Received: 5 April 2024 | Accepted: 28 May 2024

DOI: 10.1111/mice.13286

COMPUTER-AIDED CIVIL AND INFRASTRUCTURE ENGINEERING WILEY

RESEARCH ARTICLE

Modeling of spatially embedded networks via regional spatial graph convolutional networks

Xudong Fan | **Jürgen Hackl**

Department of Civil and Environmental Engineering, Princeton University, New Jersey, USA

Correspondence
Jürgen Hackl, Department of Civil and Environmental Engineering, Princeton University, Princeton, NJ 08544, USA.
Email: hackl@princeton.edu

Present address
Jürgen Hackl, E322 Engineering Quadrangle, Princeton, NJ 08544, USA

Funding information
Energy Research Fund; Andlinger Center for Energy and the Environment; School of Engineering and Applied Science; Princeton University

Abstract

Efficient representation of complex infrastructure systems is crucial for system-level management tasks, such as edge prediction, component classification, and decision-making. However, the complex interactions between the infrastructure systems and their spatial environments increased the complexity of network representation learning. This study introduces a novel geometric-based multimodal deep learning model for spatially embedded network representation learning, namely the *regional spatial graph convolutional network* (RSGCN). The developed RSGCN model simultaneously learns from the node's multimodal spatial features. To evaluate the network representation performance, the introduced RSGCN model is used to embed different infrastructure networks into latent spaces and then reconstruct the networks. A synthetic network dataset, a California Highway Network, and a New Jersey Power Network were used as testbeds. The performance of the developed model is compared with two other state-of-the-art geometric deep learning models, GraphSAGE and Spatial Graph Convolutional Network. The results demonstrate the importance of considering regional information and the effectiveness of using novel graph convolutional neural networks for a more accurate representation of complex infrastructure systems.

1 | INTRODUCTION

Real-world infrastructure systems can be represented and interpreted as complex networks (Nocera & Gardoni, 2022). For instance, road networks can be represented by using intersections as nodes and road segments as edges (D. Xu et al., 2022). Similarly, power networks can be modeled by representing buses as nodes and transmission lines as edges (Ma et al., 2021). Thereby, a critical feature of these infrastructure systems is that they are embedded in spatial environments, which in turn shape and constrain the topologies of the systems (D. Zhang et al., 2022). However, due to the complex interactions between the infrastructure systems and their spatial environments, the intricate patterns and complex relationships within the topologies of these complex systems are still unclear and difficult to capture.

A comprehensive network representation and an understanding of the intricate patterns within these complex systems are needed for many real-world challenges (De Bacco et al., 2017; Peixoto, 2019; N. Zhang & Alipour, 2023; Dunton & Gardoni, 2023). For instance, previous studies have captured the intricate connection patterns from partially observed networks and used these patterns to rebuild

the network structures of the road network. (Molinero & Hernando, 2020; Dunton & Gardoni, 2023). The intricate patterns within the networks are also vital for link feature prediction, which have been used to classify unknown edges to their belonging layers in a multilayer infrastructure system (T. Cai et al., 2017; Z. Xu et al., 2020). Furthermore, an efficient network representation technique can embed complex networks into a latent space, which enhances the network's properties analysis and prediction (Fan et al., 2022; Mao et al., 2023).

Previous studies have developed a variety of methods for network representation and intricate pattern understanding. The proposed models can be broadly classified into two groups, the geometric algorithm-based methods and geometric deep learning-based methods. The former approach used fitting the network connection with some geometric-based equations, such as the stochastic block models (Holland et al., 1983), random networks (Pikovsky, 2018), small-world networks (Aksoy et al., 2019), and scale-free networks (Zheng et al., 2012). However, this approach is constrained by predefined criteria, such as functions of distance and node degrees (Dettmann & Georgiou, 2016). On the contrary, the latter approach relaxed the constraints of predefined equations using neural networks, such as graph convolutional neural networks (GCN) (Ding et al., 2024). However, most geometric deep learning models treat the node features homogeneously, often by simply concatenating all features into a single vector. In the real world, the dimensions of node features may vary depending on the physical types of the features. On the other hand, studies about multimodal GCN have been emerging, current studies are focusing on image processing, knowledge graphs, molecular graphs, and physical/chemistry networks (Ektefaie et al., 2023). There remains a lack of studies investigating the training of geometric deep-learning models to accommodate nodes with varying feature dimensions.

This work addresses the limitations of traditional geometric algorithms and integrates spatial environments into complex network's pattern discovery, a *regional spatial graph convolutional network* (RSGCN) is presented for spatially embedded networks. The introduced RSGCN model is integrated into a novel *partition-then-ensembling framework* for the network topology reconstruction task, which is a challenging yet fundamental obstacle in applying network science caused by the restricted access to data (Aksoy et al., 2019). The network reconstruction framework can benefit large-scale infrastructure system management in various ways, such as system risk estimation and system properties analysis (Vaccariello et al., 2020). In addition, the results of network reconstruction can be effectively used to evaluate the performance of the network representation models, considering the complexity of ensuring similar graph properties between generated networks and the original (Hackl & Adey, 2019).

In sum, this work advances the state of the art of network representation, graph learning, and intricate patterns discovery for spatially embedded networks as follows:

1. This work derives an RSGCN that integrates fully connected neural network layers and convolutional neural network layers into the graph convolutional process. Consequently, this novel RSGCN model is capable of processing node features with varying dimensions.
2. The node's regional information can be seamlessly processed with traditional vector features, enabling a more efficient capturing of intricate patterns of spatially embedded networks. The results of this study demonstrate that incorporating the node's regional information significantly enhances the modeling accuracy of the network connection compared to models lacking this information.
3. The developed RSGCN model is integrated into a partition-then-ensembling framework to address computing challenges in large networks. The large network is a common challenge in network science, considering the corresponding adjacency matrix is quadratically increased with the increase of a node number. A large network is first partitioned into batches of subgraphs for the training and predicting process. The predicted subgraphs are then reassembled back to a large network. This approach significantly improved computing efficiency and enabled RSGCN to be applied to networks of any size.
4. The developed RSGCN model is a generic network representation learning method for spatially embedded networks. The learned network representation can be applied to other real-world tasks, such as traffic flow prediction, edge prediction, and decision-making for networked infrastructure systems.

The remainder of this article is organized as follows. Section 2 presents the related works about network representation, intricate patterns discovery, and spatially embedded networks. Section 3 introduces the main object of this study, which is a spatially embedded network reconstruction task. Section 3 also introduces the developed RSGCN model and the developed partition-then-ensembling network reconstruction framework. After that, Section 4 provides details about testbeds and evaluation metrics used in this study, including a synthetic network dataset, a highway network dataset, and a power transmission dataset. The corresponding results of the network reconstruction are introduced in Section 5. Finally, discussions and concluding remarks on this study are given in

Sections 6 and 7. The developed codes are available on Zenodo.[1]

2 | BACKGROUND

Various techniques have been proposed to capture the representations and intricate patterns of complex networks. For the network representations, DeepWalk (Perozzi et al., 2014) and Node2Vec (Grover & Leskovec, 2016) are two common traditional graph embedding methods. The former approach has been used for a multilayer infrastructure network's community detection and edge prediction (J. Li et al., 2018). The latter approach has been used to discover urban function regions using GPS-based trajectory data (L. Cai et al., 2022). In addition to the network representation, notable infrastructure systems' intricate patterns have also been discovered using traditional geometric algorithms. For instance, power systems of the same voltage level have been found to have similar structure properties with small-world networks, which is essential for generating synthetic electric infrastructure networks (Aksoy et al., 2019). Single-parameter controlled hierarchical planar and spatial networks have also been proposed to mimic the connection behaviors of road networks (Molinero & Hernando, 2020). The complex relationships within the infrastructure systems may not be fully represented by a single geometric algorithm. A hybrid geometric algorithm has also been proposed, which combines relative neighborhood graphs, Gabriel graphs, and Erdős–Rényi random graph (Hackl & Adey, 2019). The real-world power systems have also been quantified and embedded into the tunable spanning tree procedure, which is in turn used for generating synthetic power systems (Soltan & Zussman, 2016). Lastly, physical information has also been considered in developing the connection patterns of infrastructure systems. For instance, the slope and elevation information has been considered for modeling the sewer system's topology (Dunton & Gardoni, 2023). The cost-optimal approach has also been used for the generation of synthetic sewer and wastewater systems (Moeini & Afshar, 2018; Chahinian et al., 2019).

Recently, due to the successful progress of deep learning and neural networks, machine learning based methods have been widely applied in different engineering domains (Rafiei & Adeli, 2016, 2018), including the network representation learning of complex networks (Lian & Xu, 2022; Che et al., 2022). Unlike conventional geometric-based methods, geometric deep-learning models offer a more flexible and end-to-end learning process, which facilitates network representation and intricate pattern discovery (Ding et al., 2024). The GCN was first proposed for architecture in geometric deep-learning models. After that, a large number of GCN variants have been proposed for different tasks. Notable variants include the GraphSAGE (Hamilton et al., 2017) and spatial graph convolutional networks (SGCN) (Danel et al., 2020). The former architecture introduced advanced sampling strategies for the node's neighbors, resulting in a higher node classification accuracy in multiple datasets. The latter SGCN architecture first introduced the position features of nodes into the learning process of a molecular classification task. GCN and its variants have been receiving more and more attention in civil and infrastructure engineering. For example, a GCN has been integrated into a deep reinforcement learning process for the water system's restoration decision-making (Fan et al., 2022). More optimal decisions can be obtained due to a better learning process of using GCN. A graph attention architecture was used to capture the spatial correlations within traffic networks for traffic flow prediction (Z. Wang et al., 2023). In addition, the GCNs have been widely used in power systems for fault detection, power outage prediction, power flow simulation, and system control (Liao et al., 2022).

Both traditional geometric-based algorithms and geometric deep-learning models often treat complex networks as abstract networks and ignore the networks' embedded spatial environments. For example, the reconstruction of power networks is often based on their graph properties, such as the node degree distribution, edge length distribution, and network connectivity metrics (Aksoy et al., 2019). However, the infrastructure systems are known as spatially embedded (Dong et al., 2020). Advanced network learning models need to efficiently represent both the network's spatial environments and its graph properties.

In summary, combining the spatial environment of the infrastructure system with more advanced geometric learning methods is urgently needed for effectively capturing the network representation and intricate patterns of complex infrastructure systems.

3 | METHODOLOGY

3.1 | Preliminaries

In this study, the infrastructure systems are modeled as spatially embedded networks, considering the strong impacts of spatial environments on the network structures of these systems. Specifically, a spatially embedded network $\boldsymbol{G}$ can be modeled by its vertices $\boldsymbol{V}$, edges $\boldsymbol{E}$, and spatial environments $\boldsymbol{S}$. The detailed definitions and

[1] https://doi.org/10.5281/zenodo.11584148

symbols for the spatially embedded networks used in this study are represented as follows.

1. *A network* $\boldsymbol{G}$: A network $\boldsymbol{G}(\boldsymbol{V}, \boldsymbol{E}, \boldsymbol{S})$ which represents the infrastructure systems, where the $\boldsymbol{V}$ denotes the vertices, $\boldsymbol{E}$ denotes the edges, and $\boldsymbol{S}$ denotes the spatial environment. The corresponding adjacency matrix of graph $\boldsymbol{G}$ can be represented by $\boldsymbol{A}$.
2. *A set of vertices* $\boldsymbol{V}$: v represents a vertice of critical infrastructure networks, such as an intersection of road networks or a connection point in power networks. $v \in \boldsymbol{V}$.
3. *Node features* $p \in \mathcal{P}$, $x \in \mathcal{X}$, $r \in \mathcal{R}$: This study groups the node features into three classes. The *node position feature* refers to the coordinates of the nodes p, such as the latitude and longitude of each node. The *node point feature* refers to the vectorized features arranged in one dimension, denoted by x, examples of node point features include the social-economic factors that the node is located. And, the *node's regional feature* refers to the two-dimensional regional data centered on each node, denoted by r. Examples of this regional feature including the regional elevation change in the real world. In this study, the network's spatial environment $\boldsymbol{S}$ consists of $\mathcal{P}, \mathcal{X}, \mathcal{R}$.
4. *A set of edges* $\boldsymbol{E}$: the edge $e_{i,j}$ is the connection between a pair of vertices (v_i, v_j), such as a road segment or a power line. The connection probability between two vertices is dependent on $\boldsymbol{S}$, which can be described by $\mathbb{P}(e_{i,j}) = g(v_i, v_j|\boldsymbol{S})$.

The objective of this study is to reconstruct the topology of spatially embedded networks by using the developed RSGCN model as a surrogate connection function. Connection functions are used to calculate the existence probability of an edge between two vertices (Hackl & Adey, 2019). The objective can be mathematically described as Equation (1), that is, with a given set of nodes and regional information, the developed model aims to find a connection function $f(v_i, v_j|\theta)$ to reconstruct the adjacency matrix A, so that the difference between the reconstructed $\check{A}$ and original adjacency matrix A can be minimized.

$$\begin{aligned} &\underset{\theta}{\operatorname{argmin}}\, \ell\left(\check{A}, A\right) \\ &\check{A}_{i,j} = f\left(v_i, v_j|\boldsymbol{S}, \theta\right), \end{aligned} \tag{1}$$

where A denotes the graph's adjacency matrix and $\check{A}$ is the reconstructed adjacency matrix. $\ell(,)$ is the loss function. $f(v_i, v_j|S, \theta)$ is the connection function used to get reconstructed adjacency matrix $\check{A}$, and θ is the internal parameters of function f.

3.2 | Overall framework

Considering the large size and spatial constraints of infrastructure systems, it is impractical to assume that all nodes within the infrastructure systems are potentially connected. Predicting the existence of edges between all pairs of nodes also requires intensive computing resources. Therefore, a partition-then-ensembling framework is proposed in this study as shown in Figure 1. First, the network is partitioned into a set of subgraphs. Partial of the sampled subgraphs are used for model's training process, and the rest of the subgraphs are used for testing. The final network is built by ensembling the predictions of all subgraphs together. A detailed description of each step is given in the following sections.

3.2.1 | Subgraph sampling

The graph sampling process is used to convert a single large network into a set of subgraphs. This step is particularly useful when working with large networks, as training a batch of small subgraphs is much more efficient than training a single large network. This step is also applicable to real-world infrastructure systems, where edge lengths are often constrained by physical space. Consequently, the probability of connectivity between two nodes significantly distanced apart from each other is extremely low.

In this study, the subgraphs of the large network are sampled by using a fixed region, as shown by the dash boxes in Figure 1a. The end nodes of edges that are partially within the region are also included in the subgraph. The selection of this window size is used to generate a set of subgraphs with an appropriate number of nodes and edges, which is dependent on the study area and edge length distribution. In order to cover the whole network for the final network rebuild, the sampling process is conducted by using all the nodes as sampling centers. Therefore, the number of subgraphs equals the number of nodes.

After sampling subgraphs from the large network, each subgraph is converted into a complete graph (Figure 1b), as all pairs of nodes within the subgraphs are potentially connected. The edges of the complete graph are labeled by either 1 or 0 for future training purposes. Edges existing in the original network are labeled as 1, while the other edges are labeled as 0.

3.2.2 | Edge prediction

The geometric-based deep learning models are used to capture the intricate patterns within the network by

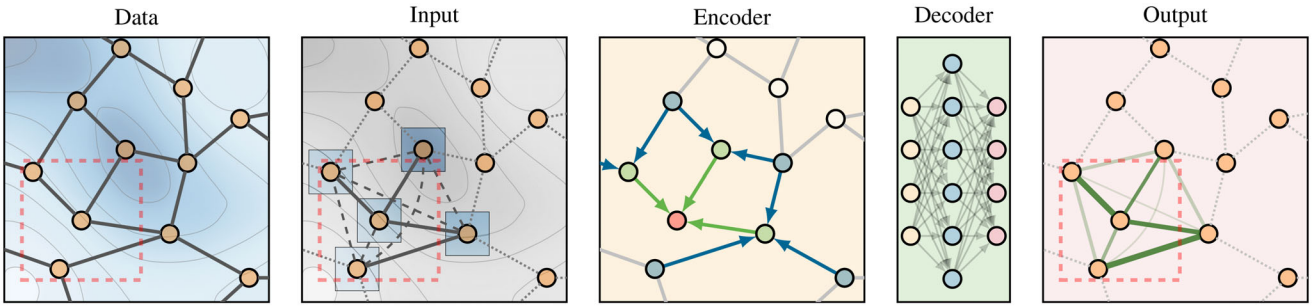


FIGURE 1 Proposed framework.

embedding the nodes into a latent space. As introduced in Section 3.1, three types of node features are considered, that is, $\mathcal{P}$, $\mathcal{X}$, and $\mathcal{R}$. These different types of spatial information can be embedded into node features, which are further used to predict the probability of edge existence between all pairs of nodes in the subgraph. In other words, geometric-based deep learning is used as a surrogate model of traditional geometric algorithm-based connection functions (Dettmann & Georgiou, 2016). Figure 1c shows a generic process of geometric deep learning models. The blue box represents generic geometric-based deep learning models, and the node's features are fed to fully connected neural networks for the edge existence prediction.

Unlike traditional graph learning models which only process homogeneous and vectorized node features, the developed RSGCN model processed node features with different dimensions simultaneously. Particularly, the developed RSGCN model processed the node's position features $\mathcal{P}$ by using a fully connected neural network, the node's regional feature $\mathcal{R}$ by using a convolutional neural network, and the node's point feature $\mathcal{X}$ by using another fully connected neural network. The processed data are then concatenated for further edge existence prediction. Details about the developed RSGCN model are introduced in Section 3.3.

3.2.3 | Network ensemble

The nodes of subgraphs are the subset of the nodes of the original network. Therefore, the final network can be rebuilt by ensembling the edge existence probability of all edges in subgraphs. An edge may exist in multiple subgraphs due to the sampling strategy. In this study, the final network is rebuilt by using the averaged edge existence probability. An edge is classified as existence if the averaged existence probability is higher than a predefined probability threshold. Consequently, this developed ensembling process is highly efficient because this strategy automatically excludes edges between nodes that are extremely far apart.

ALGORITHM 1 Pseudocode for " strategy

Input: Predicted Graph **G**
Get connected components list $\mathbf{g} = [g_1, g_2, \dots g_n]$
for any (g_m, g_n) in **g do**
  Get edge list $\mathbf{e} = [e_1, e_2, \dots, e_n]$, where e_i connected (g_m, g_n)
  Get existence probability of **p** as $[p_1, p_2, \dots, p_n]$
  Connect edge based on $\mathbf{e}[argmax(\mathbf{p})]$
end for

It is important to note that the geometric deep learning based models predict the existence probability of all edges simultaneously, which may not guarantee the rebuilt network is a single connected component. A connected component in graph theory indicates a graph whose all pairs of nodes are connected via at least an available path. To address this, a *"probability relax"* strategy is proposed, which relaxes the existence probability threshold for certain node pairs if strong connectivity is required. The pseudocode of the *"probability relax"* is provided below. Specifically, when multiple connected components exist in the predicted graph, the edges between these connected components are established if the edges have the next highest existence probability, even if this probability is lower than the threshold. The edge list e in Algorithm 1 can be empty if none edges in the sampling set connected two connected components.

This approach guarantees the rebuilt network is a single connected component. However, many networks naturally have multiple connected components, and this step may not be necessary.

3.3 | RSGCN model

This section introduces the detailed architecture of the developed RSGCN model. A unique contribution of this

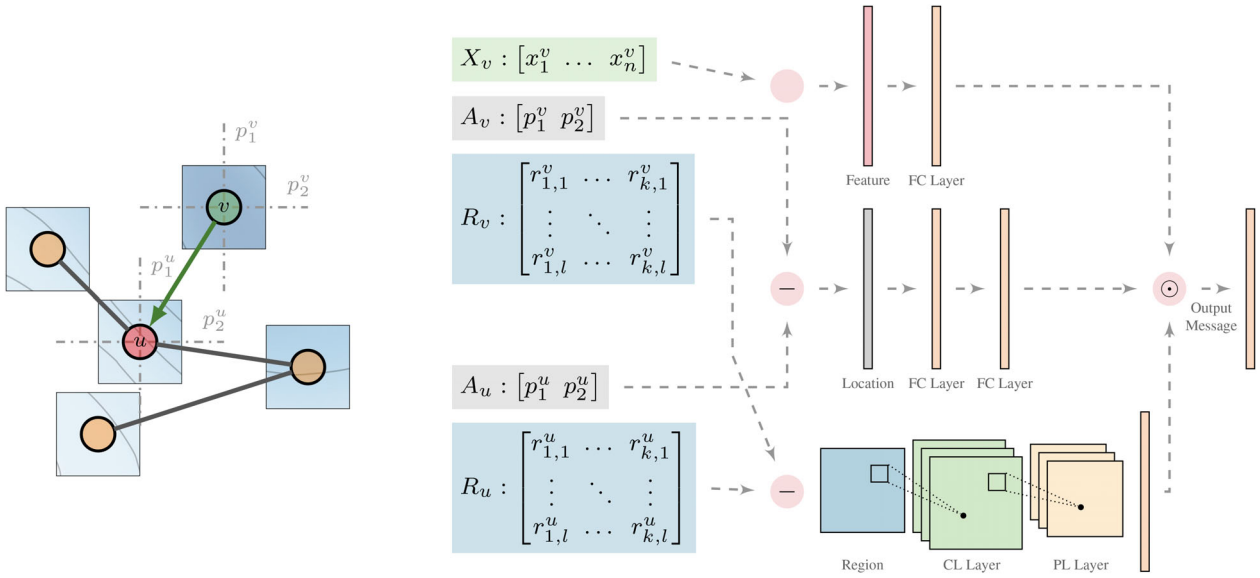


FIGURE 2 Illustrative architecture one single layer of RSGCN (The blocks with the same colors indicate the same dimensions. The weights and bias of each block are independent. FC, fully connected; CL, convolutional layer; PL, pooling layer; $[p_1, p_2]$, node's position feature; x, node point features; r, node regional features; $(-)$, minus operator; $\odot$, element-wise product.

RSGCN is its ability for multimodal fusion, whereas conventional GCN only processes homogeneous vectorized node features. A single layer of the RSGCN model is shown in Figure 2, which visualizes how different types of data are handled. A detailed introduction to the RSGCN model is given in the following sections. The corresponding codes can also be found on Zenodo.[2]

3.3.1 | Node position feature

Previous studies on spatial graph convolutional networks have demonstrated the superiority of adding the node's position feature to other data sources (Danel et al., 2020). This study uses the relative positions of neighboring nodes over absolute positions. To process the node's position feature, two fully connected neural networks are utilized for extracting the location feature. The extracted location feature is represented by Equation (2).

$$\widetilde{p} = \sigma[W_2^p(\sigma((p_i - p_j)W_1^p))], \tag{2}$$

where σ is the *ReLU* activation function, p_i, p_j is the position of nodes i and j, and W^p is the weights of layers for position feature processing.

3.3.2 | Node point feature

The node point feature represents the vectorized node feature $\mathcal{X}$, which is commonly a concatenated vector of various homogeneous one-dimensional features. This is also the most widely used node feature in previous graph convolutional learning. The dimension of this vector is case specified. In order to process the node point feature, a fully connected neural network is used, as shown in Equation (3).

$$\widetilde{x} = \sigma(x_i W_1^x), \tag{3}$$

where σ is the activation function. *ReLU* activation function is used in this study. x_i is the node point feature of node i, and $\widetilde{x}$ is the processed feature, W_1^x denotes the weights of a fully connected neural network for node point feature processing.

3.3.3 | Node regional feature

Lastly, the regional feature of each node is a two-dimensional matrix, as shown in Figures 1 and 2. The value of the matrix is the spatial environment of that region. A convolutional neural network is utilized for feature extraction. The convolutional process can be mathematically described by Equation (4) (J. Wu, 2017). The striding, padding, and flattening processes are omitted in Equation (4) for concise purposes. One fully connected layer is appended to the convolutional layer so that the output dimension of regional information can be the same as the other layers.

$$\widetilde{r} = \sigma\left(\sum_k^{m_1}\sum_l^{m_2} K_{[k,l]}(r_i - r_j)\right), \tag{4}$$

[2] https://doi.org/10.5281/zenodo.11584148

where $\widetilde{r}$ is the convolved value of the output, K is the kernel window, r_i and r_j are the input two-dimensional regional information of nodes i and j. m_1 is the height of the input data, and m_2 is its width. k, l are the coordinates of the elements in $\widetilde{r}$.

3.3.4 | Message passing

The processed information needs to be combined so that the message can be forwarded from one node to its neighbors. An element-wise multiplication is used to combine the information (Equation 5), and the conventional graph convolutional process is used for the message parsing (Equation 6). The graph-convolved information is used to replace the node's point feature x and forwarded to the next layer for computation.

$$\widetilde{m} = \widetilde{p} \odot \widetilde{x} \odot \widetilde{r}, \tag{5}$$

where $\widetilde{m}$ is the transformed message. $\odot$ represents the element-wise multiplication.

$$x_i^l = \sigma\left(x_i^{l-1} + \sum_{j \in N_i} \cdot \widetilde{m}_j \right), \tag{6}$$

where x_i^l is the convolved feature of node i at lth layer, σ is the *ReLU* activation function, and $\widetilde{m}_j$ is the transformed message from neighbor nodes (Equation 5).

Figure 2 shows a single layer of the developed RSGCN. The final developed connection probability prediction model includes two RSGCN layers as encoders and three fully connected neural networks as decoders, as shown in Figure 1c. Specifically, the RSGCN layers are used to embed the nodes' features within the graph data into latent space. After graph embedding, the node's features of ends of edges are concatenated and fed to fully connected neural networks for decoding. The outputs of the fully connected neural networks are the existence probabilities of all edges, which are trained by the prelabeled edge data in the training set. The activation function between all hidden layers is the *ReLU* activation function. The activation function of the last layer is the *Sigmoid* activation function. The prediction of edge existence probability can be defined in Equation (7).

$$\mathbb{P}(e_{i,j}) = \sigma\left(W_1 \cdot \left(x_i \oplus x_j\right)\right) \qquad \forall (i,j), \tag{7}$$

where σ is the *Sigmoid* activation function. $\oplus$ is a concatenating operator. $\mathbb{P}(e_{i,j})$ is the connection probability of edge $e_{i,j}$.

3.4 | Performance evaluation

Three matrices are used to evaluate the performance of the geometric deep-learning models in the network reconstruction task, that is, F1 score, Kullback–Leibler divergence (K-L divergence) of edge length distributions, and K-L divergence of node degree distributions. The node degree of a node is the number of edges connected to it. Specifically, the F1 score is used to evaluate the prediction accuracies by considering the network reconstruction task as a binary classification problem. The outputs of the prediction models are the existence probability of the edges between all node pairs. An edge is classified as "existence" if its probability is higher than a predefined threshold; otherwise, it is classified as "nonexistence." In this study, this threshold is set as 0.5 for all case studies. Discussions regarding the selection of this threshold can be found in Section 6.

Equation (8) shows the definition of the F1 score, where the true positive (TP) represents the edges that are originally existent and also predicted as existent. The true negative (TN) represents the edges that were originally nonexistent and also predicted as nonexistent. The false positive (FP) represents the edges that are originally nonexistent but predicted as existent. And, the false negative (FN) represents the edges that are originally existent but predicted as nonexistent.

$$\mathrm{F1}_{\mathrm{Score}} = \frac{\mathrm{TP}}{\mathrm{TP} + \frac{1}{2}(\mathrm{FP} + \mathrm{FN})} \tag{8}$$

The distribution similarities of edge lengths and node degrees between the rebuilt networks and the original networks are evaluated using the K-L divergence (Cover & Thomas, 2006). The K-L divergence measures the statistical distance between two probability distributions, which can be calculated by Equation (9). It should be noted that the K-L divergence is an asymmetric index. $P(z)$ is the distribution of graph properties of the original graphs, that is, the node degrees and edge lengths. And $Q(z)$ is that of predicted graphs. The lower K-L divergence distance indicates more similarities between the two distributions. 0 indicates that the two distributions have identical quantities of information.

$$D_{KL}(P||Q) = \sum_x P(z) \log\left(\frac{P(z)}{Q(z)}\right). \tag{9}$$

4 | APPLICATION

Three graph datasets are used as testbeds, including a synthetic generated spatially embedded network dataset, the

ALGORITHM 2 Pseudocode for spatial environment generation

Input: number of distributions t, distribution covariances $\vec{\sigma}$, distribution mean values $\vec{\mu}$.

Output: $\boldsymbol{S}$

$\boldsymbol{S} = O_{(500,500)}$

for $i \in (0, t)$ **do**

Select a random location $\boldsymbol{p}$ within $\boldsymbol{S}$.

$N(\boldsymbol{p}|\mu_i, \sigma_i) = \frac{1}{(2\pi)^{\sigma_i/2}} \frac{1}{|\Sigma|^{1/2}} exp\{-\frac{1}{2}(\boldsymbol{p} - \boldsymbol{\mu_i})^T \Sigma^{-1}(\boldsymbol{p} - \boldsymbol{\mu_i})\}$

$\boldsymbol{S} = \boldsymbol{S} + N$

end for

$\boldsymbol{S} = \frac{\boldsymbol{S} - min(\boldsymbol{S})}{max(\boldsymbol{S}) - min(\boldsymbol{S})}$

ALGORITHM 3 Pseudocode for single spatially embedded network generation

Input: number of nodes n, environment $\boldsymbol{S}$

Output:spatially embedded graph G

for $i \in (1, n)$ **do**

$p_i = (U(0, 500), U(0, 500))$

end for

for $i \in (1, n)$ **do**

$r_i = \boldsymbol{S}_{[p_i[0]-25:p_i[0]+25, p_i[1]-25:p_i[1]+25]}$

for $j \in (i, n)$ **do**

$r_j = \boldsymbol{S}_{[p_j[0]-25:p_j[0]+25, p_j[1]-25:p_j[1]+25]}$

if $d(p_i, p_j) < 200$ & $std(r_i - r_j) < 0.1$ **then**

$e_{(i,j)} = 1$

else

$e_{(i,j)} = 0$

end if

end for

end for

$G = (V, E)$

California Highway Network, and the New Jersey Power Transmission Network. The synthetically generated spatially embedded network dataset is used as an ideal test scenario, where the graphs are strictly generated by their distance and spatial feature similarities. On the other hand, the California Highway Network and New Jersey Power Transmission Network are networks collected from real-world datasets, whose intricate connection patterns are more complex than the synthetically generated dataset.

4.1 | Synthetic graph dataset

A synthetic spatially embedded graph generator is proposed to generate sufficient spatially embedded graphs. The generator includes two steps, spatial environment generation and graph generation. The spatial environment is generated by summarizing multiple randomly generated two-dimensional Gaussian distributions. The spatially embedded graphs are generated based on the nodes' distances and regional similarities. The pseudocode of spatial environment generation is shown in Algorithm 2, and the pseudocode of spatially embedded network generation is shown in Algorithm 3. In Algorithm 2, a two-dimensional environment $\boldsymbol{S}$ is first initialized by a zero matrix with the size of 500×500. t is a predefined number that controls how many Gaussian distributions are utilized. $\vec{\sigma}$ and $\vec{\mu}$ are a list of covariance values and mean values of these two-dimensional Gaussian distributions. The $\vec{\sigma}$ defines the diagonal values of the covariance matrix. The lengths of $\vec{\sigma}$ and $\vec{\mu}$ equal the predefined number t. Each element of $\vec{\sigma}$ and $\vec{\mu}$ is a two-dimensional vector. For each time, a random location $\boldsymbol{p}$ within the space $\boldsymbol{S}$ is selected, and then a two-dimensional Gaussian distribution is generated. The distribution is added to the environment space $\boldsymbol{S}$. In the end, the final environment $\boldsymbol{S}$ is normalized, so that the maximum value of the environment is 1 and the minimum value is 0.

The purpose of Algorithm 2 is to provide a repeatable and straightforward way to randomly generate spatial environments. However, it is worth noting there are many alternative ways for the spatial environment generation, such as Gaussian random field (Pichot, 2016).

The algorithm for single spatially embedded network generation is shown in Algorithm 3. A predefined number n is used to control the number of nodes in the network. The nodes' coordinates are randomly selected by using a uniform distribution $U(0, 500)$ so that all nodes are located within the predefined space. For each node v_i, a regional value is determined by the window of $\boldsymbol{S}_{[p_i[0]-25:p_i[0]+25, p_i[1]-25:p_i[1]+25]}$. Therefore, the window size of the regional information is 50. The edges between all pairs of nodes are labeled as 1 if the node's distance is shorter than 200, and the standard deviation of the difference of their regional values is lower than 0.1. Otherwise, the edge is labeled as 0. The edges labeled with 1 represent real connect edges, and edges labeled with 0 represent the nonconnect edges. The final graph G is an undirected graph with the aforementioned determined nodes and edges.

A total of 1000 random spatially embedded graphs are generated by repeating the Algorithms 2 and 3. The random parameters used in the generation process are defined in Table 1. Specifically, the numbers of Gaussian distributions are randomly selected from 10 to 30. This range is subjectively defined to ensure the spatial space has enough uncertainty. The two-dimensional Gaussian distribution's covariances σ are randomly selected between 1000 to 8000 so that each distribution can have an appropriate impact

TABLE 1 Parameters used for synthetic graph generation.

Name	Description	Value
t	Number of distributions	$U(10, 30)$
σ	Covariance of distributions	$U(1000, 8000)$
μ	Mean value of distributions	$U(0, 500)$
n	Number of nodes	$U(7, 21)$

Abbreviation: U, the uniform random.

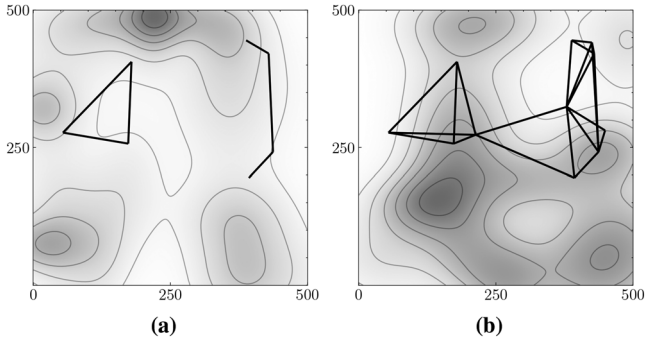


FIGURE 3 Examples of generated spatial space and graph.

influence within the 500×500 space. The mean values μ are selected from 0 to 500 so that the centers of the generated distributions are located randomly within the given space. Lastly, the node numbers of the graphs vary from 7 to 21. The numbers selected in Table 1 aim to generate spatially embedded networks with reasonable node numbers and edge numbers within a space of 500×500.

Figure 3 shows two examples of the graph generated by visualizing only the edges labeled as "1." As can be seen, the generated graph may have multiple connected components, such as Figure 3a. It may also be a single connected component as shown in Figure 3b. In the generated dataset, the edge number varies from 21 to 210, the node degree varies from 0 to 15, and the edge Euclidean length varies from 20.4 to 179.6.

4.2 | California Highway Network

A real-world infrastructure system, the California Highway Network (F. Li et al., 2005), is tested. Figure 4a (left side) shows the topology of the highway network and the area's surface elevation. In order to reduce the computational burden, the middle nodes of each single line are removed. The cleaned road network contains 1252 nodes and 1820 edges. The elevation data of the studied area is obtained from the NASA Shuttle Radar Topography Mission (SRTM) dataset (JPL, 2013). The resolution of the digital elevation data is 1 arc-second (approximately 30 m).

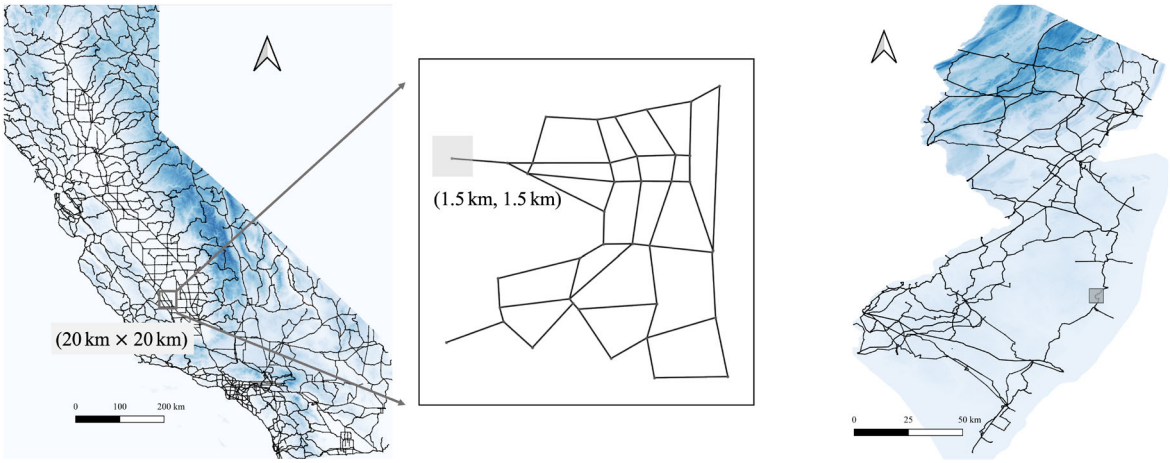
The network is sampled with a $20\ \text{km} \times 20\ \text{km}$ window size and the regional node spatial information is sampled by using a $1.5\ \text{km} \times 1.5\ \text{km}$ window, as shown in Figure 4a (right side). In the final sampled subgraph dataset, the node numbers of subgraphs range from 3 to 57, and the edge numbers range from 3 to 1596. Regarding the considered features, the node's position feature is the node's coordinates (lat and lon). The node's point feature includes the elevation value (L), the population density (D), and the median house value (H) of where the node is located. The node's regional feature includes the regional elevation value (r) within a $1.5\ \text{km} \times 1.5\ \text{km}$ window. The population density and median house value information are obtained from the U.S. Census Bureau (U.S. Census Bureau, 2024), which have been considered for understanding the relationship between road networks and socioeconomic factors in previous studies (Hu et al., 2018).

4.3 | New Jersey Power Transmission Network

Another real-world infrastructure system, power transmission networks, is considered in this study to validate the generic of the proposed model. Figure 4b shows the transmission network used in New Jersey, USA. The transmission network data are obtained from the Homeland Infrastructure Foundation-level Data (HIFLD) (HIFLD, 2024), which contains the national-wide transmission network varying from 69 kV up to 765 kV. The transmission network is relatively sparse compared to the road network.

The same graph-cleaning strategy with the road networks is used, that is, removing the middle nodes of long edges. After cleaning, the network contains 251 nodes and 464 edges. In addition, the subgraphs of the power system are sampled with a $25\ \text{km} \times 25\ \text{km}$ window size. The regional information of each node is sampled with a $1.5\ \text{km} \times 1.5\ \text{km}$ window. An example of the sampling window is also shown in Figure 4, where the larger rectangular is the subgraph sampling size. The node's position features, the node's point features, and the node's 'regional spatial features have the same definitions and data sources as the road network.

Table 2 shows the data sources and data types used for the California Highway Network and New Jersey Power Transmission Network. Specifically, the node's position feature is the node's coordinates, i.e. the latitude and longitude ([lat, lon]). The node's point features are the one-dimensional features located at the node's position, that is, the node's point elevation value (L), the population density (D), and the median house value (H). The node's regional feature is the elevation change within a region, denoted by r.



(a) California Highway Network **(b)** New Jersey Power Network

FIGURE 4 California Highway Network (left: topology of road network. right: illustrative of a random sample) and New Jersey Power Network.

TABLE 2 Node feature for network generation.

Symbol	Description	Source	Type
lat	Node latitude coordinate	Original map	$\mathcal{P}$
lon	Node longitude coordinate	Original map	$\mathcal{P}$
r	Node's regional elevation	NASA SRTM	$\mathcal{R}$
L	Node's point elevation	NASA SRTM	$\mathcal{X}$
D	Population density of the node's location	U.S. Census Bureau	$\mathcal{X}$
H	Median house value	U.S. Census Bureau	$\mathcal{X}$

Abbreviations: D, population density; H, Median house value; L, node's point elevation; lat, node's latitude; lon, node's longitude; NASA SRTM, NASA Shuttle Radar Topography Mission (SRTM) dataset; $\mathcal{P}$, the node's position feature; $\mathcal{R}$, the node's regional feature; r, regional elevation; $\mathcal{X}$, the node's point feature.

TABLE 3 Feature handling.

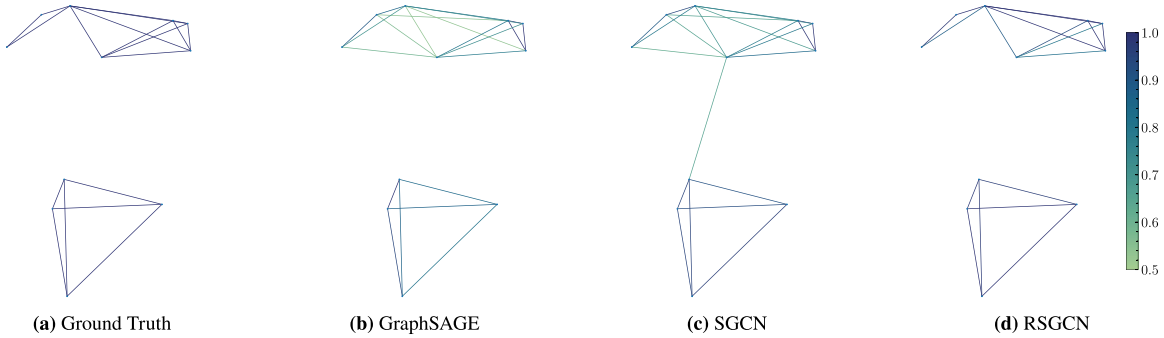
Model	$\mathcal{P}$	$\mathcal{X}$	$\mathcal{R}$
GraphSAGE	FC ($\mathcal{P}$ and $\mathcal{X}$ concatenated)		-
SGCN	FC	FC	-
RSGCN	FC	FC	CNN

Abbreviations: CNN, convolutional neural network; FC, fully connected neural network; RSGCN, regional spatial graph convolutional network; SGCN, spatial graph convolutional networks.

4.4 | Network reconstruction models

Two geometric deep learning models are used as benchmarks for comparison purposes, that is, the GraphSAGE model (Hamilton et al., 2017) and the SGCN model (Danel et al., 2020). The considered models are trained with the same subgraph data but utilize the node features in distinct ways. Specifically, the GraphSAGE model utilizes node position and point spatial feature homogeneously by concatenating these values as a single vector, which can be represented by $[\mathcal{P}, \mathcal{X}]$. On the contrary, the SGCN model separates the processing of node position and point spatial features. Node coordinates are processed by a fully connected neural network, while the node's point features are processed by another fully connected neural network. As a result, the node feature can be represented by two separate vectors $[\mathcal{P}]$ and $[\mathcal{X}]$. Previous studies have demonstrated that the SGCN model has outperformed traditional GCNs in multiple MoleculeNet benchmarks (Z. Wu et al., 2018). As introduced in Section 3, the RSGCN model handled the node's position feature $\mathcal{P}$, regional feature $\mathcal{R}$, and point feature $\mathcal{X}$ independently. Table 3 outlines the key differences in the feature-handling approaches of considered models.

The final architectures and hyperparameters of each model are determined by avoiding significant overfitting or underfitting. For the GraphSAGE model, two GraphSAGE layers are used for embedding the network into a latent space, which is named as encoding process. After network encoding, two fully connected neural networks are used for estimating the edge existence probability based on the embedded node features, which is named as decoding process. For the SGCN model, each SGCN layer contains two fully connected neural networks. One of the neural networks is used to procedure the node's position feature, and the other neural network is used to procedure the node's features in the graph convolutional process. Three SGCN layers are used for the network encoding, and another three fully connected neural networks are used for the network encoding. Lastly, the developed RSGCN model contains two developed RSGCN layers for encoding and



(a) Ground Truth **(b)** GraphSAGE **(c)** SGCN **(d)** RSGCN

FIGURE 5 Rebuilt results of the synthetic graph (colors indicate the predicted edge existence probability, color bar ranges from 0.5 to 1.0).

TABLE 4 Inputs and parameter numbers of each model.

Model	Input format
GraphSAGE	$[\text{lat}, \text{lon}, L, D, H]$
SGCN	$[\text{lat}, \text{lon}] + [L, D, H]$
RSGCN	$[\text{lat}, \text{lon}] + [L, D, H] + [r]$

Abbreviations: D, population density; H, Median house value; L, node's point elevation; lat, node's latitude; lon, node's longitude; r, regional elevation; RSGCN, regional spatial graph convolutional network; SGCN, spatial graph convolutional networks.

three fully connected neural networks for decoding, as introduced in Section 3.3. Table 4 shows the final considered features of each model used in highway road network and power network. It can be seen the GraphSAGE model concatenates all inputs as a vector and processes them using fully connected neural networks. For the RSGCN model, the input two-dimensional matrix size is 50×50, and the convolutional kernel size is 3. The final embedded node feature size is 128. All three models are trained for 600 epochs. Detailed parameters and configurations of the considered graph learning models can also be found in the GitHub repository.

5 | RESULTS

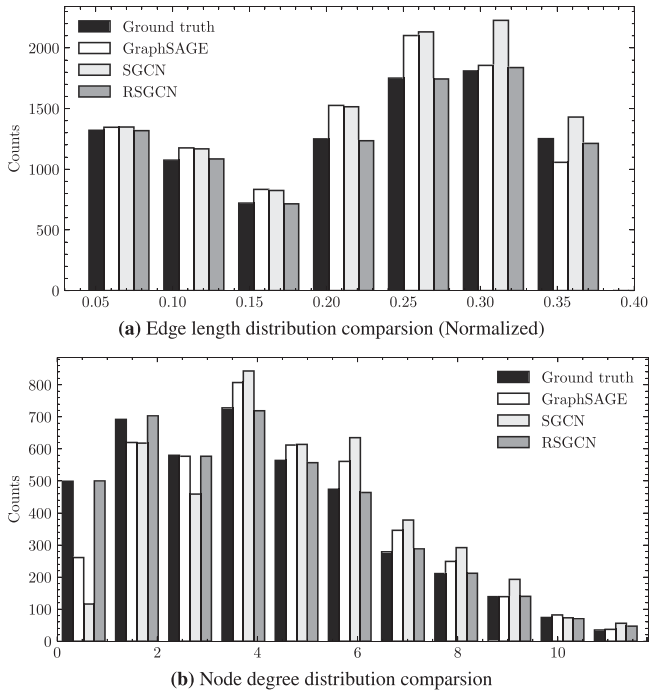
5.1 | Synthetic graph dataset

The randomly generated synthetic graph is a set of spatially embedded networks as described in Section 4.1. The node's point feature is the spatial environment value where the point is located. The node's position feature is the point's coordinates, and the node's regional spatial feature is a 50×50 spatial window size where the node is centered.

The GraphSAGE, SGCN, and RSGCN models are trained with 70% of the generated graphs and then tested by the remaining 30% graphs. Figure 5 shows a random reconstructed graph from the testing set, where only edges with existence probabilities higher than 0.5 are shown. Figure 5a shows the original graph. This particular graph sample contains 55 edges and 11 nodes. The reconstructed graph by GraphSAGE (Figure 5b) shows that some of the edges are rebuilt with relatively low confidence. There are also more edges established compared to the original graph. Similar issues also happened to the SGCN model as shown in Figure 5c. Several nonexistent edges were also predicted as existent edges, such as the edge between two connected components. Compared to the GraphSAGE and SGCN models, the RSGCN predicted the edge existence with 100% accuracy for this sample.

The predicted distributions of edge lengths and node degrees for all graphs in the testing set are compared in Figure 6. The bars filled with solid colors represent the data in the testing set, which also serves as the ground truth. The bars marked with various hatch patterns correspond to the results predicted by using all considered models, respectively. It can be observed that the graphs reconstructed by GraphSAGE and SGCN tend to have more edges compared to those reconstructed by RSGCN, with most bins being higher than in the original data. Moreover, the networks rebuilt by the GraphSAGE and SGCN models exhibit fewer nodes with small degrees and more nodes with higher degrees, as demonstrated in the bottom row of Figure 6. In contrast, the graphs rebuilt by the RSGCN model show edge length and node degree distributions more similar to those of the original data.

The performance of all three models on both training and testing sets can also be quantitatively evaluated, as shown in Table 5. The model with the best performance is indicated by the bold values. The RSGCN model outperformed the other models in all considered metrics. Particularly, the RSGCN achieved an overall prediction accuracy of 94%, which is 5.1% higher than the SGCN model and 6.65% higher than the GraphSAGE model in the testing set. In addition, the K-L divergence values for both edge length distribution and node degrees are much smaller than those of the GraphSAGE and SGCN models. A smaller value of K-L divergence indicates more similar



(a) Edge length distribution comparsion (Normalized)

(b) Node degree distribution comparsion

FIGURE 6 Synthetic dataset comparison. RSGCN, regional spatial graph convolutional network; SGCN, spatial graph convolutional networks.

TABLE 5 Comparison of models' performance on synthetic graph set.

Model	F1 score		K-L(E)		K-L(N)	
	Train	Test	Train	Test	Train	Test
GraphSAGE	0.18	0.19	0.007	0.005	0.03	0.04
SGCN	0.89	0.88	0.002	0.003	0.14	0.17
RSGCN	**0.96**	**0.94**	$\mathbf{4e^{-4}}$	$\mathbf{1e^{-3}}$	$\mathbf{3e^{-3}}$	$\mathbf{4e^{-3}}$

Note: The bold values represent the best performance.
Abbreviations: K-L(E), Kullback–Leibler (edge length distribution); K-L(N), Kullback–Leibler (node degree distribution); RSGCN, regional spatial graph convolutional network; SGCN, spatial graph convolutional networks.

distributions. The model shows a slightly better performance on the training set. However, the performance differences between the training and testing sets are minor, indicating appropriate model parameters have been selected.

5.2 | California Highway Network

In addition to testing the models in a synthetically generated dataset, the GraphSAGE, SGCN, and RSGCN models are also used to rebuild a real-world infrastructure system: a California Highway Network. The complete highway network is partitioned into 1252 subgraphs. Seventy percent of the subgraphs are used for training, and 30% are used for testing. Table 6 shows the final performance of the training set and testing set of each model, respectively. The

TABLE 6 Performance on road network's training and testing sets.

Model	F1 score		K-L(E)		K-L(N)	
	train	test	train	test	train	test
GraphSAGE	0.55	0.56	0.16	0.14	0.47	0.52
SGCN	0.78	0.77	**0.007**	0.008	0.18	**0.19**
RSGCN	**0.90**	**0.89**	0.008	**0.007**	**0.16**	**0.19**

Note: The bold values represent the best performance.
Abbreviations: K-L(E), Kullback–Leibler (edge length distribution); K-L(N), Kullback–Leibler (node degree distribution); RSGCN, regional spatial graph convolutional network; SGCN, spatial graph convolutional networks.

results show that the models performed relatively similarly on both the training and testing sets, indicating that they are neither overfitting nor underfitting. Furthermore, the RSGCN model significantly outperforms the SGCN and GraphsSAGE model based on the F1 score. The table also shows that the K-L divergences of edge and node distributions in the predicted results of the RSGCN model are similar to those of the SGCN model. This similarity may be partly attributed to the fact that all subgraphs in the testing set have similar node degrees and edge lengths, resulting in minor differences in distributions.

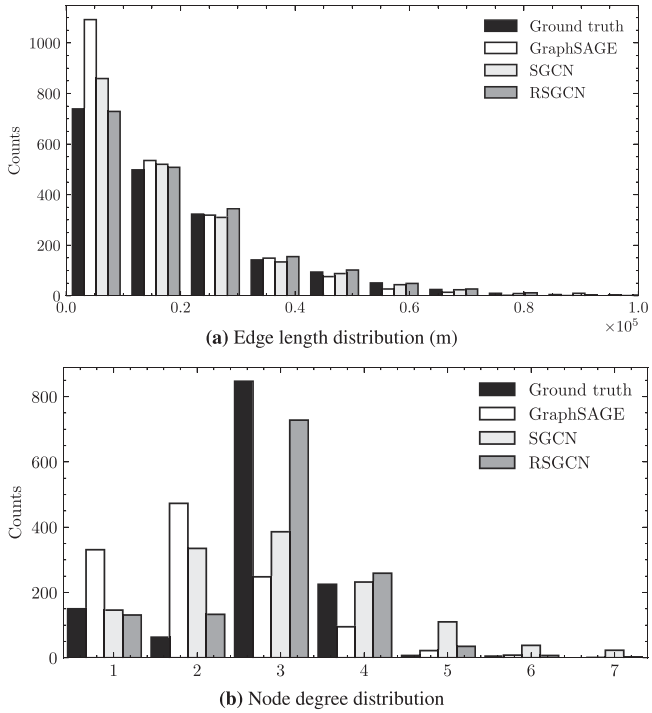
Based on the sampling strategy, it is understandable that edges may overlap between the training set and test sets, even though they are in subgraphs with different structures. In order to better evaluate the model's performance, this study also identified 2985 edges that only exist in the testing set. These "nonoverlap" edges include 293 edges labeled with 1 and 2692 edges labeled with 0. The model accurately predicted 269 edges labeled with 1 and 2691 edges labeled with 0. The overall F1 score for these "nonoverlap" edges is 0.85.

Figure 7 shows the original highway network and the rebuilt networks. The color of each edge represents the predicted existence probabilities. The edges connected by the *"probability relax"* strategy are labeled with a value of 0.5, which is the lowest connection probability shown in the figures. Figure 7b–d shows the networks reconstructed by GraphSAGE, SGCN, and RSGCN, respectively. It can be seen the GraphSAGE model only predicted a few edges whose existence probabilities are higher than 0.5. As a result, the rebuilt graph is very sparse and cannot accurately reflect the real network topology pattern of the original network. Compared to the GraphSAGE model, the SGCN model predicts more existing edges, as shown in Figure 7c. The result demonstrates the benefits of separating the process of the node's point spatial features and the node's position features. However, many edges, especially the edges located in the middle area of the graph, were not accurately reconstructed.

Lastly, the RSGCN significantly outperformed the GraphSAGE and SGCN models, as shown in 7d. A total of 19 edges were established based on the *"probability*

(a) Original network **(b)** GraphSAGE **(c)** SGCN **(d)** RSGCN

FIGURE 7 Road modeling results. RSGCN, regional spatial graph convolutional network; SGCN, spatial graph convolutional networks.



(a) Edge length distribution (m)

(b) Node degree distribution

FIGURE 8 Road network's edge length distribution (upper row) and node degree distribution (bottom row) of different models. RSGCN, regional spatial graph convolutional network; SGCN, spatial graph convolutional networks.

relax" strategy. Most of the edges were predicted accurately by the RSGCN model. In addition, the RSGCN model successfully rebuilt the long and short edges of the original graph. Although the confidence about the edge existence is relatively low in the middle area, the overall predicted existence probabilities are higher than those of the other models.

In order to obtain a more detailed comparison, the edge length and node degree distributions of all predicted graphs are compared in Figure 8. The edge lengths are

TABLE 7 Comparison of models' performance on the full road network.

Model	F1	K-L (Edge)	K-L (Node)	*T* (min)
GraphSAGE	0.561	0.137	0.523	43
SGCN	0.781	0.008	0.387	62
RSGCN	**0.895**	**0.001**	**0.047**	136

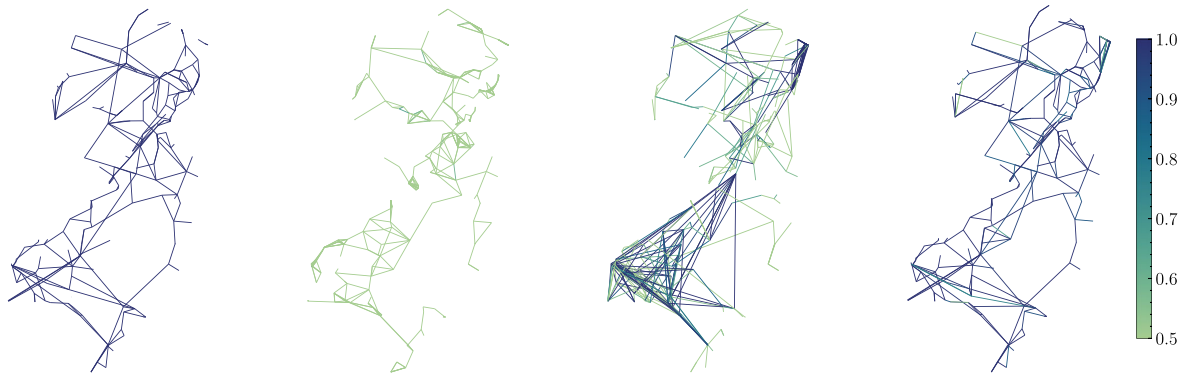
Note: The bold values represent the best performance.
Abbreviations: K-L (Edge), Kullback–Leibler (Edge); K-L (Node), Kullback–Leibler (Node); RSGCN, regional spatial graph convolutional network; SGCN, spatial graph convolutional networks; *T*, time of the training process in minutes.

Euclidean distance of nodes' coordinates. It can be seen that both GraphSAGE and SGCN models tend to overestimate short edges but slightly underestimate long edges. These two models also significantly underestimate nodes whose node degrees are 3. On the other hand, the RSGCN presents a better representation of the original graph's edge length and node degree distributions.

The graph modelings' performance on the full road network is shown in Table 7. The prediction accuracy of the RSGCN is 89.5%, which is 33.4% higher than the GraphSAGE model and 11.4% higher than the SGCN model. Meanwhile, the K-L divergence of the edge length and node distribution is also significantly smaller. The results further demonstrate the necessity of considering regional spatial information for spatially embedded graphs.

5.3 | New Jersey Power Transmission Network

The New Jersey Power Transmission Network serves as an additional testbed for validating the generality of the developed model. The GraphSAGE, SGCN, and the developed RSGCN were trained and tested by 70% and 30% sampling



(a) Original network **(b)** GraphSAGE **(c)** SGCN **(d)** RSGCN

FIGURE 9 NJ transmission network. RSGCN, regional spatial graph convolutional network; SGCN, spatial graph convolutional networks.

TABLE 8 Performance on NJ Power Network's training and testing sets.

Model	F1 score		K-L(E)		K-L(N)	
	Train	Test	Train	Test	Train	Test
GraphSAGE	0.11	0.11	0.20	0.21	2.20	3.11
SGCN	0.65	0.51	0.008	0.010	0.21	0.32
RSGCN	0.91	0.89	0.001	0.002	0.02	0.03

Abbreviations: K-L(E), Kullback–Leibler (edge); K-L(N), Kullback–Leibler (node); RSGCN, regional spatial graph convolutional network; SGCN, spatial graph convolutional networks.

TABLE 9 Comparison of models' performance on NJ power network.

Model	F1	K-L (Edge)	K-L (Node)	*T* (min)
GraphSAGE	0.12	0.270	0.509	28
SGCN	0.52	0.102	0.284	47
RSGCN	**0.90**	**0.002**	**0.016**	98

Note: The bold values represent the best performance.
Abbreviations: K-L (Edge), Kullback–Leibler (Edge); K-L (Node), Kullback–Leibler (Node); RSGCN, regional -spatial graph convolutional network; SGCN, spatial graph convolutional networks; *T*, time of the training process in minutes.

graphs. Table 8 shows the models' performance in both training and testing sets, which demonstrates no significant overfitting or underfitting in the training process. The introduced RSGCN model also outperforms the other two models in all metrics.

Similar to the case study of California Highway Network, a total of 662 "nonoverlap" edges are identified, including 63 edges labeled with 1 and 599 edges labeled with 0. The model accurately predicted 61 edges labeled with 1 and 586 edges labeled with 0. The overall F1 score for unseen edges is 0.89.

The rebuilt networks of the New Jersey transmission line by using all considered models are shown in Figure 9. Figure 9a shows the original transmission network after simplifying, and Figure 9b–d shows the rebuilt transmission topologies by the GraphSAGE, SGCN, and RSGCN models, respectively. Figure 9b shows that the GraphSAGE model struggles to accurately model the network topology. Almost all connections are made based on the *"probability relax"* strategy, rather than being directly inferred by the model. This issue is likely because of the highly imbalanced edge labels in the training set, which makes the model tend to predict most edges as "nonexistent." As a result, the reconstructed network is significantly sparser compared to the original map.

The SGCN model presents a slight improvement in network reconstruction performance compared to GraphSAGE by establishing more edges. However, it also generates more edges that should not exist. Consequently, the predicted edges cannot accurately reflect the original network's structure. Some nodes have extremely high node degrees, such as nodes in the bottom left and upper right regions. The SGCN model also struggles to accurately establish short edges as shown in Figure 9c.

Compared to the SGCN and GraphSAGE models, the RSGCN model presents a better network reconstruction. The rebuilt network accurately reflects the connection patterns of the original network, as shown in Figure 9d. A total of seven edges are established based on the *"probability relax"* strategy. In addition, there are a large portion of the edges are predicted with relatively high confidence, and only a few edges are predicted with a confidence of lower than 70%. However, some edges are mispredicted even though their existence probabilities are high, such as some short edges located in the top right region.

Table 9 summarizes the quantitative evaluation results of all models on the New Jersey Power Network. It can

be seen that the GraphSAGE model failed to predict most of the edges, resulting in a low F-1 score and high K-L divergence. The performance of the SGCN model was slightly improved to the GraphSAGE model, but the F1 score is only around 0.52. Compared to the SGCN and GraphSAGE models, the RSGCN model's F1 score achieved 0.907, demonstrating the superiority of the model and the importance of considering regional information.

6 | DISCUSSION AND LIMITATIONS

The results show that the developed RSGCN model presents a higher topology reconstruction accuracy compared to the GraphSAGE and SGCN models. The results also indicate a higher improvement when applying the RSGCN model to real-world datasets. Part of the reason can be attributed to the complexity of the testbeds. The intricate patterns within synthetic networks are easier to capture, leading to similar performance levels among all considered models. Furthermore, the design, management, and operation of real-world infrastructure systems are often constrained by geographical factors such as elevation changes and land usage (Y. Wang et al., 2017). Incorporating the regional elevation change into the model for highway networks and power networks thereby enhanced the model's accuracy. In addition, it is worth noting that the selection of node regional features should have a direct influence on the network structure, otherwise, additional noise may be introduced into the RSGCN training process.

A few factors have been subjectively selected in this study in order to evaluate the performance of the proposed RSGCN model and network reconstruction framework. To make this study more generic for the other applications, the influences of these factors on the model performance are discussed below.

The influence of sampling strategy: A higher accuracy was also observed in the rebuilt highway and power networks. The results are partially caused by the ensembling process, as all of the subgraphs in the training set and test set are used. The machine learning models usually have better accuracy in the training set due to the inherent learning mechanisms. However, the results can still demonstrate good performance when comparing the results between the training and testing sets as shown in Tables 6 and 8. In addition, the proposed model also successfully established 2820 of 2985 "nonoverlap" edges in the California Highway test set and 647 out of 662 "nonoverlap" edges in the NJ Power Network. Lastly, the performance of the proposed model in rebuilding synthetic graphs on the testing set further validates its effectiveness, as none of the graphs in the testing set were seen during the training process 5.

The influence of partition and regional window sizes: The partition size has a major influence on the sizes of sampled graphs. A smaller partition size leads to a smaller sampled graph, resulting a less training time and computational resources required. However, it will also exclude many edges that are longer than the size, lowering the final network reconstruction accuracy. On the other hand, a larger subgraph size requires more computational resources, such as graphics processing unit (GPU) memory. It will also increase the learning complexity as the more nodes of a subgraph, the more edges exist in the corresponding fully connected networks. Compared to the partition size, the regional window size influences how large elevation changes around each node should be considered. A larger window size allows more spatial changes can be considered but also requires more computational resources for the training process. The partition sizes used in this study are selected based on the longest edges in each dataset. The window sizes are determined based on the digital elevation model resolution and computing resources.

The influence of probability threshold: In this study, a probability existence threshold of 0.5 is selected. This threshold is selected to provide an unbiased initial balance point considering the edge existence prediction is a binary classification task. Lowering this threshold allows more edges to be classified as "connected" but introduces more false positive errors. On the other hand, increasing this threshold will lead to more type II errors. In reality, minimizing which type of error is often subjective and requires a trial-and-error process.

The broad application of the proposed RSGCN model: The developed RSGCN model is also a generic network representation model, which can be applied to a wide of real-world applications when using different models to decode the embedded node features. For example, an accurate network representation is the key to traffic prediction in a traffic network (Tang & Zeng, 2022). It can also make more optimal decisions for networked infrastructure systems when replacing conventional artificial neural networks with more advanced graph neural networks (Chen et al., 2021). Lastly, a better network representation can also benefit from more accurate edge feature prediction (Yuan et al., 2022).

Although the RSGCN model outperforms conventional geometric-based deep learning algorithms, there are some limitations regarding the model itself and the network generation framework.

The guarantee of network connectivity: It should be noted that the proposed RSGCN model and network modeling framework are used to predict the connectivity probability between all pairs of nodes at the same time. When using a threshold to determine the edge existence, the network connectivity may not be able to be strictly guaranteed like

graph-theory-based methods (Aksoy et al., 2019). Such an inability might limit the applicability of the proposed model in real-world infrastructure networks. A variety of strategies can be considered as potential solutions. In addition to the *"probability relax"* strategy proposed in this study, lowering the global probability threshold can also increase the predicted graph connectivity. In addition, incorporating the predicted edge existence probability into traditional network modeling algorithms can also be a potential solution, such as small-world graphs or physical-informed planning strategies (T.-Y. Zhang et al., 2024). Fast and efficient machine learning algorithms are emerging in recent years, such as dynamic ensemble learning and dynamic classification algorithms (Rafiei & Adeli, 2017; Alam et al., 2020). More advanced methods will be investigated in future studies.

Lack of node generation in the framework: Another limitation of this study is it does not include node generation in the network modeling. This study assumes these node locations are available, some previous studies also used similar assumptions (Ahmad et al., 2022; Sitzenfrei et al., 2020). However, it is known that the nodes' locations are also constrained by their spatial space, and generating nodes is a critical topic in network modeling (Vaccariello et al., 2020). Future studies will include the node generation process.

High computational demand due to multimodal data: The parameter number of the proposed RSGCN model is significantly larger than that of the GraphSAGE model and SGCN model. Four A100 GPUs were used for the models' training. The training process took around 2 h for the California Highway Network and around 1.5 h for the NJ Power Network. The main reason for the different training times is caused by the different number of subgraphs. However, it should be noted that due to the similar sampling window size and considered features, the utilized GPU memory has no significant difference when training these two networks. Increasing the number and dimensions of considered features will increase the memory needs. Future work will focus on improving the learning efficiency and reducing the model size.

Limited considered features: Lastly, for illustration purposes, only the node coordinates, surface elevation, population density, and median house value were used for the network reconstruction. It is known that a large number of factors may influence the network topology, especially socioeconomic and land use factors. Identifying the optimal factors for network structure prediction is still challenging. Further studies will investigate the importance of different factors for improving the topology prediction performance.

7 | CONCLUSION

A novel geometric-based deep learning architecture, the RSGCN model, was developed for complex network representation and intricate pattern capturing. The introduced model can process node features with different data dimensions, such as the vectorized node's position feature, vectorized node's point feature, and two-dimensional node's regional feature. The developed RSGCN model was integrated with a partition-then-ensembling framework and was used for predicting the edge existence probability within the spatially embedded networks. In addition to the RSGCN model, two other geometric-based deep learning models, GraphSAGE and SGCN, were used as benchmarks. The results have demonstrated the superiority of the developed RSGCN model and the importance of considering regional features for spatially embedded network representation. The RSGCN model outperformed the second-best model, the SGCN model, by 5.1%, 11.4%, and 38.0% in the perspective of F1 scores. Furthermore, the developed partition-then-ensembling framework efficiently addressed the challenge of large networks by sampling a large network into a batch of subgraphs.

Although the RSGCN model was only used for network reconstruction, it is essentially a technique for graph representation and network intricate patterns capture. Considering its ability to leverage different types of node features, the proposed method can be applied to other challenges in infrastructure systems. Future studies will investigate the wide applications of the proposed RSGCN model.

ACKNOWLEDGMENTS

This work has been supported by a grant from the Energy Research Fund administered by the Andlinger Center for Energy and the Environment as well as the School of Engineering and Applied Science (SEAS) Seed Grant at Princeton University. Furthermore, the authors are pleased to acknowledge that the work reported on in this paper was substantially performed using the Princeton Research Computing resources at Princeton University, which is a consortium of groups led by the Princeton Institute for Computational Science and Engineering (PICSciE) and Office of Information Technology's Research Computing.

ORCID

Xudong Fan https://orcid.org/0000-0002-8924-0179
Jürgen Hackl https://orcid.org/0000-0002-8849-5751

REFERENCES

Ahmad, N., Chester, M., Bondank, E., Arabi, M., Johnson, N., & Ruddell, B. L. (2022). A synthetic water distribution network

model for urban resilience. *Sustainable and Resilient Infrastructure*, *7*(5), 333–347.

Aksoy, S. G., Purvine, E., Cotilla-Sanchez, E., & Halappanavar, M. (2019). A generative graph model for electrical infrastructure networks. *Journal of Complex Networks*, *7*(1), 128–162.

Alam, K. M. R., Siddique, N., & Adeli, H. (2020). A dynamic ensemble learning algorithm for neural networks. *Neural Computing and Applications*, *32*(12), 8675–8690.

Cai, L., Zhang, L., Liang, Y., & Li, J. (2022). Discovery of urban functional regions based on Node2vec. *Applied Intelligence*, *52*(14), 16886–16899.

Cai, T., Liang, T., & Rakhlin, A. (2017). On detection and structural reconstruction of small-world random networks. *IEEE Transactions on Network Science and Engineering*, *4*(3), 165–176.

Chahinian, N., Delenne, C., Commandré, B., Derras, M., Deruelle, L., & Bailly, J.-S. (2019). Automatic mapping of urban wastewater networks based on manhole cover locations. *Computers, Environment and Urban Systems*, *78*, 101370.

Che, X., Zheng, Y., Chen, X., Song, S., & Li, S. (2022). Decoding color visual working memory from EEG signals using graph convolutional neural networks. *International Journal of Neural Systems*, *32*(02), 2250003.

Chen, S., Dong, J., Ha, P. Y. J., Li, Y., & Labi, S. (2021). Graph neural network and reinforcement learning for multi-agent cooperative control of connected autonomous vehicles. *Computer-Aided Civil and Infrastructure Engineering*, *36*(7), 838–857.

Cover, T. M., & Thomas, J. A. (2006). *Elements of information theory* (2nd ed). Wiley-Interscience.

Danel, T., Spurek, P., Tabor, J., Śmieja, M., Struski, Ł., Słowik, A., & Maziarka, Ł. (2020). Spatial graph convolutional networks. In *Neural information processing, communications in computer and information science* (pp. 668–675). Springer International Publishing.

De Bacco, C., Power, E. A., Larremore, D. B., & Moore, C. (2017). Community detection, link prediction, and layer interdependence in multilayer networks. *Physical Review E*, *95*(4), 042317.

Dettmann, C. P., & Georgiou, O. (2016). Random geometric graphs with general connection functions. *Physical Review E*, *93*(3), 032313.

Ding, J., Liu, C., Zheng, Y., Zhang, Y., Yu, Z., Li, R., Chen, H., Piao, J., Wang, H., Liu, J., & Li, Y. (2024). *Artificial intelligence for complex network: potential, methodology and application*. arXiv. https://doi.org/10.48550/arXiv.2402.16887

Dong, S., Wang, H., Mostafizi, A., & Song, X. (2020). A network-of-networks percolation analysis of cascading failures in spatially co-located road-sewer infrastructure networks. *Physica A: Statistical Mechanics and its Applications*, *538*, 122971.

Dunton, A., & Gardoni, P. (2024). Generating network representations of small-scale infrastructure using generally available data. *Computer-Aided Civil and Infrastructure Engineering*, *39*(8), 1143–1158.

Ektefaie, Y., Dasoulas, G., Noori, A., Farhat, M., & Zitnik, M. (2023). Multimodal learning with graphs. *Nature Machine Intelligence*, *5*(4), 340–350.

Fan, X., Zhang, X., & Yu, X. B. (2022). A graph convolution network-deep reinforcement learning model for resilient water distribution network repair decisions. *Computer-Aided Civil and Infrastructure Engineering*, *37*(12), 1547–1565.

Grover, A., & Leskovec, J. (2016). Node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, (pp. 855–864). KDD '16. Association for Computing Machinery.

Hackl, J., & Adey, B. T. (2019). Modelling multi-layer spatially embedded random networks. *Journal of Complex Networks*, *7*(2), 254–280.

Hamilton, W., Ying, Z., & Leskovec, J. (2017). Inductive Representation Learning on Large Graphs. In *advances in neural information processing systems* (Vol *30*). Curran Associates, Inc.

HIFLD. (2024). Homeland infrastructure foundation-level data - HIFLD. Retrieved January 24, 2024, from https://hifld-geoplatform.hub.arcgis.com

Holland, P. W., Laskey, K. B., & Leinhardt, S. (1983). Stochastic block models: First steps. *Social Networks*, *5*(2), 109–137.

Hu, X., Wu, C., Wang, J., & Qiu, R. (2018). Identification of spatial variation in road network and its driving patterns: Economy and population. *Regional Science and Urban Economics*, *71*, 37–45.

JPL, N. (2013). NASA Shuttle Radar topography mission global 1 arc second. Jet Propulsion Laboratory, NASA, USA.

Li, F., Cheng, D., Hadjieleftheriou, M., Kollios, G., & Teng, S.-H. (2005). On trip planning queries in spatial databases. In Bauzer Medeiros, C., Egenhofer, M. J., & Bertino, E. (Eds.), *Advances in spatial and temporal databases* (pp. 273–290). Lecture Notes in Computer Science, volume 3633. Springer.

Li, J., Chen, C., Tong, H., & Liu, H. (2018). Multi-layered network embedding. In *Proceedings of the 2018 SIAM International conference on data mining (SDM)* (pp. 684–692). Society for Industrial and Applied Mathematics.

Lian, J., & Xu, F. (2022). Spatial enhanced pattern through graph convolutional neural network for epileptic EEG identification. *International Journal of Neural Systems*, *32*(09), 2250033.

Liao, W., Bak-Jensen, B., Pillai, J. R., Wang, Y., & Wang, Y. (2022). A review of graph neural networks and their applications in power systems. *Journal of Modern Power Systems and Clean Energy*, *10*(2), 345–360.

Ma, X., Zhou, H., & Li, Z. (2021). On the resilience of modern power systems: A complex network perspective. *Renewable and Sustainable Energy Reviews*, *152*, 111646.

Mao, J., Cao, L., Gao, C., Wang, H., Fan, H., Jin, D., & Li, Y. (2023). Detecting vulnerable nodes in urban infrastructure interdependent network. In *Proceedings of the 29th ACM SIGKDD conference on knowledge discovery and data mining* (pp. 4617–4627). KDD '23. Association for Computing Machinery.

Moeini, R., & Afshar, M. H. (2018). Extension of the hybrid ant colony optimization algorithm for layout and size optimization of sewer networks. *Journal of Environmental Informatics*, *33*(2), 68–81.

Molinero, C., & Hernando, A. (2020). *A model for the generation of road networks*. arXiv. https://doi.org/10.48550/arXiv.2001.08180

Nocera, F., & Gardoni, P. (2022). Selection of the modeling resolution of infrastructure. *Computer-Aided Civil and Infrastructure Engineering*, *37*(11), 1352–1367.

Peixoto, T. P. (2019). Network reconstruction and community detection from dynamics. *Physical Review Letters*, *123*(12), 128301.

Perozzi, B., Al-Rfou, R., & Skiena, S. (2014). DeepWalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on knowledge discovery and*

data mining (pp. 701–710). KDD '14. Association for Computing Machinery.

Pichot, G. (2016). *Algorithms for gaussian random field generation*. Report, INRIA Paris.

Pikovsky, A. (2018). Reconstruction of a random phase dynamics network from observations. *Physics Letters A*, *382*(4), 147–152.

Rafiei, M. H., & Adeli, H. (2016). A novel machine learning model for estimation of sale prices of real estate units. *Journal of Construction Engineering and Management*, *142*(2), 04015066.

Rafiei, M. H., & Adeli, H. (2017). A new neural dynamic classification algorithm. *IEEE Transactions on Neural Networks and Learning Systems*, *28*(12), 3074–3083.

Rafiei, M. H., & Adeli, H. (2018). Novel machine-learning model for estimating construction costs considering economic variables and indexes. *Journal of Construction Engineering and Management*, *144*(12), 04018106.

Sitzenfrei, R., Wang, Q., Kapelan, Z., & Savić, D. (2020). Using complex network analysis for optimization of water distribution networks. *Water Resources Research*, *56*(8), e2020WR027929.

Soltan, S., & Zussman, G. (2016). Generation of synthetic spatially embedded power grid networks. In *2016 IEEE Power and Energy Society General Meeting (PESGM)* (pp. 1–5). IEEE.

Tang, J., & Zeng, J. (2022). Spatiotemporal gated graph attention network for urban traffic flow prediction based on license plate recognition data. *Computer-Aided Civil and Infrastructure Engineering*, *37*(1), 3–23.

U.S. Census Bureau. (2024). American Community Survey 5-year data (2009-2022). Retrieved January 18, 2024, from https://www.census.gov/data/developers/data-sets/acs-5year.html

Vaccariello, E., Leone, P., & Stievano, I. S. (2020). Generation of synthetic models of gas distribution networks with spatial and multi-level features. *International Journal of Electrical Power & Energy Systems*, *117*, 105656.

Wang, Y., Zou, Y., Henrickson, K., Wang, Y., Tang, J., & Park, B.-J. (2017). Google Earth elevation data extraction and accuracy assessment for transportation applications. *PLoS ONE*, *12*(4), e0175756.

Wang, Z., Zhuang, D., Li, Y., Zhao, J., Sun, P., Wang, S., & Hu, Y. (2023). ST-GIN: An uncertainty quantification approach in traffic data imputation with spatio-temporal graph attention and bidirectional recurrent united neural networks. In *2023 IEEE 26th international conference on intelligent transportation systems (ITSC)* (pp. 1454–1459). IEEE.

Wu, J. (2017). *Introduction to convolutional neural networks*. National Key Lab for Novel Software Technology. Nanjing University. China, *5*, 495.

Wu, Z., Ramsundar, B., N.Feinberg, E., Gomes, J., Geniesse, C., S.Pappu, A., Leswing, K., & Pande, V. (2018). MoleculeNet: A benchmark for molecular machine learning. *Chemical Science*, *9*(2), 513–530.

Xu, D., Wang, Y., Peng, P., Lin, L., & Liu, Y. (2022). The evaluation of the urban road network based on the complex network. *IEEE Intelligent Transportation Systems Magazine*, *14*(3), 200–211.

Xu, Z., Ramirez-Marquez, J. E., Liu, Y., & Xiahou, T. (2020). A new resilience-based component importance measure for multi-state networks. *Reliability Engineering & System Safety*, *193*, 106591.

Yuan, F., Xu, Y., Li, Q., & Mostafavi, A. (2022). Spatio-temporal graph convolutional networks for road network inundation status prediction during urban flooding. *Computers, Environment and Urban Systems*, *97*, 101870.

Zhang, D., Zhou, C., Sun, D., & Qian, Y. (2022). The influence of the spatial pattern of urban road networks on the quality of business environments: The case of Dalian city. *Environment, Development and Sustainability*, *24*(7), 9429–9446.

Zhang, N., & Alipour, A. (2023). A stochastic programming approach to enhance the resilience of infrastructure under weather-related risk. *Computer-Aided Civil and Infrastructure Engineering*, *38*(4), 411–432.

Zhang, T.-Y., Yao, E.-J., Yang, Y., Yang, H.-M., & Wang, D. Z. W. (2024). Multi-network coordinated charging infrastructure planning for the self-sufficient renewable power highway. *Computer-Aided Civil and Infrastructure Engineering*. Advance online publication. https://doi.org/10.1111/mice.13196

Zheng, G., Liu, S., & Qi, X. (2012). Scale-free topology evolution for wireless sensor networks with reconstruction mechanism. *Computers & Electrical Engineering*, *38*(3), 643–651.

How to cite this article: Fan, X., & Hackl, J. (2024). Modeling of spatially embedded networks via regional spatial graph convolutional networks. *Computer-Aided Civil and Infrastructure Engineering*, 1–18. https://doi.org/10.1111/mice.13286