Non-Euclidean Spatial Graph Neural Network

Anonymous Author(s)

ABSTRACT

Spatial networks are networks whose graph topology is constrained by their embedded spatial space. Understanding the coupled spatial-graph properties is crucial for extracting powerful representations from spatial networks. Therefore, merely combining individual spatial and network representations cannot reveal the underlying interaction mechanism of spatial networks. Besides, existing spatial network representation learning methods can only consider networks embedded in Euclidean space, and can not well exploit the rich geometric information carried by irregular and non-uniform non-Euclidean space. In order to address this issue, in this paper we propose a novel generic framework to learn the representation of spatial networks that are embedded in non-Euclidean manifold space. Specifically, a novel message-passing-based neural network is proposed to combine graph topology and spatial geometry, where spatial geometry is extracted as messages on the edges. We theoretically guarantee that the learned representations are provably invariant to important symmetries such as rotation or translation, and simultaneously maintain sufficient ability in distinguishing different geometric structures. The strength of our proposed method is demonstrated through extensive experiments on both synthetic and real-world datasets.

KEYWORDS

spatial networks, graph neural networks, geometric deep learning

ACM Reference Format:
Anonymous Author(s). 2023. Non-Euclidean Spatial Graph Neural Network. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Spatial networks are types of networks whose nodes and edges are embedded in a geometric spatial manifold. Spatial networks are ubiquitous in the real world, such as biological neural networks [15], transportation networks [17], and mobility networks [9], where the spatial and network properties are deeply coupled together. For example, railroads are often built according to the terrain to achieve the lowest construction cost [57]. As shown in Figure 1, understanding the mechanism of network interactions and their embedded spatial constraints, along with their coupled interaction, is crucial for learning effective representations of spatial networks.

Although many efforts [1, 3, 4, 10, 16, 68] have been put toward understanding the mechanism of spatial networks in some traditional research domains such as physics or mathematics, they usually require predefined human heuristics and prior knowledge of the analytical formulation of embedded spatial manifolds, which is usually unavailable in many real-world cases. In the era of deep learning, existing representation learning works on spatial networks [38, 58, 72] can only consider networks that are embedded in Euclidean space, where edge connections between nodes are described as straight lines. However, many real-world networks are embedded in non-Euclidean spaces, such as manifolds. The over-simplified approximations in flat Euclidean space will inevitably lose the rich geometric information carried by the irregular manifolds. Examples in Figure 2 that share the same network topology and nodes' spatial coordinates, respectively, but with significantly different connecting curves between nodes, are non-distinguishable for existing representation learning methods on spatial networks. Therefore, jointly taking the irregularity of the embedded manifold with the network topology into account is crucial to extract powerful representations for spatial networks.

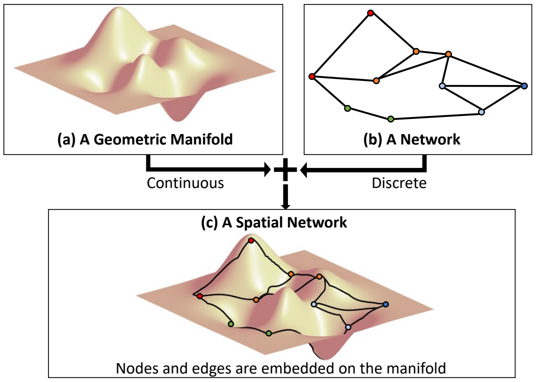

Figure 1: Spatial network contains not only network topology information but also their interaction with the embedded spatial surface.

Despite the respective progress in representation learning on network data and spatial data respectively, the representation learning for spatial networks has been largely underexplored and has just started to attract fast-increasing attention, especially when considering the spatial space as an irregular non-Euclidean manifold. However, there is no trivial way to simply combine previous representation learning methods on network data and spatial data together to accomplish the task of representation learning on spatial networks due to several unique challenges: (1) **Difficulty in jointly considering discrete network and continuous spatial manifolds information, and their coupled interactions.** As shown in the example in Figure 2, some spatial networks may share the same spatial and network properties, respectively, but have significantly different interaction mechanisms. Simply combining spatial and graphical methods cannot distinguish these spatial networks. (2) **Difficulty in extracting the geometric information of nodes and edges embedded in the irregular manifold.** In

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

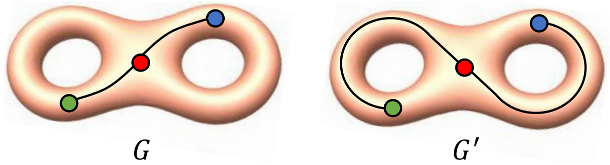

Figure 2: Two spatial networks with different connectivity mechanisms on a holomorphic manifold. The left figure reflects that nodes tend to be connected by the shortest distance (called the first law of geography [64]), while the right figure reflects the spatial pattern in which nodes tend to be connected by circuitous lines. Distinguishing these two spatial networks requires new approaches to jointly consider the spatial curves on the manifold and network topology.

real-world situations, the manifolds that networks embed in are often irregular and inhomogeneous in space, where an explicit analytical form is usually infeasible. Thus, how to represent the geometric information of nodes and edges that are embedded in the manifold is challenging. (3) **Difficulty for the learned representations to maintain rotation and translation invariance and all the geometric information.** Rotation and translation invariance are natural and crucial for many applications of spatial networks [5, 31]. How to learn rotation- and translation-invariant representations that still maintain sufficient ability in distinguishing different geometric patterns is challenging.

In order to address all the above-mentioned challenges, this paper proposes a novel deep framework for learning the representation of spatial networks on non-Euclidean manifolds. Specifically, to jointly consider discrete network information and continuous spatial information, we propose a novel message-passing neural network to handle the interaction of spatial and network information. Specifically, we first discretize continuous manifolds to discrete mesh tessellation, and then extract spatial curve information as edge embeddings through a sequential model on discretized units. To handle the irregularity of spatial manifolds, we characterize several geometric features from each unit on the curvilinear path. We then give the theoretical proof to ensure the learned representation is invariant to rotation and translation transformations. Furthermore, we demonstrate that our extracted geometric features on the spatial curves have bounded approximation errors, suggesting that the expressive power of the learned representations is guaranteed. To demonstrate the strength of our theoretical findings, extensive experiments are performed on both synthetic and real-world datasets.

2 RELATED WORK

Spatial Networks. There has been a long time of traditional research efforts on the subjects of spatial networks [5]. Back to the field of quantitative geography forty years ago, [8, 26] discovered the relevance of spatial constraints in the formation and evolution of networks, and developed models to characterize spatial networks. Advances in complex networks led to new insights regarding modern quantitative solutions [1, 3, 4, 10, 16, 68], appearing in more practical fields such as transportation networks [2, 40, 41], mobility networks [9, 13], biological networks [15, 59], and computational chemistry [22, 38, 58]. However, most of the existing works focused on studying the Euclidean space networks and few attempts on non-Euclidean space require human domain knowledge about the analytical formulation of embedded space, which is unrealistic for many real-world cases.

Geometric deep learning. In the deep learning era, many research works tried to extend standard deep learning methods to geometric data such as manifolds and graphs [7].

(i) *Geometric Deep Learning on Manifolds.* Existing deep learning methods on manifolds can be categorized according to the way they describe 3D data formats [55, 61, 69]. 3D ShapeNets[69] represents a geometric 3D shape as a probability distribution on a 3D voxel grid. Su et al. [61] construct multi-view representations of 3D shapes by using multiple images from different viewpoints. The main disadvantage of these methods is the efficiency of representing 3D data. Models are difficult to scale because they require much higher complexity. Alternatively, there exists another stream of works that represents manifold data by describing the 3D shape surface [7, 45]. Pioneering works [48] generalize convolutional networks to non-Euclidean manifolds with a local geodesic system. Following works continue exploring the architecture of convolutions such as using localized spectral convolutional networks [49, 71], introducing pseudo-coordinate systems [45, 47], or training CNNs with symmetric positive definite (SPD) matrix [12, 33, 67].

(ii) *Geometric Deep Learning on Graphs.* The earliest work we observed that generalizes convolution layers to graphs is M.Gori et al. [24]. Latterly, the following work [39] successfully built convolutional deep learning architectures on graphs, leading to increased interest in the field. Now there exist two significant streams of graph convolution. The first stream is spectral-based. Noticeable models such as ChebNet, AGCN, and DGCN utilize spectral graph filters and their variants [37, 43, 44, 63, 73]. A main disadvantage of spectral approaches is that the spectral definition of convolution relies on the domain-dependent Fourier basis [7]. It is difficult to transfer a model learned on one graph to another graph with a different Fourier basis. The other stream of work is spatial-based, which directly performs convolution operations based in a message-passing manner. Methods such as GCN and GIN extract and aggregate neighborhood information for convolution [20, 28, 50, 52, 56, 70].

Although great success has been achieved towards geometric deep learning on graphs and manifolds separately, limited existing works can jointly consider them and their interactions. Few existing attempts are either domain-specific [22] or can only consider the Euclidean space [38, 58, 72].

3 PROBLEM FORMULATION

In this section, we first formalize spatial networks, and the problem of representation learning on spatial networks, then we introduce the challenges in order to solve this problem.

Spatial graphs (also known as spatial networks [5]) are networks for which the nodes and edges are embedded in a geometric manifold surface. In this paper, we consider the connected smooth compact two-dimensional surface M, which is most commonly observed in our real-world 3D space. Locally around each point x the manifold is homeomorphic to a two-dimensional Euclidean space referred to as the tangent plane and denoted by T_xM. Given the manifold M, a spatial network is typically defined as $G_M = (V, E, M_V, M_E)$ such that V is the set of nodes and $E \subseteq V \times V$

is the set of edges. $e_{ij} \in E$ is an edge connecting nodes v_i and $v_j \in V$. M_V and M_E denote the subset of the manifold M that nodes and edges embed in, which is defined as $M_V \subseteq M, M_E \subseteq M$. Particularly, M_V can be described as a set of 3D Cartesian coordinate points where we have $p_i \in M$ for each point p_i representing the coordinates of node v_i. M_E can be described as a set of curved lines that connect the nodes on the manifold, where for each $e_{ij} \in E$ we have its corresponding curved line as $l_{ij} \in M_E$. Specifically, a smooth curved line can be defined as a mapping function $l : [0, T] \rightarrow M_E$.

The main goal of this paper is to learn the representation mapping function $f : G_M \rightarrow \mathbb{R}^{\mathcal{D}}$ to map an input spatial network to a high-dimensional vector, with the simultaneous satisfaction of strong discriminative power and significant symmetry properties, which is an extremely hard problem due to several unique challenges: (1) Difficulty in jointly considering discrete network and continuous spatial manifolds information, and their coupled interactions. (2) Difficulty in extracting the geometric information of nodes and edges embedded in the irregular manifold. (3) Difficulty for the learned representations to maintain rotation and translation invariance and all the geometric information. The set of *rotation and translation invariant functions* on the spatial network is defined as $f(G_M) = f(\mathcal{T}(G_M)) = f((V, E, \mathcal{T}(M_V), \mathcal{T}(M_E)))$, for all $\mathcal{T} \in \mathrm{SE}(3)$, where $\mathrm{SE}(3)$ is the continuous Lie group of rotation and translation transformations in $\mathbb{R}^3$.

4 METHOD

In order to design an effective method for learning powerful representations of spatial networks by addressing the above-mentioned challenges, we propose a novel method named **M**anifold **S**pace **G**raph **N**eural **N**etwork (**MSGNN**). To jointly learn network information and its embedded spatial manifold information, we propose a general learning message-passing framework. As shown in Figure 3(a), in order to represent a continuous curve on a manifold, we first discretize the manifold into a mesh tessellation, and then learn the representation of curves through a sequential model of mesh units. Curve representations are then treated as messages on edges, and coupled spatial graph information is learned by passing and aggregating messages to nodes with graph convolutional layers. As shown in Figure 3(b), to deal with the irregularities of spatial curves and their embedded geometric manifolds, we propose to characterize several geometric features on each mesh unit of the curvilinear paths. Finally, we theoretically prove that the extracted spatial curve representations with a guarantee on the properties of *rotation-invariant*, *translation-invariant*, and *geometric information-lossless*

In the rest of this section, we first introduce the details of the sequence representation of manifold-constrained spatial path in subsection 4.2. Then in subsection 4.1, we present the design details of our whole framework to jointly combine network information and spatial path information. Finally, in subsection 4.3, we give the theoretical proof of the expressive power on learned representations and associated approximation bound.

4.1 Framework

In this section, we first propose the overall framework that can jointly consider the network topology and its embedded spatial space information. Although there is a large amount of existing work on learning representations on network and spatial data separately, simply combining the network and spatial representations can not handle the coupled spatial graph information. Few existing works [38, 58, 72] on learning spatial networks representations only consider the spatial structure of nodes by considering their Euclidean coordinates. While for the edge connection between nodes, they simply treat it as a straight line connecting nodes in Euclidean space. Although these methods could achieve relatively good performance when the embedded space can be well approximated by Euclidean space, they can not well handle irregular and non-uniform embedding space, where the Euclidean space approximation will deviate largely from the real cases. Moreover, simply using Euclidean space approximation can not reflect the true geometry of the connection paths between nodes on the manifold surface, which will inevitably hurt the expressiveness of the learned representation. Thus, to fill this gap, in this paper we propose the first general framework to deal with spatial networks embedded in non-Euclidean manifold space.

Due to the incompatibility between discrete network data and continuous spatial data, the first question is how to combine these two data in one end-to-end framework. Besides, the explicit analytical format function of the spatial manifold is usually extremely difficult to obtain because of the irregularity and non-uniformity of the real-world manifold surfaces. To address this problem, we first propose to discretize the continuous manifold space into discrete mesh data. Triangular mesh is a commonly used data format to provide an efficient way to approximate manifold shapes because they explicitly preserve shape surfaces and topology. Specifically, a triangular mesh can be defined as a collection of C triangle faces $\mathcal{F} = \{f^{(1)}, f^{(2)}, \ldots, f^{(C)}\}$, where the vertices of each $f^{(c)} \in \mathcal{F}$ are located on the surface of manifold M.

Given the discretized mesh to describe the embedded spatial surface, we can describe the spatial curved lines between nodes as a sequence of discrete units. As shown in Figure 3(b), the sequence consists of triangular faces that the curve line passes. Therefore, the spatial information of each edge can be represented as a sequence of mesh units and line segments embedded on them. More concretely, for each edge $e_{ij} \in E$ exists in the given spatial graph, it corresponds to a curved spatial line l_{ij} embedded on manifold M. Given the discretized triangular mesh, the embedded line l_{ij} can be represented as a set of K line segments $l_{ij} = \{l_{ij}^{(1)}, l_{ij}^{(2)}, \ldots, l_{ij}^{(K)}\}$ with their corresponding embedded triangle faces $F_{ij} = \{f_{ij}^{(1)}, f_{ij}^{(2)}, \ldots f_{ij}^{(K)}\}$, where K is the number of faces on this spatial path and each line segment $l_{ij}^{(k)}$ is embedded in its corresponding face $f_{ij}^{(k)}$. In summary, we can represent the spatial information on edge e_{ij} as a sequence of pairs $((l_{ij}^{(1)}, f_{ij}^{(1)}), (l_{ij}^{(2)}, f_{ij}^{(2)}), \ldots, (l_{ij}^{(K)}, f_{ij}^{(K)}))$.

Given the sequences of pairs of faces and their embedded line segments, the next question is to incorporate them with the graph topology in a general model to learn the coupled spatial-graph representations. Since the length of the sequence may vary on different edges, some common methods such as MLP or CNN cannot be easily generalized to extract representations here. To address this issue, we propose a novel approach that mimics natural language processing approaches by analogizing each pair of spatial units as a token in a sentence, which is shown in Figure 3(b). Therefore, a recurrent

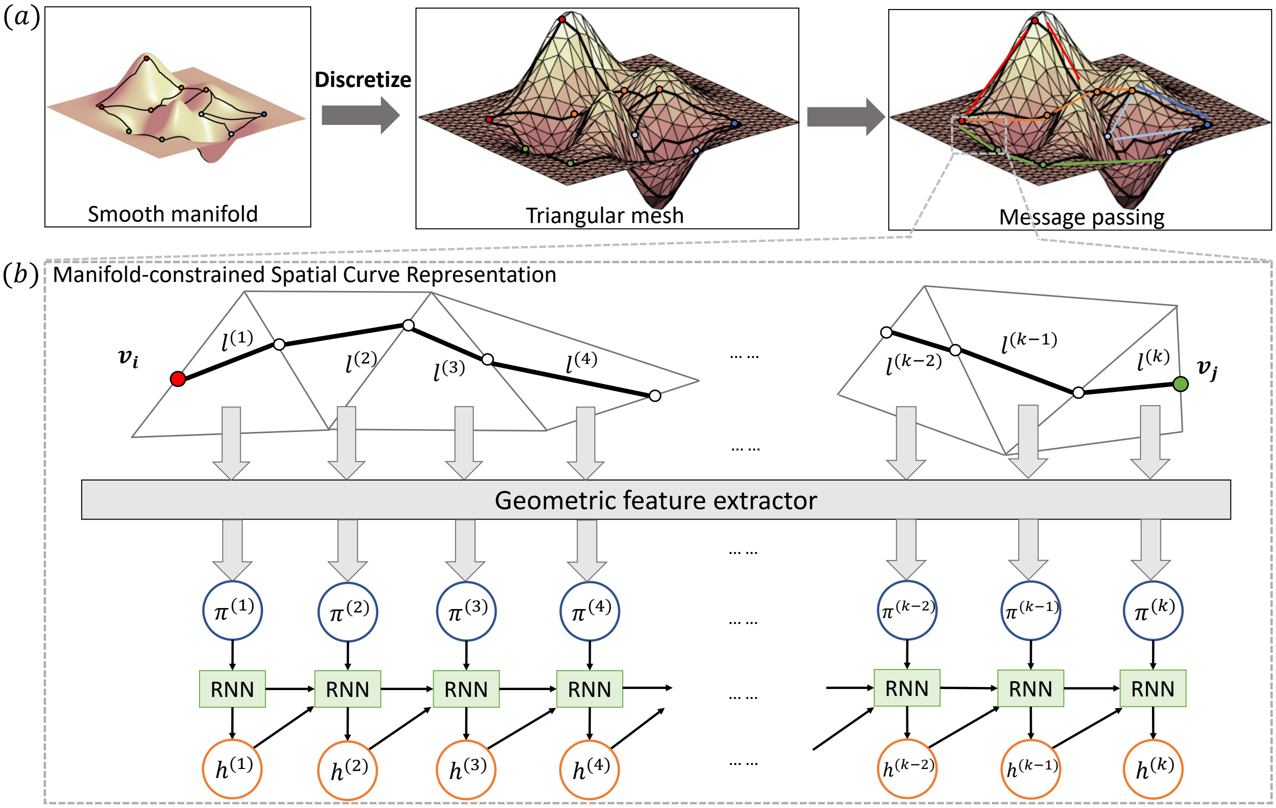
(a)
Discretize
Smooth manifold
Triangular mesh
Message passing
(b)
Manifold-constrained Spatial Curve Representation
v_i
$l^{(1)}$
$l^{(2)}$
$l^{(3)}$
$l^{(4)}$
$l^{(k-2)}$
$l^{(k-1)}$
$l^{(k)}$
v_j
Geometric feature extractor
$\pi^{(1)}$ $\pi^{(2)}$ $\pi^{(3)}$ $\pi^{(4)}$ $\pi^{(k-2)}$ $\pi^{(k-1)}$ $\pi^{(k)}$
RNN
$h^{(1)}$ $h^{(2)}$ $h^{(3)}$ $h^{(4)}$ $h^{(k-2)}$ $h^{(k-1)}$ $h^{(k)}$
……

Figure 3: Illustration of the overall proposed framework. (a) The discretization process of the continuous manifold and convolutional neural networks for passing and aggregating the geometric information on spatial curves. (b) The recurrent neural network (RNN) module extracts the geometric information along the irregular spatial curves between nodes.

graph neural network (RNN) model such as GRU or LSTM is a natural choice to extract latent embeddings from these sequences. Formally, the extracted latent embedding $h(e_{ij})$ on edge e_{ij} can be denoted as $\text{RNN}(\pi(l_{ij}^{(1)}, f_{ij}^{(1)}), \pi(l_{ij}^{(2)}, f_{ij}^{(2)}), \ldots, \pi(l_{ij}^{(K)}, f_{ij}^{(K)}))$, where $\pi(\cdot)$ denotes the geometric information of the unit and will be introduced in Section 4.2. The extracted spatial information can then be treated as the message on graph edges. A graph message passing neural network model is then performed to jointly combine node information and all incoming messages on edges into updated node embeddings, where the update function is as follows:

$$\begin{aligned}\hat{h}(v_i) &= \text{AGGREGATE}\{\xi(h(v_j), h(e_{ij}))|j \in \mathcal{N}(i)\},\\ h(e_{ij}) &= \text{RNN}(\pi(l_{ij}^{(1)}, f_{ij}^{(1)}), \pi(l_{ij}^{(2)}, f_{ij}^{(2)}), \ldots, \pi(l_{ij}^{(K)}, f_{ij}^{(K)})),\end{aligned} \tag{1}$$

where h represents latent embeddings and ξ denotes a nonlinear transformation function such as multiple layer perceptron (MLP). AGGREGATE denotes any feasible set aggregation function.

4.2 Manifold-constrained spatial curve representation

Given the above framework, a key question is how to define the spatial information extractor $\pi(\cdot)$ on each unit of a line segment and its embedded triangular mesh. As mentioned previously, we need a novel way to represent the spatial path information of the network edge that can preserve all the geometric shape information, and simultaneously maintain rotation- and translation-invariance. Obviously, simply feeding the Cartesian coordinates of each line segment in units can not guarantee invariance to the important symmetries such as rotation-and translation-transformations. Although there exist a few spatial information representation methods in the domain of spatial deep learning that can guarantee rotation and translation invariant features, we can not directly use them because they either can not capture the coupled spatial-graph properties because they are purely spatial-based methods, or can not guarantee all the geometric structure information is preserved because the information they extracted is not lossless. Here, the term lossless information means given the extracted geometric features, the information is sufficient to recover the input geometric structure. To handle this issue, for each sequence of spatial path line segments and their embedded meshes, we propose to extract a combination of geometric features on each mesh unit and the relative spatial relationship with their neighboring units. We theoretically guarantee the extracted information is sufficient to recover the full original geometries and also stay invariant to rotation and translation transformations.

Without loss of generality, we consider the spatial path between node v_i and node v_j in the given node set V of graph that there exists an edge e_{ij} between them. Formally, for each unit $(l_{ij}^{(k)}, f_{ij}^{(k)})$

belongs to the sequence of edge e_{ij}, where $k \in [1, K]$, we use $\pi(l_{ij}^{(k)}, f_{ij}^{(k)})$ to represent the extracted geometric features on this unit. The sequence of spatial information on edge e_{ij} is denoted as $((l_{ij}^{(1)}, f_{ij}^{(1)}), (l_{ij}^{(2)}, f_{ij}^{(2)}), \ldots, (l_{ij}^{(K)}, f_{ij}^{(K)}))$. For the purpose of simplicity, we omit the subscript symbol $_{ij}$ in the rest of this section.

In order to extract the necessary spatial information, we need to extract both the spatial information on the line segment and its embedded triangle mesh. Specifically, we need to extract sufficient features to preserve the geometric structure of spatial curves and the relative orientation directions of embedded triangle faces. For simplicity, we use $\pi(l^{(k)}), \pi(f^{(k)})$ to represent the spatial information on the line segment and its embedded mesh face, respectively. For each line segment $l^{(k)}$, we first extract their length $d^{(k)}$ as the distance between its two endpoints, which can confine the relative position to a sphere in 3D space. Then the angle $\theta^{(k)}$ between line segment $l^{(k)}$ and the directly connecting line L between nodes v_i and v_j is extracted to further locate a position to a circle on the sphere. Then the line segment can only rotate around the line L. In order to fix the relative position, we further extract the relative torsion angles $\phi^{(k,k-1)}$ and $\phi^{(k,k+1)}$ between current line segment $l^{(k)}$ and its pair of neighboring line segments $l^{(k-1)}$ and $l^{(k+1)}$. Besides the geometric structure of line segments, extracting sufficient features on the embedded mesh faces $\pi(f^{(k)})$ is also important to understanding the spatial information of the surface environment. Since the mesh face is an approximated tangent space of the ground truth manifold, the curvature vector direction plays a crucial role in understanding its spatial information. But simply calculating the orientation $\mathbf{n}^{(k)}$ of the curvature vector can not guarantee the rotation- and translation-invariance. Hence, we propose to calculate the relative angles $\varphi^{(k,k-1)}$ and $\varphi^{(k,k+1)}$ between the curvature vector of the given face and its two neighboring faces. However, only the relative angles between the neighboring curvature vectors still can not guarantee the full geometric structure is preserved. In order to solve this issue, we further calculate the angle $\varphi^{(k-1,k+1)}$ between its two neighboring curvature vectors. Thus, three dependent angles will form a triangle in the space, thus the relative orientation will be fixed. In summary, combining the commonly used geometry features distance d, angle θ, torsion angle ϕ and orientation angle φ in the three-dimensional geometry study area, is a data representation that can satisfy rotation- and translation-invariant. Jointly using them can also ensure the geometric structure of spatial curved paths is preserved losslessly.

Mathematically, for the k-th mesh unit on the sequence of spatial path that formed edge e_{ij} between node v_i and v_j, the proposed spatial information representation can be expressed as

$$\begin{aligned}\pi(l^{(k)}, f^{(k)}) =&(d^{(k)}, \theta^{(k)}, \phi^{(k,k-1)}, \phi^{(k,k+1)},\\ &\varphi^{(k,k-1)}, \varphi^{(k,k+1)}, \varphi^{(k-1,k+1)}),\end{aligned} \tag{2}$$

where

$$\begin{aligned}
d^{(k)} &= ||\mathbf{l}^{(k)}||_2, \theta^{(k)} = \arccos(\langle \frac{\mathbf{l}^{(k)}}{d^{(k)}}, \frac{\mathbf{L}_{i,j}}{d_{i,j}} \rangle),\\
\mathbf{L}_{i,j} &= \mathbf{p}_j - \mathbf{p}_i, d_{i,j} = ||\mathbf{L}_{i,j}||_2\\
\phi^{(k,k-1)} &= \langle \frac{\mathbf{c}^{(k-1)} \times \mathbf{c}^{(k)}}{||\mathbf{c}^{(k-1)} \times \mathbf{c}^{(k)}||_2}, \frac{\mathbf{L}_{ij}}{||\mathbf{L}_{ij}||_2} \rangle \cdot \bar{\phi}^{(k,k-1)},\\
\phi^{(k,k+1)} &= \langle \frac{\mathbf{c}^{(k)} \times \mathbf{c}^{(k+1)}}{||\mathbf{c}^{(k)} \times \mathbf{c}^{(k+1)}||_2}, \frac{\mathbf{L}_{ij}}{||\mathbf{L}_{ij}||_2} \rangle \cdot \bar{\phi}^{(k,k+1)},\\
\bar{\phi}^{(k,k-1)} &= \arccos\langle \mathbf{c}^{(k)}, \mathbf{c}^{(k-1)} \rangle, \bar{\phi}^{(k,k+1)} = \arccos\langle \mathbf{c}^{(k)}, \mathbf{c}^{(k+1)} \rangle\\
\mathbf{c}^{(k)} &= \frac{\mathbf{L}_{ij} \times \mathbf{l}^{(k)}}{||\mathbf{L}_{ij} \times \mathbf{l}^{(k)}||_2}, \varphi^{(k-1,k+1)} = \arccos(\langle \mathbf{n}^{(k-1)}, \mathbf{n}^{(k+1)} \rangle),\\
\mathbf{c}^{(k-1)} &= \frac{\mathbf{L}_{ij} \times \mathbf{l}^{(k-1)}}{||\mathbf{L}_{ij} \times \mathbf{l}^{(k-1)}||_2}, \varphi^{(k-1,k)} = \arccos(\langle \mathbf{n}^{(k-1)}, \mathbf{n}^{(k)} \rangle),\\
\mathbf{c}^{(k+1)} &= \frac{\mathbf{L}_{ij} \times \mathbf{l}^{(k+1)}}{||\mathbf{L}_{ij} \times \mathbf{l}^{(k+1)}||_2}, \varphi^{(k,k+1)} = \arccos(\langle \mathbf{n}^{(k)}, \mathbf{n}^{(k+1)} \rangle)
\end{aligned} \tag{3}$$

Theorem 1. *Here the distances $d \in [0, \infty)$, angle $\theta \in [0, \pi)$, torsions $\phi \in [-\pi, \pi)$, and relative orientation angle $\varphi \in [0, \pi)$ are rigorously invariant under all rotation and translation transformations $\mathcal{T} \in \mathrm{SE}(3)$.*

The proof is straightforward and can be found in Appendix.

4.3 Theoretical guarantee on approximation bound

It is remarkable to mention that the proposed representation in Equation 2 not only satisfies the invariance under rotation and translation transformation but also retains the necessary information to recover the geometric structure of original spatial networks under weak conditions, as described in the following theorem.

Theorem 2. *Given a spatial network $G_M = (V, E, M_V, M_E)$, if G_M is a connected graph, then for any edge e_{ij} with spatial curve sequence that has length of sequence $\tau \geq 3$, given Cartesian coordinates of two endpoints and one arbitrary point, the whole Cartesian coordinates of the given spatial networks can be determined by the spatial representation defined in Equation 2.*

The proof for Theorem 2 is a consequence of Lemma 3.

Lemma 3. *Given Cartesian coordinates of two end nodes v_i and v_j, and one arbitrary point $p^{(k-1)}$ in a spatial curve l_{ij}, the Cartesian coordinate $p^{(k)}$ that form line segment $l^{(k)}$ with $p^{(k-1)}$ can be determined by the representation defined in Equation 2.*

As stated in Lemma 3, the Cartesian coordinate of point $p^{(k)}$ can be determined by its connected point and two end nodes. Due to the property of connectivity of graph, we can iteratively determine the coordinate of another connected point to the previous points with known coordinates. Thus, start from an arbitrary point the Cartesian coordinates of whole spatial network is determined. The detailed proof can be found in Appendix.

It is also worth noticing that in our proposed model we represent the spatial path embedded in a manifold as a sequence of line segments and their embedded meshes to handle the continuous spatial data. However, since the mesh data approximates continuous manifold surfaces by many connected discrete triangle surfaces, it would have an approximation error from the ground truth spatial surface. More concretely, in our model, we calculate the spatial path representation in Equation 2 on each mesh unit. A key question is how different the calculated geometric features are from the ground truth, which is critical to the quality of learned representations. In the following, we further give a theoretical analysis of the approximation bound of few calculated spatial geometric features with respect to the size of meshes used. Specifically, we focus on the theoretical guarantee of angle φ, which describes the crucial relative orientation environment of the embedded mesh data.

Before presenting the angle bound between any pair of faces, we introduce some notations: the empty ball size $ebs(p)$ is the radius of the smallest medial ball tangent to M at point p; the local feature size $lfs(p)$ is the distance between point p and the nearest point on M; $|pq|$ means the distance between point p and point q; n_p means an outward-directed vector normal to M at point p. N_p^f is a vector normal to any face f at point p. $\angle(n_p, n_q)$ is the angle between two vectors n_p and n_q. The following definitions are also required for our theoretical analysis:

Definition 1 (Plane Angle). *Because a face is a triangle, then for any face $f_k^{(i,j)}$, a plane angle of $f_k^{(i,j)}$ at vertex $v_k^{(i,j)}$ is the triangle angle associated with vertex $v_k^{(i,j)}$.*

Definition 2 (Circumradius). *For any face $f_k^{(i,j)}$, its diametric ball $B_k^{(i,j)}$ is the smallest closed ball such that $f_k^{(i,j)} \subset B_k^{(i,j)}$ and all vertices of $f_k^{(i,j)}$ lie in the boundary of $B_k^{(i,j)}$. circumcenter and circumradius are the center and radius of $B_k^{(i,j)}$.*

Based on the above notations and definitions, we have the following angle bound:

Theorem 4 (Angle Bound). *Assume M is bounded and smooth. Let $f_{k_1}^{(i_1,j_1)}$ and $f_{k_2}^{(i_2,j_2)}$ be two faces whose vertices lie on M. Let $R_{k_1}^{(i_1,j_1)}$ and $R_{k_2}^{(i_2,j_2)}$ be circumradii (i.e. plural form of circumradius) of $f_{k_1}^{(i_1,j_1)}$ and $f_{k_2}^{(i_2,j_2)}$, respectively. Let $\phi_{k_1}^{(i_1,j_1)}$ and $\phi_{k_2}^{(i_2,j_2)}$ be plane angles of $f_{k_1}^{(i_1,j_1)}$ and $f_{k_2}^{(i_2,j_2)}$ at $v_{k_1}^{(i_1,j_1)}$ and $v_{k_2}^{(i_2,j_2)}$, respectively. Let $\delta = |v_{k_1}^{(i_1,j_1)} v_{k_2}^{(i_2,j_2)}| / lfs(v_{k_1}^{(i_1,j_1)})$. If $\delta < \sqrt{4\sqrt{5}-8}$, then*

$$\begin{aligned}\angle(N_{v_{k_1}^{(i_1,j_1)}}^{f_{k_1}^{(i_1,j_1)}}, N_{v_{k_2}^{(i_2,j_2)}}^{f_{k_2}^{(i_2,j_2)}}) &\leqslant \arccos(1 - \frac{\delta^2}{2\sqrt{1-\delta^2}}) \\ &+ \arcsin(\frac{R_{k_1}^{(i_1,j_1)}}{ebs(v_{k_1}^{(i_1,j_1)})} \max(\cot(\frac{\phi_{k_1}^{(i_1,j_1)}}{2}), 1)) \\ &+ \arcsin(\frac{R_{k_2}^{(i_2,j_2)}}{ebs(v_{k_2}^{(i_2,j_2)})} \max(\cot(\frac{\phi_{k_2}^{(i_2,j_2)}}{2}), 1)) \qquad (4)\end{aligned}$$

where $\angle(N_{v_{k_1}^{(i_1,j_1)}}^{f_{k_1}^{(i_1,j_1)}}, N_{v_{k_2}^{(i_2,j_2)}}^{f_{k_2}^{(i_2,j_2)}})$ is the angle between two vectors $N_{v_{k_1}^{(i_1,j_1)}}^{f_{k_1}^{(i_1,j_1)}}$ and $N_{v_{k_2}^{(i_2,j_2)}}^{f_{k_2}^{(i_2,j_2)}}$ normal to $f_{k_1}^{(i_1,j_1)}$ and $f_{k_2}^{(i_2,j_2)}$ at vertices $v_{k_1}^{(i_1,j_1)}$ and $v_{k_2}^{(i_2,j_2)}$, respectively.

The complete proof of Theorem 4 is shown in the Appendix due to space limit. Theorem 4 shows that the angle bound is defined by the plane angle and the circumradius.

5 EXPERIMENTAL RESULTS

In this section, we first introduce the experimental settings, then the effectiveness of our proposed framework on both synthetic and real-world datasets is presented through a set of comprehensive experiments. All the experiments are conducted on a 64-bit machine with four NVIDIA TITAN Xp GPU. The proposed method is implemented with Pytorch deep learning framework [53]. The link to our code and appendix is at an anonymous GitHub repository[1].

5.1 Experimental Settings

5.1.1 Synthetic datasets. In order to examine the effectiveness of our proposed MSGNN method in learning the coupled network and spatial manifold information, we follow previous works [5] to generate a set of synthetic datasets. We generalize the preferential attachment model [1] to a spatial variant that all nodes and edges are embedded in a defined spatial manifold surface. Specifically, we first randomly generate a manifold surface in 3D space from a designed candidate pool of geometric shapes such as sphere or paraboloid. The process to generate such spatial networks starts from an initial connected network of m_0 nodes that are randomly sampled on the manifold surface. Then we introduce a new node v_j to connect to the existing network at each iteration step. The new node is allowed to make $m \leq m_0$ connections towards existing nodes with a probability $\Pi_{j\rightarrow i} \sim k_i F[d_g(i,j)]$, where k_i is the degree of node v_i and F is an exponential function $F(d_g) = e^{-d_g/r_c}$ of the geodesic distance $d_g(i,j)$ between the newly added node v_j and the node v_i on the manifold. Therefore, the formation mechanism of generated spatial networks is jointly determined by the spatial and network information. General characteristics of spatial networks [5] such as spatial diameter D, and spatial radius r are set as the prediction targets. Besides, we also add the interaction range r_c, which is a significant coupled spatial-graph label that affects the formation of the spatial networks, as another prediction target. We vary the type of embedded manifolds and other parameters of spatial networks to collect 5,000 samples in our synthetic dataset.

5.1.2 Real-world datasets. To further evaluate the performance of our proposed MSGNN and comparison methods in real-world scenarios, five public benchmark real-world spatial network datasets with different application domains are utilized as benchmark datasets in our experiments. Specifically, we include one brain network dataset and two 3D shapes datasets for graph classification task, and two airline transportation networks for link prediction task. We provide a brief description of these datasets as follows:

(1) **HCP brain networks.** Cerebral cortex, which is the outer layer of neural tissue in the human brain, typically presents an geometric surface with many folded peaks and grooves [23, 27]. This irregular structure, which can be described as a two-dimensional manifold surface, plays a crucial role in understanding the underlying connectivity behavior of neurons. Given the brain data, our goal is to classify the subject activity states from the processed functional connectivity (FC) network along with its manifold environment, which is an important task in understanding the underlying connectivity mechanism [42]. Specifically, the data is processed from Magnetic Resonance Imaging (MRI) data of the human connectome

[1] The link to our code and appendix is at https://github.com/anonymous-submitter-kdd/kdd2023manifold_spatial_networks

Target	GCN	GIN	PointNet	PPFNet	MeshCNN	CurvaNet	SchNet	SGMP	MSGNN
r_c	3.35±0.14	2.55±0.18	2.45±0.08	2.68±0.11	2.06±0.13	1.54±0.10	0.97±0.06	1.08±0.05	**0.83±0.04**
D	2.20±0.15	2.73±0.21	1.82±0.06	1.98±0.12	1.87±0.06	1.93±0.11	1.94±0.10	1.86±0.05	**1.67±0.05**
r	2.63±0.14	2.60±0.26	1.95±0.09	2.07±0.09	1.60±0.07	1.74±0.06	1.98±0.07	1.88±0.05	**1.47±0.11**

Table 1: The RMSE results of the synthetic dataset. Smaller value indicates better performance. The best performance for each predictive target is shown in bold, while we also underline the second-best performing models.

project (HCP) [65]. Each data is associated with one activity state (rest, gamble, etc) as the prediction target. The geometric shape of cerebral cortex is segmented into 68 predefined regions of interest (ROIs), which are obtained by applying probabilistic tracking on the diffusion MRI data using the Probtrackx tool from FMRIB Software Library [34]. By following the preprocessing procedure in [66], FC is defined as the Pearson correlation between blood oxygen level-dependent times of two ROIs obtained from resting-state functional MRI data. Then a threshold value is applied to FC to filter out the correlated edges with to construct the brain networks [21, 62], where the spatial trajectories are defined as the geodesic paths between the center of ROIs [11, 60].

(2) **Air transportation networks.** Forecasting the future structure of the air transport network has important implications for the economic integration and development prospects of regions and countries. Link prediction aims to predict whether two airports will be connected by direct flights in the future development stage of flight network [46, 51]. Specifically, we adopt two publicly available flight networks *Flight-NA* and *Flight-GL*, where *Flight-NA* contains 456 airports and 71,959 airlines in the North America [6, 18] and *Flight-GL* [35] contains 3,214 airports and 66,771 airlines spanning the globe. The earth surface is considered as the manifold to include the curved airline trajectory.

(3) **3D shapes classification.** We further conduct experiments on classifying 3D shapes in two datasets *SHREC* and *FAUST*, where *SHREC* contains watertight triangle meshes with 30 classes and *FAUST* contains scans of human poses in a wide range. We follow previous studies [29] to sample a lower resolution (∼500 faces) from a higher solution. The vertices of the lower-resolution triangle tessellation are then treated as the nodes and their geodesic trajectories are treated as the spatial curves.

5.1.3 Comparison models. To the best of our knowledge, there is little previous work that can handle general spatial networks. We compare our proposed MSGNN against several categories of competitive methods, spanning two graph neural networks methods **GCN** and **GIN**, two spatial deep learning methods on point clouds **PointNet** and **PPFNet**, two spatial deep learning methods on mesh **MeshCNN** and **CurvaNet**. We also include two state-of-the-art deep learning methods on Euclidean spatial networks **SchNet** and **SGMP**. For the purpose of a fair comparison, we feed the Cartesian coordinates as node attributes for methods in the class of GNNs. While for the class of spatial deep learning methods on point clouds and mesh, we add graph connectivity information by treating the node embedding from a trained Node2Vec model [25] as additional node attributes. We briefly introduce these methods as follows:

(1) **GCN** [37] is a commonly used GNN model by using a localized first-order approximation of spectral graph convolutions;
(2) **GIN** [70] is a GNN model with provably powerful discriminating power among the class of 1-order GNNs;
(3) **PointNet** [54] is a general deep learning method for representing the global 3D shape as a set of points in Euclidean space;
(4) **PPFNet** [14] is a spatial deep learning framework to learn a globally aware 3D descriptor by constructing point pair features within the local vicinity;
(5) **MeshCNN** [29] is a deep learning method to learn the representations for mesh data. It applies convolutions on mesh edges and the four edges of their incident triangles;
(6) **CurvaNet** [30] is a deep learning method for mesh data by further treating the curvature value of each mesh unit with respect to its surrounding units;
(7) **SchNet** [58] is a deep learning model designed for spatial networks in Euclidean space. It utilizes a continuous filter function on the distances between nodes and their first-order neighbors;
(8) **SGMP** [72] is a deep learning method for spatial networks in Euclidean space. It further includes the directional information by aggregating the length three path messages based on a physical representation of distances, angles, and torsions.

For a fair comparison, we require all models to follow the same search range for model hyperparameters, such as the number of convolutional layers or the dimensionality of hidden embeddings.

5.2 Effectiveness results

5.2.1 Synthetic datasets Results. Here we report the root mean squared error (RMSE) results of our proposed MSGNN with comparison methods on the synthetic dataset in Table 1. We summarize our observations on the model effectiveness below:

(1) The results demonstrate the strength of our proposed MSGNN by consistently achieving the best results in predicting all three coupled spatial-graph targets. Specifically, our model outperformed all the benchmark models by over 38.0% on average, as well as outperformed the second-best model by 17.6% on average.

(2) Our proposed MSGNN method consistently achieves superior performance with respect to all predictive targets, which proves the robustness of MSGNN. In comparison, the spatial neural network methods on point clouds and mesh have significantly different performances on different tasks. For example, they shows competitive performance to spatial network methods on targets spatial diameter D, and spatial radius r. While their performance on predicting interaction range r_c is significantly worse than spatial network methods by over 53.0% on average, which may indicate that simply combining graph and spatial representation can not capture the interactions between these two data sources.

(3) It is also worth noting that the category of methods on spatial networks (SchNet, SGMP, and MSGNN) show a more competitive performance than methods in other categories, by over 34.2% on average, which indicates that either graph neural network or spatial neural network methods have limited capability to effectively learn coupled spatial-graph properties. MSGNN shows a stronger performance compared to other methods on spatial networks by 18.2%

	GCN	GIN	PointNet	PPFNet	MeshCNN	CurvaNet	SchNet	SGMP	MSGNN
HCP	0.835±0.014	0.920±0.007	0.845±0.027	0.876±0.008	0.784±0.031	0.759±0.025	0.896±0.012	0.927±0.004	**0.951±0.005**
Flight-NA	0.674±0.015	0.706±0.006	0.694±0.011	0.698±0.004	0.521±0.023	0.597±0.019	0.710±0.008	0.719±0.004	**0.730±0.005**
Flight-GL	0.722±0.003	0.756±0.014	0.737±0.010	0.715±0.012	0.556±0.032	0.628±0.025	0.750±0.008	0.761±0.009	**0.785±0.005**
SHREC	0.525±0.042	0.533±0.034	0.567±0.007	0.887±0.010	0.910±0.003	0.902±0.004	0.575±0.012	0.896±0.005	**0.918±0.004**
FAUST	0.535±0.010	0.783±0.013	0.905±0.010	0.918±0.005	0.903±0.008	0.923±0.004	0.865±0.023	0.840±0.035	**0.925±0.005**

Table 2: The accuracy results of the real-world datasets. Larger value indicates better performance. The best performance for each predictive target is shown in bold, while we also underline the second-best performing models.

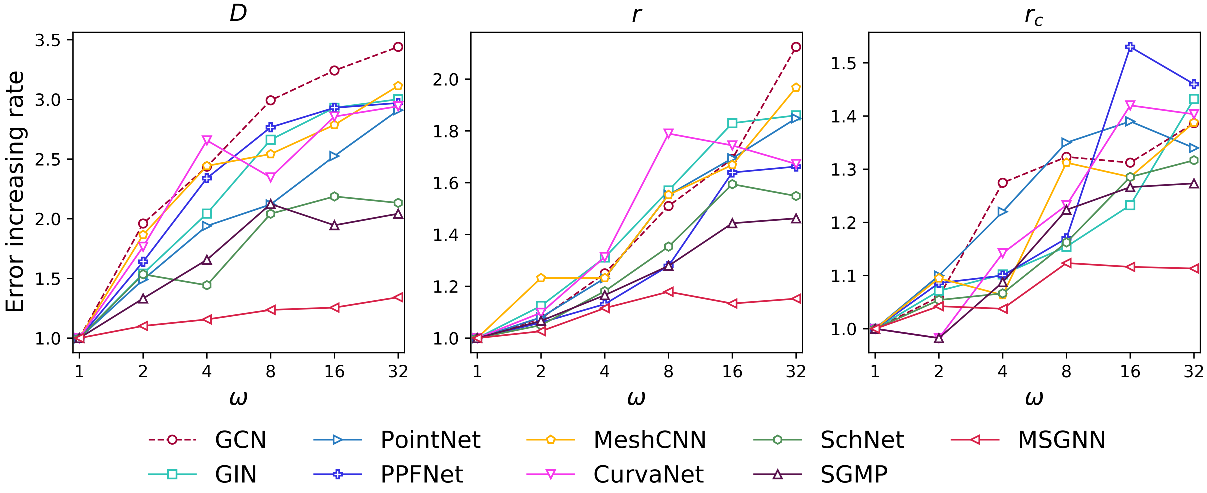

Figure 4: Accuracy trend results for our proposed MSGNN model and all competing models against varying degree of manifold irregularity. The performance of models at the lowest degree of irregularity ($\omega = 1$) is set as the base, and the values in the plot are shown as ratios.

on average, which demonstrates our method takes advantage of the irregular manifold information within the context of the spatial paths to acquire a more competitive performance.

5.2.2 Real-world datasets Results. Here we report the accuracy results of our proposed MSGNN with comparison methods on the real-world dataset in Table 2. We summarize our observations on the model effectiveness below:
(1) Our proposed MSGNN method consistently achieved the best results among all methods in all five real-world datasets. Specifically, our results outperformed all the benchmark models by over 14.1% on average and outperformed the second-best model by 4.2% on average. The superior performance demonstrates the effectiveness of MSGNN for learning powerful spatial network representations in complex real-world scenarios.
(2) In two air transportation networks, our method achieves more considerable performance gains on the global network (Flight-GL) than on the North American network (Flight-NA). One possible reason is that North America is relatively small compared to the globe, and the curved effect on the surface is not significant. This may indicate that our method can exploit the curvature of the embedded surface to further improve the representation ability.
(3) Different classes of methods perform significantly differently on different datasets. For example, the class of spatial neural networks on mesh (MeshCNN and CurvaNet) have achieved competitive results in 3D shapes classification tasks (SHREC and FAUST) by outperforming other benchmark models by 13.1% on average. However, they also performed poorly on air transportation and brain datasets by achieving the worst performance among all classes of methods. Such behavior indicates that these methods can not well handle generic spatial networks.
(4) It is also worth noting that on the SHREC dataset, only methods that consider orientation information achieved competitive results, which may arguably indicate that orientation information is important in the prediction task on this dataset.

5.3 Roubustness Results

5.3.1 Effect of manifold irregularity analysis. Compared to existing representation learning methods on spatial networks, a contribution of our work is that our method can handle irregular geometric manifolds, rather than simply using Euclidean space approximations. To investigate the impact of manifold irregularity on model performance, we further introduce a series of experiments to vary the degree of irregularity of the manifold embedded by the network. Specifically, in our synthetic dataset setting, we choose a sinusoidal surface as the manifold for embedding the network, and we use a frequency parameter ω to control the irregularity of the generated manifold surface. The mathematical formulation of the sinusoidal surface can be written as $z = \sin(\omega\sqrt{x^2 + y^2})$. Larger values of ω here indicate that the resulting manifold surface will have a larger degree of irregularity. We vary the value of ω from 1 to 32 to generate a total of 6 datasets, and we compare the performance of

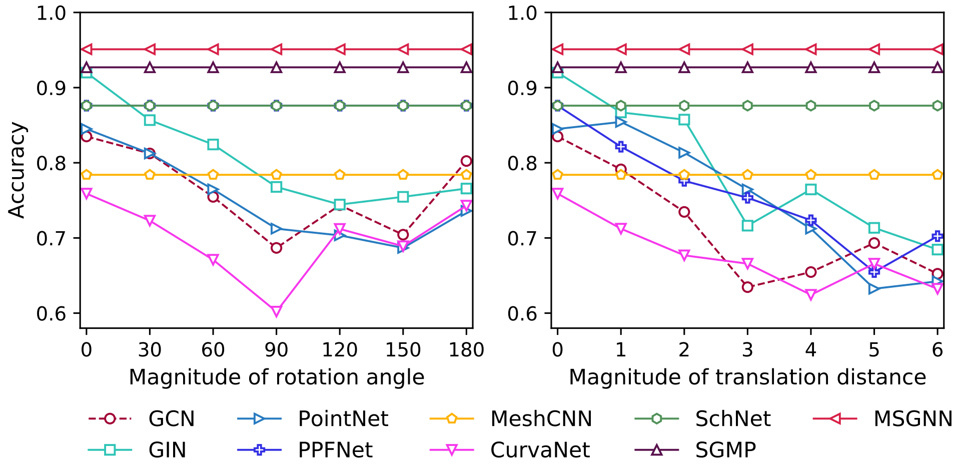

Figure 5: Robustness test on rotation and translation invariance by augmenting the data on the test set. The x-axis corresponds to the magnitude of the rotation angle (left) and translation distance (right). The y-axis shows the accuracy score on the test set.

Number of faces	1,000	2,000	4,000	8,000	16,000	32,000
HCP	0.928	0.939	0.944	0.947	0.950	0.951
Flight-GL	0.773	0.780	0.782	0.784	0.785	0.785

Table 3: Sensitivity analysis of model performance against the number of discretized mesh units.

predicting three targets on our method and competing methods. Particularly, we set the RMSE performance of all models at $\omega = 1$ as a benchmark, and then calculate the increasing rate of RMSE when ω increases. The results are shown in Figure 4.

According to the figure, compared to all baseline models, our model consistently shows a significantly slower increasing trend of the RMSE error as the degree of manifold irregularity increases. Such model behavior demonstrates that our model can effectively handle the irregularity in geometric manifolds. More interestingly, our model also shows a convergence trend in predicting the spatial radius r and the interaction range r_c, which arguably further demonstrates the robustness of our model to extremely irregular manifold environments.

5.3.2 Sensitivity analysis. We investigate the impact of the number of triangle mesh tessellations on our method to test the sensitivity of our model. We vary the number of mesh faces from high-resolution 32, 000 to low-resolution 1, 000 on two real-world datasets as shown in Table 3.

(1) According to Table 3, with the increase of triangular mesh subdivision number, the accuracy scores on all datasets show a upward and converging trend. The convergent performance trends demonstrate the effectiveness of using mesh tessellations to approximate spatial manifold surfaces. Specifically, as the resolution of the mesh tessellations increases, the approximate discrete surface is approaching the underlying continuous manifold space.

(2) It is also worth noting that as the number of mesh tessellations decreases, the performance of our proposed MSGNN on all methods gradually approaches and converges to the spatial network method on Euclidean space. The reason is that as the resolution of the mesh subdivision decreases, the approximate spatial path between nodes eventually converges to a Euclidean approximation.

5.3.3 Rotation and translation invariant test. Similar to previous work [19, 72], we also measure the robustness of rotation and translation by uniformly adding translation and rotation transformations to the input Cartesian coordinates. Due to space limitations, here we only report the accuracy results of the classification task on the brain dataset HCP, and the results are similar on all datasets. According to Figure 5, we can notice that the performance of our proposed model remains unchanged under all translation and rotation transformations. SchNet, SGMP, and MeshCNN can also achieve invariance under transformations because they also use only rotation- and translation-invariant geometric features in their models. PPFNet can be invariant under rotation transformation, but not translation transformation because it preserves the origin in the model. This experiment validates the importance of applying a rotation- and translation-invariant model, as we can observe that the performance of models without theoretical guarantees drops significantly when adding rotation and translation transformations. While our MSGNN model's performance is consistent with the theoretical guarantees that the learned representations are rigorously invariant under rotation and translation transformations.

6 CONCLUSION

This paper focuses on the crucial problem of learning representations from spatial networks that embedded in non-Euclidean manifolds, which is an underexplored area and can not be well handled by existing works. The proposed framework **M**anifold **S**pace **G**raph **N**eural **N**etwork (**MSGNN**) effectively addresses the unique challenges of representing irregular spatial networks by first converting the manifold space into a discrete mesh tessellation, and then converting the geometric information of the curves between nodes into messages on edges. Theoretical guarantees are given to prove that our learned representations are invariant to important symmetries such as rotation and translation, and simultaneously maintain strong distinguish power in geometric structures. Extensive experimental results on both synthetic and real-world datasets demonstrate the strength of our theoretical findings.

REFERENCES

[1] Réka Albert and Albert-László Barabási. 2002. Statistical mechanics of complex networks. *Reviews of modern physics* 74, 1 (2002), 47.

[2] Jayanth R Banavar, Amos Maritan, and Andrea Rinaldo. 1999. Size and form in efficient transportation networks. *Nature* 399, 6732 (1999), 130–132.

[3] Albert-László Barabási and Eric Bonabeau. 2003. Scale-free networks. *Scientific american* 288, 5 (2003), 60–69.

[4] Alain Barrat, Marc Barthelemy, and Alessandro Vespignani. 2008. *Dynamical processes on complex networks.* Cambridge university press.

[5] Marc Barthélemy. 2011. Spatial networks. *Physics Reports* 499, 1-3 (2011), 1–101.

[6] Austin R Benson, David F Gleich, and Jure Leskovec. 2016. Higher-order organization of complex networks. *Science* 353, 6295 (2016), 163–166.

[7] Wenming Cao, Zhiyue Yan, Zhiquan He, and Zhihai He. 2020. A comprehensive survey on geometric deep learning. *IEEE Access* 8 (2020), 35929–35949.

[8] Richard J Chorley and Petercoed Haggett. 1967. *Models in geography.* Technical Report.

[9] Gerardo Chowell, James M Hyman, Stephen Eubank, and Carlos Castillo-Chavez. 2003. Scaling laws for the movement of people between locations in a large city. *Physical Review E* 68, 6 (2003), 066102.

[10] Reuven Cohen and Shlomo Havlin. 2010. *Complex networks: structure, robustness and function.* Cambridge university press.

[11] David A Crowe, Shikha J Goodwin, Rachael K Blackman, Sofia Sakellaridi, Scott R Sponheim, Angus W MacDonald III, and Matthew V Chafee. 2013. Prefrontal neurons transmit signals to parietal neurons that reflect executive control of cognition. *Nature Neuroscience* 16, 10 (2013), 1484–1491.

[12] Tingting Dan, Zhuobin Huang, Hongmin Cai, Paul J. Laurienti, and Guorong Wu. 2022. Learning brain dynamics of evolving manifold functional MRI data using geometric-attention neural network. *IEEE Transactions on Medical Imaging* 41, 10 (2022), 2752–2763. https://doi.org/10.1109/tmi.2022.3169640

[13] Andrea De Montis, Marc Barthélemy, Alessandro Chessa, and Alessandro Vespignani. 2007. The structure of interurban traffic: a weighted network analysis. *Environment and Planning B: Planning and Design* 34, 5 (2007), 905–924.

[14] Haowen Deng, Tolga Birdal, and Slobodan Ilic. 2018. Ppfnet: Global context aware local features for robust 3d point matching. In *Proceedings of the IEEE conference on computer vision and pattern recognition.* 195–205.

[15] Victor M Eguiluz, Dante R Chialvo, Guillermo A Cecchi, Marwan Baliki, and A Vania Apkarian. 2005. Scale-free brain functional networks. *Physical review letters* 94, 1 (2005), 018102.

[16] Paul Erdős and Alfréd Rényi. 1960. On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci* 5, 1 (1960), 17–60.

[17] Reza Zanjirani Farahani, Elnaz Miandoabchi, Wai Yuen Szeto, and Hannaneh Rashidi. 2013. A review of urban transportation network design problems. *European journal of operational research* 229, 2 (2013), 281–302.

[18] Brendan J Frey and Delbert Dueck. 2007. Clustering by passing messages between data points. *science* 315, 5814 (2007), 972–976.

[19] Fabian B Fuchs, Daniel E Worrall, Volker Fischer, and Max Welling. 2020. SE (3)-transformers: 3D roto-translation equivariant attention networks. *arXiv preprint arXiv:2006.10503* (2020).

[20] Hongyang Gao, Zhengyang Wang, and Shuiwang Ji. 2018. Large-Scale Learnable Graph Convolutional Networks. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery &amp Data Mining.* ACM. https://doi.org/10.1145/3219819.3219947

[21] Linda Geerligs, Remco J Renken, Emi Saliasi, Natasha M Maurits, and Monicque M Lorist. 2015. A brain-wide study of age-related changes in functional connectivity. *Cerebral cortex* 25, 7 (2015), 1987–1999.

[22] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. 2017. Neural message passing for quantum chemistry. In *International Conference on Machine Learning.* PMLR, 1263–1272.

[23] Matthew F Glasser, Timothy S Coalson, Emma C Robinson, Carl D Hacker, John Harwell, Essa Yacoub, Kamil Ugurbil, Jesper Andersson, Christian F Beckmann, Mark Jenkinson, et al. 2016. A multi-modal parcellation of human cerebral cortex. *Nature* 536, 7615 (2016), 171–178.

[24] Marco Gori, Gabriele Monfardini, and Franco Scarselli. 2005. A new model for learning in graph domains. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, Vol. 2. IEEE, 729–734.

[25] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining.* 855–864.

[26] Peter Haggett and Richard J Chorley. 1969. *Network analysis in geography.* Vol. 1. Hodder Education.

[27] Matti Hämäläinen, Riitta Hari, Risto J Ilmoniemi, Jukka Knuutila, and Olli V Lounasmaa. 1993. Magnetoencephalography—theory, instrumentation, and applications to noninvasive studies of the working human brain. *Reviews of modern Physics* 65, 2 (1993), 413.

[28] William L Hamilton, Rex Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. *arXiv preprint arXiv:1706.02216* (2017).

[29] Rana Hanocka, Amir Hertz, Noa Fish, Raja Giryes, Shachar Fleishman, and Daniel Cohen-Or. 2019. Meshcnn: a network with an edge. *ACM Transactions on Graphics (TOG)* 38, 4 (2019), 1–12.

[30] Wenchong He, Zhe Jiang, Chengming Zhang, and Arpan Man Sainju. 2020. CurvaNet: Geometric deep learning based on directional curvature for 3D shape analysis. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining.* 2214–2224.

[31] Valerie D Hollard and Juan D Delius. 1982. Rotational invariance in visual pattern recognition by pigeons and humans. *Science* 218, 4574 (1982), 804–806.

[32] Theo Bendit (https://math.stackexchange.com/users/248286/theo bendit). [n. d.]. How to prove the triangle inequality for hermitian angle between vectors in a complex vector space? Mathematics Stack Exchange. arXiv:https://math.stackexchange.com/q/4044242 https://math.stackexchange.com/q/4044242 URL:https://math.stackexchange.com/q/4044242 (version: 2021-03-01).

[33] Zhuobin Huang, Hongmin Cai, Tingting Dan, Yi Lin, Paul Laurienti, and Guorong Wu. 2021. Detecting brain state changes by geometric deep learning of functional dynamics on Riemannian manifold. *Medical Image Computing and Computer Assisted Intervention – MICCAI 2021* (2021), 543–552. https://doi.org/10.1007/978-3-030-87234-2_51

[34] Mark Jenkinson, Christian F Beckmann, Timothy EJ Behrens, Mark W Woolrich, and Stephen M Smith. 2012. Fsl. *Neuroimage* 62, 2 (2012), 782–790.

[35] Junteng Jia and Austin R Benson. 2019. Random spatial network models for core-periphery structure. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining.*

[36] Marc Khoury and Jonathan Richard Shewchuk. 2019. Approximation Bounds for Interpolation and Normals on Triangulated Surfaces and Manifolds. *arXiv preprint arXiv:1911.03424* (2019).

[37] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).

[38] Johannes Klicpera, Janek Groß, and Stephan Günnemann. 2020. Directional message passing for molecular graphs. *arXiv preprint arXiv:2003.03123* (2020).

[39] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems* 25 (2012), 1097–1105.

[40] Maciej Kurant and Patrick Thiran. 2006. Extraction and analysis of traffic and topologies of transportation networks. *Physical Review E* 74, 3 (2006), 036114.

[41] Vito Latora and Massimo Marchiori. 2005. Vulnerability and protection of infrastructure networks. *Physical Review E* 71, 1 (2005), 015103.

[42] Nora Leonardi, Jonas Richiardi, Markus Gschwind, Samanta Simioni, Jean-Marie Annoni, Myriam Schluep, Patrik Vuilleumier, and Dimitri Van De Ville. 2013. Principal components of functional connectivity: a new approach to study dynamic brain connectivity during rest. *NeuroImage* 83 (2013), 937–950.

[43] Ron Levie, Federico Monti, Xavier Bresson, and Michael M Bronstein. 2018. Cayleynets: Graph convolutional neural networks with complex rational spectral filters. *IEEE Transactions on Signal Processing* 67, 1 (2018), 97–109.

[44] Ruoyu Li, Sheng Wang, Feiyun Zhu, and Junzhou Huang. 2018. Adaptive Graph Convolutional Neural Networks. *Proceedings of the AAAI Conference on Artificial Intelligence* 32, 1 (Apr. 2018). https://doi.org/10.1609/aaai.v32i1.11691

[45] Or Litany, Tal Remez, Emanuele Rodola, Alex Bronstein, and Michael Bronstein. 2017. Deep functional maps: Structured prediction for dense shape correspondence. In *Proceedings of the IEEE international conference on computer vision.* 5659–5667.

[46] Linyuan Lü and Tao Zhou. 2011. Link prediction in complex networks: A survey. *Physica A: statistical mechanics and its applications* 390, 6 (2011), 1150–1170.

[47] Haggai Maron, Meirav Galun, Noam Aigerman, Miri Trope, Nadav Dym, Ersin Yumer, Vladimir G Kim, and Yaron Lipman. 2017. Convolutional neural networks on surfaces via seamless toric covers. *ACM Trans. Graph.* 36, 4 (2017), 71–1.

[48] Jonathan Masci, Davide Boscaini, Michael Bronstein, and Pierre Vandergheynst. 2015. Geodesic convolutional neural networks on riemannian manifolds. In *Proceedings of the IEEE international conference on computer vision workshops.* 37–45.

[49] Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodolà, Jan Svoboda, and Michael M. Bronstein. 2016. Geometric deep learning on graphs and manifolds using mixture model CNNs. https://doi.org/10.48550/ARXIV.1611.08402

[50] Christopher Morris, Martin Ritzert, Matthias Fey, William L Hamilton, Jan Eric Lenssen, Gaurav Rattan, and Martin Grohe. 2019. Weisfeiler and leman go neural: Higher-order graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 4602–4609.

[51] Falko Müller. 2023. Link and edge weight prediction in air transport networks—An RNN approach. *Physica A: Statistical Mechanics and its Applications* (2023), 128490.

[52] Mathias Niepert, Mohamed Ahmed, and Konstantin Kutzkov. 2016. Learning convolutional neural networks for graphs. In *International conference on machine learning.* PMLR, 2014–2023.

[53] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *arXiv*

preprint arXiv:1912.01703 (2019).

[54] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. 2017. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition.* 652–660.

[55] Charles R. Qi, Hao Su, Matthias Niessner, Angela Dai, Mengyuan Yan, and Leonidas J. Guibas. 2016. Volumetric and Multi-View CNNs for Object Classification on 3D Data. https://doi.org/10.48550/ARXIV.1604.03265

[56] T. D. Rukmanda, K. A. Sugeng, and H. Murfi. 2018. Modification of architecture learning convolutional neural network for graph. *AIP Conference Proceedings* (2018). https://doi.org/10.1063/1.5064197

[57] Stephen Salsbury. 2019. The emergence of an early large-scale technical system: The American railroad network. In *The development of large technical systems.* Routledge, 37–68.

[58] Kristof T Schütt, Pieter-Jan Kindermans, Huziel E Sauceda, Stefan Chmiela, Alexandre Tkatchenko, and Klaus-Robert Müller. 2017. Schnet: A continuous-filter convolutional neural network for modeling quantum interactions. *arXiv preprint arXiv:1706.08566* (2017).

[59] Olaf Sporns, Giulio Tononi, and Gerald M Edelman. 2000. Theoretical neuroanatomy: relating anatomical and functional connectivity in graphs and cortical connection matrices. *Cerebral cortex* 10, 2 (2000), 127–141.

[60] Kartik K Sreenivasan, Clayton E Curtis, and Mark D'Esposito. 2014. Revisiting the role of persistent neural activity during working memory. *Trends in cognitive sciences* 18, 2 (2014), 82–89.

[61] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. 2015. Multi-view Convolutional Neural Networks for 3D Shape Recognition. https://doi.org/10.48550/ARXIV.1505.00880

[62] Kaustubh Supekar, Vinod Menon, Daniel Rubin, Mark Musen, and Michael D Greicius. 2008. Network analysis of intrinsic functional brain connectivity in Alzheimer's disease. *PLoS Comput Biol* 4, 6 (2008), e1000100.

[63] Shanshan Tang, Bo Li, and Haijun Yu. 2019. ChebNet: Efficient and Stable Constructions of Deep Neural Networks with Rectified Power Units using Chebyshev Approximations. https://doi.org/10.48550/ARXIV.1911.05467

[64] Waldo R Tobler. 1970. A computer movie simulating urban growth in the Detroit region. *Economic geography* 46, sup1 (1970), 234–240.

[65] David C Van Essen, Stephen M Smith, Deanna M Barch, Timothy EJ Behrens, Essa Yacoub, Kamil Ugurbil, Wu-Minn HCP Consortium, et al. 2013. The WU-Minn human connectome project: an overview. *Neuroimage* 80 (2013), 62–79.

[66] Jichuan Wang and Xiaoqian Wang. 2019. *Structural equation modeling: Applications using Mplus.* John Wiley & Sons.

[67] Rui Wang, Xiao-Jun Wu, and Josef Kittler. 2022. SymNet: A Simple Symmetric Positive Definite Manifold Deep Learning Method for Image Set Classification. *IEEE Transactions on Neural Networks and Learning Systems* 33, 5 (2022), 2208–2222. https://doi.org/10.1109/TNNLS.2020.3044176

[68] Duncan J Watts and Steven H Strogatz. 1998. Collective dynamics of 'small-world'networks. *nature* 393, 6684 (1998), 440–442.

[69] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 2014. 3D ShapeNets: A Deep Representation for Volumetric Shapes. https://doi.org/10.48550/ARXIV.1406.5670

[70] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2018. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826* (2018).

[71] Li Yi, Hao Su, Xingwen Guo, and Leonidas Guibas. 2016. SyncSpecCNN: Synchronized Spectral CNN for 3D Shape Segmentation. https://doi.org/10.48550/ARXIV.1612.00606

[72] Zheng Zhang and Liang Zhao. 2021. Representation learning on spatial networks. *Advances in Neural Information Processing Systems* 34 (2021), 2303–2318.

[73] Yu Zheng, Chen Gao, Liang Chen, Depeng Jin, and Yong Li. 2021. DGCN: Diversified Recommendation with Graph Convolutional Networks. In *Proceedings of the Web Conference 2021.* ACM. https://doi.org/10.1145/3442381.3449835

A MATHEMATICAL PROOFS

A.1 Proof of Theorem 1

Proof. Intuitively, distance, angle, torsion, and orientation angle are invariant to translation and rotation transformations, since only relative coordinates are used in the formula. Formally, for translation transformations $\mathcal{T} \in SE(3)$ and rotation transformations $\mathcal{R} \in SE(3)$, the following identity equations hold:

$$\begin{aligned} \mathcal{T}(\mathbf{x}-\mathbf{y}) &= \mathbf{x}-\mathbf{y}, \\ \langle \mathcal{R}(\mathbf{x}), \mathcal{R}(\mathbf{y}) \rangle &= \langle \mathbf{x}, \mathbf{y} \rangle \\ \mathcal{R}(\mathbf{x}) \times \mathcal{R}(\mathbf{y}) &= \mathcal{R}(\mathbf{x} \times \mathbf{y}) \end{aligned} \tag{5}$$

Thus we have

$$\begin{aligned} d &= ||\mathbf{l}||_2 = ||\mathbf{p}-\mathbf{p}'||_2 = ||\mathcal{T}(\mathbf{p})-\mathcal{T}(\mathbf{p}')||_2, \\ d &= ||\mathbf{l}||_2 = \langle \mathbf{l}, \mathbf{l} \rangle = \langle \mathcal{R}(\mathbf{l}), \mathcal{R}(\mathbf{l}) \rangle, \\ \theta &= \arccos(\langle \frac{\mathbf{l}}{d}, \frac{\mathbf{L_{ij}}}{d} \rangle) = \arccos(\langle \frac{\mathcal{R}(\mathbf{l})}{d}, \frac{\mathcal{R}(\mathbf{L_{ij}})}{d} \rangle), \\ \phi^{(k,k+1)} &= \langle \frac{\mathbf{c}^{(k)} \times \mathbf{c}^{(k+1)}}{||\mathbf{c}^{(k)} \times \mathbf{c}^{(k+1)}||_2}, \frac{\mathbf{L}_{ij}}{||\mathbf{L}_{ij}||_2} \rangle \cdot \bar{\phi}^{(k,k+1)} \\ \text{where} \quad \mathbf{c} &= \frac{\mathbf{L}_{ij} \times \mathbf{l}}{||\mathbf{L}_{ij} \times \mathbf{l}||_2}, \\ \mathbf{c}^{(k)} \times \mathbf{c}^{(k+1)} &= \frac{(\mathbf{L}_{ij} \times \mathbf{l}^{(k)}) \times (\mathbf{L}_{ij} \times \mathbf{l}^{(k+1)})}{||\mathbf{L}_{ij} \times \mathbf{l}^{(k)}||_2 ||\mathbf{L}_{ij} \times \mathbf{l}^{(k+1)}||_2}, \\ &= \frac{(\mathbf{L}_{ij} \cdot (\mathbf{l}^{(k)} \times \mathbf{l}^{(k+1)}))\mathbf{L}_{ij}}{||\mathbf{L}_{ij} \times \mathbf{l}^{(k)}||_2 ||\mathbf{L}_{ij} \times \mathbf{l}^{(k+1)}||_2}, \\ \text{thus } \langle & \frac{\mathbf{c}^{(k)} \times \mathbf{c}^{(k+1)}}{||\mathbf{c}^{(k)} \times \mathbf{c}^{(k+1)}||_2}, \frac{\mathbf{L}_{ij}}{||\mathbf{L}_{ij}||_2} \rangle \\ =& \langle \frac{(\mathbf{L}_{ij} \cdot (\mathbf{l}^{(k)} \times \mathbf{l}^{(k+1)}))\mathbf{L}_{ij}}{||\mathbf{c}^{(k)} \times \mathbf{c}^{(k+1)}||_2 ||\mathbf{L}_{ij} \times \mathbf{l}^{(k)}||_2 ||\mathbf{L}_{ij} \times \mathbf{l}^{(k+1)}||_2}, \frac{\mathbf{L}_{ij}}{||\mathbf{L}_{ij}||_2} \rangle \\ =& \frac{\mathbf{L}_{ij} \cdot (\mathbf{l}^{(k)} \times \mathbf{l}^{(k+1)})}{||\mathbf{c}^{(k)} \times \mathbf{c}^{(k+1)}||_2 ||\mathbf{L}_{ij} \times \mathbf{l}^{(k)}||_2 ||\mathbf{L}_{ij} \times \mathbf{l}^{(k+1)}||_2} \\ =& \frac{\mathcal{R}(\mathbf{L}_{ij}) \cdot (\mathcal{R}(\mathbf{l}^{(k)} \times \mathbf{l}^{(k+1)}))}{||\mathbf{c}^{(k)} \times \mathbf{c}^{(k+1)}||_2 ||\mathbf{L}_{ij} \times \mathbf{l}^{(k)}||_2 ||\mathbf{L}_{ij} \times \mathbf{l}^{(k+1)}||_2} \\ \varphi^{(k-1,k)} &= \arccos(\langle \mathbf{n}^{(k-1)}, \mathbf{n}^{(k)} \rangle) = \arccos(\mathcal{R}(\langle \mathbf{n}^{(k-1)}, \mathbf{n}^{(k)} \rangle)), \end{aligned}$$

All extracted geometric features are invariant under rotation and translation transformations. □

A.2 Proof of Theorem 2

Before proving Theorem 4, we first prove the Lemma 3.

Proof. Note that from Equation 2 we have followings:

$$\begin{aligned} ||\mathbf{L}_{i,j} \times \mathbf{l}^{(k)}||_2 &= ||(\mathbf{p}_i - \mathbf{p}_j) \times (\mathbf{p}^{(k)} - \mathbf{p}^{(k-1)})||_2 = d^k d_{i,j} \sin\theta^{(k)}, \\ \mathbf{c}^{(k)} \times \mathbf{c}^{(k-1)} &= \frac{1}{d_{i,j}^2 d^{(k)} d^{(k-1)} \sin\theta^{(k)} \sin\theta^{(k-1)}} \\ & \qquad (\mathbf{L}_{i,j} \times (\mathbf{p}^{(k)} - \mathbf{p}_j)) \times (\mathbf{L}_{i,j} \times (\mathbf{p}^{(k-1)} - \mathbf{p}_j)) \\ &= \frac{1}{d_{i,j}^2 d^{(k)} d^{(k-1)} \sin\theta^{(k)} \sin\theta^{(k-1)}} \end{aligned}$$

$$(\mathbf{L}_{i,j} \cdot ((\mathbf{p}^{(k)} - \mathbf{p}_j) \times (\mathbf{p}^{(k-1)} - \mathbf{p}_j)))\mathbf{L}_{i,j},$$

$$\langle \mathbf{c}^{(\mathbf{k})}, \mathbf{c}^{(\mathbf{k-1})} \rangle = \cos \bar{\phi}^{(k,k-1)} = \cos \phi^{(k,k-1)},$$

$$\frac{\langle \mathbf{c}^{(k)} \times \mathbf{c}^{(k-1)}, \mathbf{L}_{i,j} \rangle}{d_{i,j} \sin \bar{\phi}^{(k,k-1)}} = \frac{\mathbf{L}_{i,j} \cdot ((\mathbf{p}_j - \mathbf{p}^{(k)}) \times (\mathbf{p}_j - \mathbf{p}^{(k-1)}))}{d_{i,j} d^{(k)} d^{(k-1)} \sin \theta^{(k)} \sin \theta^{(k-1)} \sin \bar{\phi}^{(k,k-1)}}.$$

Suppose there exists two different positions of point $p^{(k)}$ and $p^{(k)\prime}$ that satisfy Equation 2, then it implies that

$$\frac{\mathbf{L}_{i,j} \cdot ((\mathbf{p}_j - \mathbf{p}^{(k)}) \times (\mathbf{p}_j - \mathbf{p}^{(k-1)}))}{d_{i,j} d^{(k)} d^{(k-1)} \sin \theta^{(k)} \sin \theta^{(k-1)} \sin \bar{\phi}^{(k,k-1)}}$$

$$= \frac{\mathbf{L}_{i,j} \cdot ((\mathbf{p}_j - \mathbf{p}^{(k)\prime}) \times (\mathbf{p}_j - \mathbf{p}^{(k-1)}))}{d_{i,j} d^{(k)} d^{(k-1)} \sin \theta^{(k)} \sin \theta^{(k-1)} \sin \bar{\phi}^{(k,k-1)}}$$

$$0 = (\mathbf{p}^{(k)} - \mathbf{p}^{(k)\prime}) \cdot ((\mathbf{p}_j - \mathbf{p}^{(k)}) \times (\mathbf{p}_j - \mathbf{p}^{(k-1)})).$$

Here $((\mathbf{p}_j - \mathbf{p}^{(k)}) \times (\mathbf{p}_j - \mathbf{p}^{(k-1)}))$ is nonzero as long as $\mathbf{p}_j$, $\mathbf{p}^{(k)}$, $\mathbf{p}^{(k-1)}$ is non-colinear. Therefore, the above equation causes a contradiction which proves that $p^{(k)}$ can not have two different positions given the representation in Equation 2. □

Then we provide the proof for Theorem 2.

Proof. As stated by Lemma 3, the Cartesian coordinates of a point $p^{(k)}$ can be determined from the two endpoints and their connected neighbor points $p^{(k-1)}$. Due to the connectivity of the spatial graph, we can repeatedly compute the coordinates of a connected point to the set of points with known coordinates. Thus, starting from an arbitrary point, the Cartesian coordinates of the entire spatial network are determined. □

A.3 Proof of Theorem 4

Before proving Theorem 4, we first prove that the angle satisfies the triangle inequality.

Lemma 5. *For any three vectors $\vec{x}, \vec{y}, \vec{z}$, we have $\angle(\vec{x}, \vec{y}) \leq \angle(\vec{x}, \vec{z}) + \angle(\vec{z}, \vec{y})$.*

Proof. This proof is borrowed from [32], and we restate it below to make it self-contained.

Without loss of generality, we assume that $\vec{x}, \vec{y}$ and $\vec{z}$ are unit vectors. Let

$$G = \begin{bmatrix} 1 & \langle \vec{x}, \vec{y} \rangle & \langle \vec{x}, \vec{z} \rangle \\ \langle \vec{y}, \vec{x} \rangle & 1 & \langle \vec{y}, \vec{z} \rangle \\ \langle \vec{z}, \vec{x} \rangle & \langle \vec{z}, \vec{y} \rangle & 1 \end{bmatrix}$$

Note that G is a Gramian matrix, and hence positive-semidefinite. Thus $\det G \geq 0$, we have

$$1 + 2\langle \vec{x}, \vec{y} \rangle \langle \vec{y}, \vec{z} \rangle \langle \vec{z}, \vec{x} \rangle - |\langle \vec{x}, \vec{y} \rangle|^2 - |\langle \vec{y}, \vec{z} \rangle|^2 - |\langle \vec{z}, \vec{x} \rangle|^2 \geq 0.$$

This is equivalent to

$$(1 - |\langle \vec{x}, \vec{y} \rangle|^2)(1 - |\langle \vec{y}, \vec{z} \rangle|^2) \geq |\langle \vec{x}, \vec{y} \rangle \langle \vec{y}, \vec{z} \rangle - \langle \vec{z}, \vec{x} \rangle|^2.$$

by expanding both sides and canceling the common term$|\langle \vec{x}, \vec{y} \rangle|^2 |\langle \vec{y}, \vec{z} \rangle|^2$. Taking the square root of both sides, we get

$$|\langle \vec{x}, \vec{y} \rangle \langle \vec{y}, \vec{z} \rangle| - |\langle \vec{z}, \vec{x} \rangle| \leq |\langle \vec{x}, \vec{y} \rangle \langle \vec{y}, \vec{z} \rangle - \langle \vec{z}, \vec{x} \rangle|$$
$$\leq \sqrt{1 - |\langle \vec{x}, \vec{y} \rangle|^2} \sqrt{1 - |\langle \vec{y}, \vec{z} \rangle|^2},$$

thus

$$|\langle \vec{z}, \vec{x} \rangle| \geq |\langle \vec{x}, \vec{y} \rangle \langle \vec{y}, \vec{z} \rangle| - \langle \vec{z}, \vec{x} \rangle| \geq \sqrt{1 - |\langle \vec{x}, \vec{y} \rangle|^2} \sqrt{1 - |\langle \vec{y}, \vec{z} \rangle|^2}$$

Note that $\cos \angle(\vec{x}, \vec{y}) = |\langle \vec{x}, \vec{y} \rangle|$ holds for any unit vectors $\vec{x}$, $\vec{y}$, so this inequality becomes

$$\cos \angle(\vec{x}, \vec{z}) \geq \cos \angle(\vec{x}, \vec{y}) \cos \angle(\vec{y}, \vec{z}) - \sqrt{1 - \cos\langle \vec{x}, \vec{y} \rangle^2} \sqrt{1 - \cos\langle \vec{y}, \vec{z} \rangle^2}$$
$$= \cos \angle(\vec{x}, \vec{y}) \cos \angle(\vec{y}, \vec{z}) - \sin \angle(\vec{x}, \vec{y}) \sin \angle(\vec{y}, \vec{z})$$
$$= \cos(\angle(\vec{x}, \vec{y}) + \angle(\vec{y}, \vec{z}))$$

here $|\sin \angle(\vec{x}, \vec{y})| = \sin \angle(\vec{x}, \vec{y})$ and $|\sin \angle(\vec{y}, \vec{z})| = \sin \angle(\vec{y}, \vec{z})$ because $\angle(\vec{x}, \vec{y}) \in [0, \pi], \angle(\vec{y}, \vec{z}) \in [0, \pi]$. Because $\cos(\bullet)$ is strictly monotone decreasing on $[0, \pi]$, we have

$$\angle(\vec{x}, \vec{z}) \leq \angle(\vec{x}, \vec{y}) + \angle(\vec{y}, \vec{z}).$$

This proof is completed. □

Now we prove Theorem 4 as follows:

Proof. By applying Lemma 5, we have

$$\angle(N^{f_{k_1}^{(i_1,j_1)}}_{v_{k_1}^{(i_1,j_1)}}, N^{f_{k_2}^{(i_2,j_2)}}_{v_{k_2}^{(i_2,j_2)}}) \leq \angle(N^{f_{k_1}^{(i_1,j_1)}}_{v_{k_1}^{(i_1,j_1)}}, N^{M}_{v_{k_1}^{(i_1,j_1)}}) + \angle(N^{M}_{v_{k_1}^{(i_1,j_1)}}), (N^{M}_{v_{k_2}^{(i_2,j_2)}})$$
$$+ \angle(N^{M}_{v_{k_2}^{(i_2,j_2)}}, N^{f_{k_2}^{(i_2,j_2)}}_{v_{k_2}^{(i_2,j_2)}}).$$

By Lemma 2 in [36], we have

$$\angle(N^{f_{k_1}^{(i_1,j_1)}}_{v_{k_1}^{(i_1,j_1)}}, N^{M}_{v_{k_1}^{(i_1,j_1)}}) \leq \arcsin(\frac{R_{k_1}^{(i_1,j_1)}}{ebs(v_{k_1}^{(i_1,j_1)})} \max(\cot(\frac{\phi_{k_1}^{(i_1,j_1)}}{2}), 1))$$

$$\angle(N^{M}_{v_{k_2}^{(i_2,j_2)}}, N^{f_{k_2}^{(i_2,j_2)}}_{v_{k_2}^{(i_2,j_2)}}) \leq \arcsin(\frac{R_{k_2}^{(i_2,j_2)}}{ebs(v_{k_2}^{(i_2,j_2)})} \max(\cot(\frac{\phi_{k_2}^{(i_2,j_2)}}{2}), 1))$$

By Lemma 4 in [36], we have

$$\angle(N^{M}_{v_{k_1}^{(i_1,j_1)}}), (N^{M}_{v_{k_2}^{(i_2,j_2)}}) \leq \arccos(1 - \frac{\delta^2}{2\sqrt{1 - \delta^2}})$$

We sum up all these inequalities, and Theorem 4 is proven. □