al. (2019b) that the stability of the geometric scattering models is dependent solely on the spectral gap of the graph diffusion matrices or the lazy random walk matrix associated with a graph. The spectral gap of a matrix is defined as the difference between its two largest eigenvalues. Observing the similarity in the structure of the two transforms, where the graph diffusion matrix in geometric scattering is replaced by $\mathbf{T}$ in (9) for SSN, we can assert that the stability of SSN is entirely dependent on the spectral gap of the diffusion matrix $\mathbf{T}$.

Let the eigenvalues of $\mathbf{T}$ and $\tilde{\mathbf{T}}$ be denoted by $1 = \lambda_0 > \lambda_1 \geq \ldots \geq \lambda_N \geq 0$ and $1 = \tilde{\lambda}_0 > \tilde{\lambda}_1 \geq \ldots \geq \tilde{\lambda}_N \geq 0$, respectively. Note that the largest eigenvalues of these matrices are 1, a result of the construction of random walk matrices. Specifically, because all individual random walk matrices corresponding to all five possible movements (including staying put) are normalized and further scaled by a factor of $\frac{1}{5}$. Consequently, the spectral gaps of $\mathbf{T}$ and $\tilde{\mathbf{T}}$, denoted by β and $\tilde{\beta}$, are $1 - \lambda_1$ and $1 - \tilde{\lambda}_1$, respectively. Let us define $\beta^* = \max\left\{\beta, \tilde{\beta}\right\}$. Then, from Lemma 5.1 of (Gama et al., 2019b), we have

$$\left\|\phi(\mathcal{X}) - \phi(\tilde{\mathcal{X}})\right\|_2^2 \leq 2\sqrt{\frac{\beta^{*2}(1+\beta^{*2})}{(1-\beta^{*2})^3}}\operatorname{dist}(\mathcal{X}, \tilde{\mathcal{X}}) = L\operatorname{dist}(\mathcal{X}, \tilde{\mathcal{X}}),$$

where L is a constant that is solely dependent on β^*.

D. Proof of Theorem 5.5

Consider two simplicial complexes as defined in Definition 5.3. Let the simplicial scattering embeddings of $\{\mathcal{X}_k\}_{k=0}^K$ in $\mathcal{X}$ (or $\{\tilde{\mathcal{X}}_k\}_{k=0}^K$ in $\tilde{\mathcal{X}}$, respectively) output by SSN be denoted by $\phi(\mathcal{X})$ (or $\phi(\tilde{\mathcal{X}})$). Consider a minor modification to the SSN model, where the embeddings of simplices of a particular order, say k, are now computed without propagating information from $(k+1)$-simplices. Let the output of the simplicial scattering model with such a modification be denoted by $\phi^{\backslash k+1}(\mathcal{X})$ (or $\phi^{\backslash k+1}(\tilde{\mathcal{X}})$). From Theorem 5.4, we know that there exists two constants, L and L_{k+1} that satisfy

$$\left\|\phi(\mathcal{X}) - \phi(\tilde{\mathcal{X}})\right\|_2^2 \leq L\operatorname{dist}(\mathcal{X}, \tilde{\mathcal{X}})$$

and

$$\left\|\phi^{\backslash k+1}(\mathcal{X}) - \phi^{\backslash k+1}(\tilde{\mathcal{X}})\right\|_2^2 \leq L_{\backslash k+1}\operatorname{dist}(\mathcal{X}, \tilde{\mathcal{X}}). \tag{10}$$

The objective here is to show $L \leq L_{\backslash k+1}$, which implies that the bound corresponding to simplicial scattering that uses information from $k+1$ simplices to assign embeddings to simplices is tighter than simplicial scattering that does not use $(k+1)$-simplices.

As in Appendix B, we can represent $\phi(\mathcal{X})$ using $\mathbf{T}$ defined in (9). We can define a similar matrix, denoted by $\mathbf{T}_{\backslash k+1}$, which

compactly represents the diffusion operators that result in the simplicial scattering embeddings $\phi^{\backslash k+1}(\mathcal{X})$ as

$$\mathbf{T}_{\backslash k+1} = \begin{bmatrix} \mathbf{I}_{n_0} + \mathbf{P}_0^{\mathcal{U}} & \mathbf{P}_1^{\mathcal{C}} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{P}_1^{\mathcal{B}} & \mathbf{I}_{n_1} + \mathbf{P}_1^{\mathcal{U}} + \mathbf{P}_1^{\mathcal{L}} & \mathbf{P}_2^{\mathcal{C}} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{P}_k^{\mathcal{B}} & \mathbf{I}_{N_k} + \mathbf{P}_k^{\mathcal{L}} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{P}_{k+1}^{\mathcal{B}} & \mathbf{I}_{n_{k+1}} + \mathbf{P}_{k+1}^{\mathcal{L}} + \mathbf{P}_{k+1}^{\mathcal{U}} & \mathbf{P}_{k+1}^{\mathcal{C}} & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{P}_{k-1}^{\mathcal{B}} & \mathbf{I}_{n_K} + \mathbf{P}_K^{\mathcal{L}} \end{bmatrix}.$$

Here, the random walk matrices corresponding to information aggregation from upper and coboundary adjacent simplices of k-simplices, i.e., aggregations that depend on $k+1$ simplices are ignored. Similarly, we can define $\tilde{\mathbf{T}}$ and $\tilde{\mathbf{T}}_{\backslash k+1}$ by replacing random walk matrices in $\mathbf{T}$ and $\mathbf{T}_{\backslash k+1}$, respectively, with random walk matrices constructed using the boundary matrices of $\tilde{\mathcal{X}}$. With $\mathbf{T}$, $\mathbf{T}_{\backslash k+1}$, $\tilde{\mathbf{T}}$, and $\tilde{\mathbf{T}}_{\backslash k+1}$ as defined above, we observe that

$$\begin{aligned} \mathbf{T} &= \mathbf{T}_{\backslash k+1} + \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{P}_k^{\mathcal{U}} & \mathbf{P}_{k+1}^{\mathcal{C}} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \end{bmatrix} \\ &= \mathbf{T}_{\backslash k+1} + \bar{\mathbf{T}}_{\backslash k+1}. \end{aligned} \tag{11}$$

Notice that the stability bound given by (10) increases with increasing $\beta^* = \max(\beta_1, \tilde{\beta}_1)$. Therefore, proving $L \leq L_{\backslash k+1}$ is equivalent to showing

$$\begin{aligned} \beta(\mathbf{T})^* &= \max(1 - \lambda_1(\mathbf{T}), 1 - \tilde{\lambda}_1(\mathbf{T})) \\ &\geq \beta(\mathbf{T}_{\backslash k+1})^* = \max(1 - \lambda_1(\mathbf{T}_{\backslash k+1}), 1 - \tilde{\lambda}_1(\mathbf{T}_{\backslash k+1})), \end{aligned}$$

where $\lambda_1(\mathbf{A})$ and $\beta(\mathbf{A})$ denote the second largest eigenvalue and the spectral gap of a matrix $\mathbf{A}$, respectively.

Notice that $\bar{\mathbf{T}}_{\backslash k+1}$ is a block upper triangular matrix with the only block along the diagonal being $P_k^{\mathcal{U}}$. The smallest eigenvalue of $\bar{\mathbf{T}}_{\backslash k+1}$ is thus 0. Using Weyl's inequality and the bound on the smallest eigenvalue of $\bar{\mathbf{T}}_{\backslash k+1}$, denoted by $\lambda_{n_k}(\bar{\mathbf{T}}_{\backslash k+1})$, we can relate the second largest eigenvalues of $\mathbf{T}_{\backslash k+1}$ and $\mathbf{T}$. Specifically, we have

$$\lambda_1(\mathbf{T}) - \lambda_1\left(\mathbf{T}_{\backslash k+1}\right) \geq \lambda_N(\bar{\mathbf{T}}_{\backslash k+1}) = 0. \tag{12}$$

Similarly, we could decompose $\tilde{\mathbf{T}}$ as $\tilde{\mathbf{T}} = \tilde{\mathbf{T}}_{\backslash k+1} + \bar{\tilde{\mathbf{T}}}_{\backslash k+1}$ and prove that

$$\lambda_1\left(\tilde{\mathbf{T}}\right) - \lambda_1\left(\tilde{\mathbf{T}}_{\backslash k+1}\right) \geq \lambda_N(\bar{\tilde{\mathbf{T}}}_{\backslash k+1}) = 0. \tag{13}$$

where $\lambda_N(\bar{\tilde{\mathbf{T}}}_{\backslash k+1})$ denotes the smallest eigenvalue of $\bar{\tilde{\mathbf{T}}}_{\backslash k+1}$. From (12) and (13), we have

$$\begin{aligned} \beta(\mathbf{T})^* &= \max\left(1 - \lambda_1(\mathbf{T}), 1 - \lambda_1\left(\tilde{\mathbf{T}}\right)\right) \\ &\leq \max\left(1 - \lambda_1\left(\mathbf{T}_{\backslash k+1}\right), 1 - \lambda_1\left(\tilde{\mathbf{T}}_{\backslash k+1}\right)\right) \\ &= \beta(\mathbf{T}_{\backslash k+1})^*. \end{aligned} \tag{14}$$

This, from Appendix C, proves that $L \leq L_{\backslash k+1}$, i.e., the simplicial scattering embeddings $\phi(\mathcal{X})$ are more stable than $\phi^{\backslash k+1}(\mathcal{X})$.

E. Details Related to Experiments

Datasets. The datasets used in the experiments and the number of simplices of various orders in them are summarized in Table 6. These include `high-school`, `primary-school`, `senate-bills`, `email-Enron`, `madison-reviews`,

Dataset	Simplex	#0-simplicies	#1-simplicies	#2-simplicies	Order
high-school	Group of people	327	5818	2370	3
primary-school	Group of people	242	8317	5139	3
senate-bills	Co-sponsors	294	6974	3013	3
email-Enron	Email groups	142	1655	6095	3
madison-reviews	Users who reviewed	566	2916	4962	10
algebra-questions	Users who answered questions	584	3084	8444	9
geometry-questions	Users who answered questions	429	4296	13894	9

Table 6: Dataset statistics.

Method	Type	20-80	40-60	60-40	80-20
GCN	S	68.30 ± 3.05	72.33 ± 2.67	74.33 ± 4.08	76.12 ± 2.89
ScGCN	S	23.5 ± 1.4	23.8 ± 1.2	25.1 ± 1.5	29.2 ± 2.2
GSAN	S	22.9 ± 1.2	25.4 ± 1.8	25.2 ± 1.9	28.4 ± 2.1
GSN	US	26.3 ± 2.5	28.2 ± 2.3	29.4 ± 2.7	31 ± 3.8
SCNN	S	55.57 ± 5.44	62.53 ± 2.64	66.53 ± 6.90	68.14 ± 3.18
MPSN	S	57.01 ± 2.87	64.59 ± 2.47	67.73 ± 3.60	72.65 ± 4.58
SSN	US	**62.8 ± 3.8**	**71 ± 3.2**	**75.2±3.5**	**85.6±4**

(a) Node classification accuracies on primary-school

Method	Type	20-80	40-60	60-40	80-20
GCN	S	91.6 ± 1	94.2 ± 1.2	94.6 ± 1.1	95.5 ± 1.7
GSN	US	46.7 ± 1.2	52.4 ± 2	54.2 ± 1.6	55.6 ± 2.1
SCNN	S	78.4 ± 1.6	84.7 ± 1.9	89.1 ± 1.9	89.9 ± 2.9
MPSN	S	79.8 ± 1.8	88.1 ± 2.2	91 ± 1.6	94 ± 2.4
SSN	US	**92.9 ± 1**	**96.4 ± 1.2**	**96.5 ± 1**	**96.3 ± 1**

(b) Simplex classification accuracies on madison-reviews

Method	Type	20-80	40-60	60-40	80-20
GCN	S	67.2 ± 1.5	70.0 ± 1.7	71.1 ± 2.5	72.1 ± 2.5
GSN	US	39.2 ± 2.4	41.8 ± 3	42.6 ± 2.2	43.2 ± 2.3
SCNN	S	72 ± 1	76 ± 4	76.4 ± 1	79.1 ± 1.8
MPSN	S	**73.2 ± 1.8**	**79.3 ± 1.3**	**81.7 ± 2.1**	**82.3 ± 2.1**
SSN	US	71.1 ± 1.7	74.7 ± 1.6	76.3 ± 1.0	79.9 ± 2.1

(c) Simplex classification accuracies on geometry-questions

Table 7: Experimental results with partially labeled data.

algebra-questions, and geometry-questions. In the high-school and primary-school datasets, a simplex represents a group of individuals in close proximity, with the classes denoting the locations of the students. For the senate-bills dataset, a simplex is a group of individuals co-sponsoring bills in the Senate, with the class indicating the political party of the sponsors. In the email-Enron dataset, a simplex signifies a group of people in an email group. In the madison-reviews dataset, a simplex is a group of people who have reviewed a certain type of restaurant, and the simplex class is the type of the restaurant. Lastly, in the algebra-questions and geometry-questions datasets, simplices are groups of people who answered a certain type of question, with the simplex labels being the question tags.

Feature Initialization. To initialize features for the simplices of the simplicial complexes, we adopt the procedure outlined in (Madhu & Chepuri, 2023). This procedure is based on anchor nodes and is used for the node classification, simplicial closure, and simplex prediction tasks. For the task of trajectory prediction, the initial features are already provided with the dataset. In the case of graph classification, we utilize the degrees of the nodes as features. The features of a simplex are initialized as the sum of the features of the nodes that constitute the simplex.

Setup. To ensure a fair comparison, we standardize the use of two message passing layers for all neural models, which include SNNs and GNNs. However, we adjust the hidden dimensions and learning rate for each encoder to optimize performance. The encoder parameters are optimized using the Adam optimizer. For SSN, we employ two scattering layers and experiment with different Js, ranging from 1 to 10. We present the best performance achieved using all Js. The entire process, which includes reading the data, running feature extraction for all Js, applying these features for any downstream application, and evaluating system performance, takes approximately 1 to 2 minutes. This highlights the efficiency of SSN in handling these tasks.

F. Additional Experiments

To demonstrate the effectiveness of SSN in low-resource settings, we conduct an experiment where we first extract features using SSN and then partition the data into different splits, such as $20-80$, $40-60$, $60-40$, and $80-20$. Here, the split notation a-b represents $a\%$ of the data used for training and $b\%$ for testing the linear classifier trained at the end. We carry out node classification on the primary-school dataset and simplex prediction on the madison-reviews and

	high-school	primary-school	senate-bills
$J = 1$	62.58 ± 5.83	34.29 ± 3.85	67.63 ± 4.51
$J = 2$	72.58 ± 6.34	45.71 ± 4.85	75.25 ± 5.32
$J = 3$	87.12 ± 4.56	53.47 ± 3.51	75.42 ± 4.63
$J = 4$	94.85 ± 2.46	65.10 ± 5.43	78.31 ± 3.77
$J = 5$	95.45 ± 2.35	73.47 ± 4.56	80.34 ± 3.14
$J = 6$	96.36 ± 2.27	77.14 ± 5.90	78.98 ± 3.73
$J = 7$	98.03 ± 2.04	82.45 ± 5.79	80.34 ± 4.30
$J = 8$	$\underline{98.33 \pm 1.06}$	87.76 ± 2.41	$\mathbf{83.22 \pm 3.90}$
$J = 9$	$\mathbf{99.39 \pm 0.74}$	$\underline{91.22 \pm 2.05}$	$\underline{81.53 \pm 6.80}$
$J = 10$	$\mathbf{99.39 \pm 0.74}$	$\mathbf{91.43 \pm 2.20}$	80.68 ± 2.86

Table 8: Results with varying J on node classification datasets

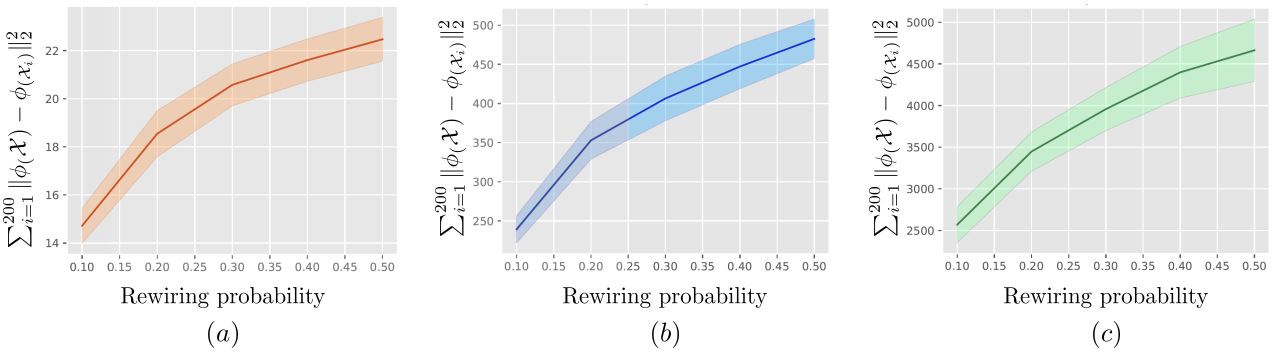

Figure 3: Stability of SSN for varying degrees of perturbations in simplicial complexes.

geometry-questions datasets. As depicted in Tables 7a, 7b, and 7c, performance improves as the size of the training data increases. Notably, SSN significantly outperforms SNNs in scenarios with less labeled data (20-80 and 40-60 splits), indicating that SSN is a superior choice when there is a scarcity of labeled data.

We demonstrate the importance of the maximum scale J through comparison across different J values, ranging from 1 to 10, as shown in Table 8. The results reveal that performance significantly improves with an increase in J at lower scales, but it eventually stagnates as J continues to increase.

We demonstrate the stability of SSN in Fig. 3, where we conduct a numerical experiment on a small-world graph (Watts & Strogatz, 1998) consisting of 200 nodes. Following this, we transform the graph into a simplicial complex, $\mathcal{X}$, using the clique lifting method. We then generate an additional 200 graphs and convert them into 200 simplicial complexes, denoted as $\{\mathcal{X}_i'\}_{i=1}^{200}$, using the same procedure. We employ SSN to extract embeddings from these simplicial complexes and compute the distance between simplicial embeddings of $\mathcal{X}$ and each of the $\mathcal{X}_i'$. We then calculate the mean of all these distances. We repeat the procedure for incrementally increasing rewiring probabilities and plot the mean distances as a function of rewiring probabilities. As the rewiring probability increases, the simplicial diffusion distance increases, resulting in less stable representations.