Deep Generative Models for Spatial Networks

Xiaojie Guo
xguo7@gmu.edu
George Mason University

Yuanqi Du*
ydu6@gmu.edu
George Mason University

Liang Zhao†
liang.zhao@emory.edu
Emory University

ABSTRACT

Spatial networks represent crucial data structures where the nodes and edges are embedded in a geometric space. Nowadays, spatial network data is becoming increasingly popular and important, ranging from microscale (e.g., protein structures), to middle-scale (e.g., biological neural networks), to macro-scale (e.g., mobility networks). Although, modeling and understanding the generative process of spatial networks are very important, they remain largely underexplored due to the significant challenges in automatically modeling and distinguishing the independency and correlation among various spatial and network factors. To address these challenges, we first propose a novel objective for joint spatial-network disentanglement from the perspective of information bottleneck as well as a novel optimization algorithm to optimize the intractable objective. Based on this, a spatial-network variational autoencoder (SND-VAE) with a new spatial-network message passing neural network (S-MPNN) is proposed to discover the independent and dependent latent factors of spatial and networks. Qualitative and quantitative experiments on both synthetic and real-world datasets demonstrate the superiority of the proposed model over the state-of-the-arts by up to 66.9% for graph generation and 37.3% for interpretability.

CCS CONCEPTS

• **Computing methodologies** → **Unsupervised learning**; **Neural networks**; *Generative and developmental approaches*; • **Mathematics of computing** → *Graph algorithms*; • **Information systems** → *Data mining*; • **Networks** → *Topology analysis and generation.*

KEYWORDS

Spatial network, disentangled representation learning, graph neural network, variational auto-encoder.

ACM Reference Format:
Xiaojie Guo, Yuanqi Du, and Liang Zhao. 2021. Deep Generative Models for Spatial Networks. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '21), August 14–18, 2021, Virtual Event, Singapore.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3447548.3467394

*Equally contributing as first author
†Corresponding author

KDD '21, August 14–18, 2021, Virtual Event, Singapore

ACM ISBN 978-1-4503-8332-5/21/08…$15.00
https://doi.org/10.1145/3447548.3467394

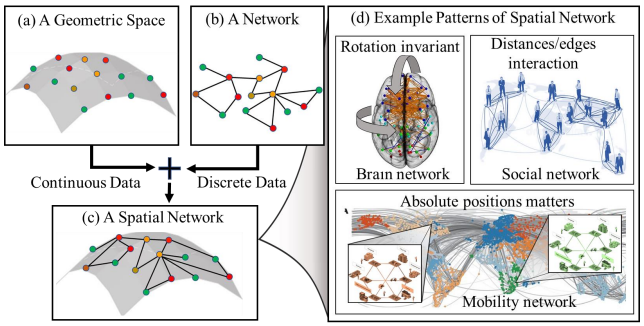

Figure 1: Spatial networks contain not only network and spatial information but also information describing their close interactions. The three real-world examples of spatial networks show the different patterns needed for different spatial networks: (1) a protein tertiary structure graph is invariant to the rotation in a geometric space; (2) two cities' absolute locations indicate key spatial heterogeneity information in their mobility networks; and (3) people who live nearer are more likely to be friend in social networks.

1 INTRODUCTION

Spatial and network data are both popular types of high-dimensional complex data that are being used in a wide variety of applications in the big data era. The study of spatial data usually focuses on the properties of continuous spatial entities under specific geometric patterns (Fig. 1(a)), while the analysis of network data concentrates on the properties of discrete objects and their pairwise relationships (Fig. 1(b)). Spanning these two data types, spatial networks represent a crucial data structure where the nodes and edges are embedded in a geometric space (Fig. 1 (c)). Nowadays, spatial network data is becoming increasingly popular and important, ranging from micro-scale (e.g., protein structures), to middle-scale (e.g., biological neural networks), to macro-scale (e.g., mobility networks). Spatial networks cannot be modeled using either spatial or network information individually, but require the simultaneous characterization of both the data and their interactions, which results in various patterns [5]. For example, a protein structure can be formalized as a spatial network with patterns that are invariant to rotation and translation. But in a mobility network, the absolute locations of the nodes are meaningful to indicate spatial heterogeneity for different nodes (e.g., cities) of networks, as shown in Fig. 1(d). Moreover, the interactions between the patterns of network topology and spatial features are also very important, as shown in the example of a social network in Fig. 1(d), where the edge formation can be dependent on the geodesic distance.

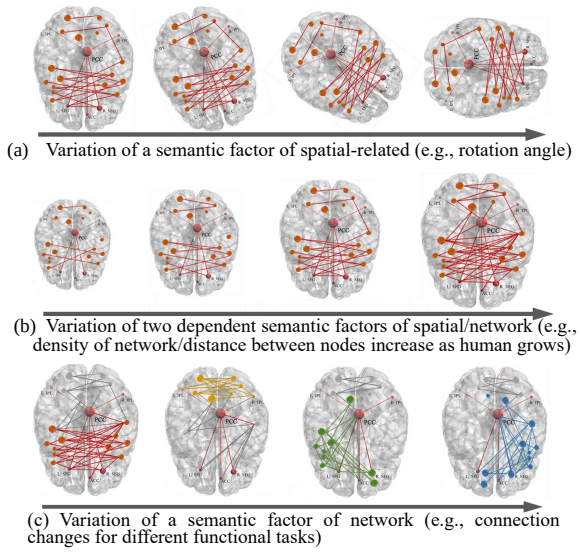

Figure 2: Visualizing the variations of three groups of semantic factors of (1) spatial, (2) spatial network, and (3) network for the spatial network example of brain network.

Modeling and understanding the generative process of spatial networks are vital for a wide variety of important applications, such as protein structure generative modeling [3, 18, 39], biological neural nets [1], and mobility network analysis [24]. Until now, the commonly-used models extend graph theory into spatial networks [5], resulting in models such as geometric graphs [38] and spatial small-world graphs [28]. These typically rely on a set of network generation principles predefined by human heuristics and knowledge. Such methods usually fit the properties that have been covered by the predefined principles very well, but are not as effective for those that have not been. Unfortunately, in many domains, the network properties and generation principles remain largely unknown, such as models that explain the mechanisms of mental diseases in brain networks like functional connectivity [1] and protein structure folding. This motivates us to find ways to directly learn the underlying spatial and graph-structure distribution patterns from the data without the needs to predefine the generation rules manually.

Recent advanced deep generative models, such as variational auto-encoders (VAE) [31], have made important progress towards modeling and understanding complex data, such as spatial data (e.g., point clouds) and graph data (e.g., molecules). The goal here is to first learn the underlying (low-dimensional) distribution of the objects and then generate the data by sampling this learned distribution. Despite many deep generative models that have been proposed for dealing with either spatial data or graph data individually, as yet the deep generative models for spatial networks remain to be explored which cannot be handled by existing techniques due to several significant challenges: **(1) Difficulty in capturing and separating various types of semantic factors**: There are three groups of semantic factors to be captured: one is related to spatial information that is independent from networks, such as the rotation of the brain network, as shown in Fig. 2(a); one is related to network information that is independent from spatial information, such as functional connectivity of brain network when people are doing different tasks, shown in Fig. 2(c); and one contains the semantic factors spanning both spatial and network dimensions that encompass the interactions between geometry and networks, such as the joint changes of size, shape, and connectivity of human brain network when growing, shown in Fig. 2(b). **(2) Difficulty in capturing the distribution that models the interaction between spatial and networks**: it is often necessary to capture the complex and various interaction patterns between spatial and network dimensions. For example, network features like friendship between two people may have mutual correlation with their spatial locations. Many real-world networks are planar which require that their edges do not cross in a plane. How to jointly learn the shared latent dimensions for both the continuous-valued spatial information and discrete-valued graph information is extremely challenging. **(3) Difficulty in preserving all the information in spatial networks**. As illustrated in Fig. 1, spatial network generative models are required to capture all the information available in spatial, network, and their joint dimensions. For example, both rotation invariant and rotation variant properties need to be modeled. Similarly, both transformation invariant and variant properties need also be characterized. Graph-structured information and their interaction with spatial information need also be covered in the model. This not only challenges the spatial network encoder and decoder, but also call for effective strategies in optimizing the information bottlenecks.

To address all the above challenges, here we for the first time propose a novel disentangled deep generative model for spatial networks. Specifically, a novel objective for spatial-network joint disentanglement is derived and proposed based on the variational autoencoder (VAE) from the perspective of an information bottleneck. To optimize the intractable objective, a novel Spatial-Network Disentangled Variational Auto-encoder (SND-VAE) model is proposed to discover the independent and dependent latent factors of spatial and networks. To deal with the information bottlenecks affecting spatial, network, and spatial-network-joint factors, a novel optimization strategy is proposed with a theoretical analysis. Finally, a new spatial-network message passing neural network (S-MPNN) is proposed that is capable of both learning the spatial-network joint embedding and preserving geometric graph information. The contributions of this paper are summarized as follows:

- **A novel spatial network generative model and its learning objective are proposed**. The proposed model learning objective is derived from the perspective of the information bottleneck and are able to capture three semantic factors including that merely explaining spatial patterns, merely explaining network patterns, and the one that spans spatial-network-joint patterns.
- **A novel spatial network generative model inference algorithm is proposed with theoretical guarantees**. To optimize the information bottlenecks for spatial, network, and spatial-network-joint semantic factors, a model inference algorithm with a two-loop optimization strategy is proposed.
- **A new spatial network message passing neural network is proposed**. The proposed spatial message passing neural networks conduct the two/three-order message transmissions featured by angle and dihedral distances for 2D/3D spatial networks.
- **Comprehensive experiments were conducted**. Qualitative and quantitative experiments on two synthetic and two real-world datasets demonstrate that SND-VAE and its extensive

models are indeed capable of learning disentangled factors for different types of spatial networks.

2 RELATED WORKS

Deep Generative Models on Network Data. Graph generation involves learning the distributions of given graphs and generating more novel graphs. Most of the existing deep generative models for network/graph data are based on variational autoencoders(VAE) [16, 21, 41], generative adversarial nets (GANs) [19], and others [20]. For example, graphRNN builds an auto-regressive generative model on these sequences utilizing LSTM model [47]; while graphVAE [41] represents each graph in terms of its adjacent matrix and feature vector and utilizes the VAE model to learn the distribution of the graphs conditioned on a latent representation at the graph level. Graphite [16] encode the nodes of each graph into node-level embedding and predict the links between each pair of nodes to generate a graph. However, these existing graph generation methods do not consider the geometry space of the network during the generation process.

Deep Generative Models on Spatial Data. State-of-the-art deep learning methods have shown a remarkable capacity to model complex spatial data, including 3D objects [15, 32, 42, 48], and geospatial data [29]. Generative models of 3D objects exists in a variety of forms, including ordered [36] and unordered point clouds [4, 42], voxels [12], and manifolds [37, 40]. As deep graph convolution continues to develop, several groups have begun to extend the applications of graph neural network into the generation of 3D objects [43]. Most of these methods construct the nearest neighbor graphs from the 3D point clouds thus transforming the point cloud generation problem into a graph generation problem.

Spatial Graph Convolution Neural Networks. Graph neural networks (GNNs) are currently attracting considerable attention in multiple domains. Recently, to accommodate both graph constitution and graph geometry, there have been efforts to extend GNNs by incorporating 3D/2D node coordinates in graph convolutions [13, 25, 30, 44, 46]. One line of inquiry treats the spatial information of the nodes as node features and then conducts the spatial graph convolution via a conventional GNN [44, 46], which, however, are not invariant to graph rotation and translation. Another approach that has been proposed utilizes the mutual distances to store geometric information, with some being domain-specific. For example, Klicpera et al [30] proposed a 2D geometry graph convolution for molecular representations that used the directional information by transforming messages based on the angles between edges. Some works are generic [9, 11] only considering the adjacent nodes in describing the 3D/2D structure without considering the and angles dihedrals that feature the geometry of nodes.

Disentanglement Representation Learning. Disentangled representation learning has gained considerable attention, in particular in the field of image representation learning [2, 10, 17, 23, 26]. The goal here is to learn representations that separate out the underlying explanatory factors responsible for variations in the data. Such representations have been shown to be relatively resilient to the complex variants involved [6], and can be used to enhance generalizability as well as improve robustness against adversarial attack [2]. This has prompted a number of approaches that modify the VAE objective by adding, removing, or altering the weight of individual terms in the task of interpretable data generation [10, 26]. Disentanglement representation and generation on spatial data have been explored recently in the domain of point clouds [4, 42], mesh [15, 32] and manifolds [37, 40]. Meanwhile, the exploration of the interpretable representation learning of graphs, which expose the semantic factors of nodes and edges is also starting to bear fruit [14, 22, 34]. However, learning representations that disentangle the latent factors of a spatial network remains largely unexplored.

3 METHODOLOGY

In this section, the problem formulation is first provided before moving on to derive the overall objective from the perspective of the information bottleneck, following which a novel optimization algorithm to optimize the intractable proposed objective is proposed. Finally, the overall architecture as well as the novel spatial network message passing operations are introduced.

3.1 Problem Formulation

Define an input spatial network as $X = (S, G)$, where $S = (\mathcal{V}, L)$ represents the geometric information of the set of nodes $\mathcal{V}$. $L \in \mathcal{R}^{N\times 2}$ or $L \in \mathcal{R}^{N\times 3}$ denote to the 2D/3D geometric coordinates of nodes, respectively. N refers to the number of nodes. $G = (\mathcal{V}, \mathcal{E}, F, E)$ refers to a network [5], where $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges. $E \in \mathcal{R}^{N\times N}$ refers to the edge weights or adjacent matrix of the topology. $F \in \mathcal{R}^{N\times f}$ denote to the node feature and f is the length of each node feature vector. It is worth noting that the spatial information S cannot be simply represented as a node feature in the network since this form of representation can not capture the patterns that are invariant to rotation and translation of the network in the geometric space.

The goal of learning disentangled generative models for a spatial network is to learn the conditional distribution $p(S, G|Z)$ of the spatial network (S, G) given three groups of generative latent variables $Z = (z_s \in \mathbb{R}^{L_1}, z_g \in \mathbb{R}^{L_2}, z_{sg} \in \mathbb{R}^{L_3})$, where L_1, L_2, and L_3 are the number of variables in each group, in order to captures the three types of semantic factors. Specifically, z_s is required to capture just the independent spatial semantic factors; z_g is required to capture just the independent network factors; and z_{sg} is required to capture just the correlated spatial and network factors. Three challenges must be overcome to achieve this goal: (1) The lack of a co-decoder for the generation of a spatial network that is capable of jointly generating both the spatial and network data; (2) the difficulty of capturing the joint patterns of spatial and network, which exposes the correlated spatial and network semantic factors; and (3) the difficulty of enforcing the structured latent representations that separate the independent and dependent semantic factors of the spatial and network data.

3.2 The objective on spatial graphs generation

3.2.1 The derivation of the overall objective. As defined in the problem formulation, the goal here is to learn the conditional distribution of X given Z, namely, to maximize the marginal likelihood of the observed spatial network X in expectation over the distribution of the latent variable set (z_s, z_g, z_{sg}) as $\mathbb{E}_{p_\theta(Z)}(p_\theta(X|z_s, z_g, z_{sg}))$.

For a given observation of spatial network $X = (S, G)$, we describe the prior distribution of the latent representation as $p(z_s, z_g, z_{sg})$, which, however, is intractable to infer. We propose solving it based on variational inference, where the posterior needs to be approximated by another distribution $q_\phi(z_s, z_g, z_{sg}|G, S)$. So, the goal is also to minimize the Kullback–Leibler (KL) divergence between the true prior and the approximate posteriors. In order to encourage this disentangling property of $q_\phi(z_s, z_g, z_{sg}|G, S)$, we introduce a constraint by trying to match the inferred posterior configurations of the latent factors to the prior $p(z_s, z_g, z_{sg})$. This can be achieved if we set each prior to be an isotropic unit Gaussian, i.e.,$\mathcal{N}(\mathbf{0}, \mathbf{1})$, leading to the constrained optimization problem as:

$$\begin{aligned} \max_{\theta,\phi} \quad & \mathbb{E}_{S,G\sim D}[\mathbb{E}_{q_\phi(Z|S,G)} log p_\theta(G, S|z_s, z_g, z_{sg})] \\ \text{s.t.} \quad & \mathbb{E}_{S,G\sim D}[D_{KL}(q_\phi(z_s, z_g, z_{sg}|S, G)||p(z_s, z_g, z_{sg})] < I, \end{aligned} \tag{1}$$

where D refers to the observed dataset of the spatial networks.

First, we decompose the main objective term based on the assumption that $S \perp G|(z_s, z_g, z_{sg})$ and $S \perp z_g$ and $G \perp z_s$ (since z_s only captures information on S and z_g only captures information on G), where $\perp$ indicates an independent relationship. We then obtain:

$$\begin{aligned} & \mathbb{E}_{q_\phi(Z|S,G)}[\log p_\theta(G, S|z_s, z_g, z_{sg})] \\ & = \mathbb{E}_{q_\phi(Z|S,G)}[\log p_\theta(G|z_s, z_g, z_{sg}) + \log p_\theta(S|z_s, z_g, z_{sg})] \\ & = \mathbb{E}_{q_\phi(Z|S,G)}[\log p_\theta(G|z_g, z_{sg}) + \log p_\theta(S|z_s, z_{sg})]. \end{aligned} \tag{2}$$

Next, we decompose the constraint term based on the assumption that $p(z_s)$, $p(z_g)$, and $p(z_{sg})$ are independent given S and G as:

$$p_\phi(z_s, z_g, z_{sg}|S, G) = p_\phi(z_s|S)p_\phi(z_g|G)p_\phi(z_{sg}|S, G). \tag{3}$$

Then the objective is written as:

$$\begin{aligned} \max_{\theta,\phi} \quad & \mathbb{E}_{S,G\sim D}\mathbb{E}_{q_\phi(Z|S,G)}[\log p_\theta(G|z_g, z_{sg}) + \log p_\theta(S|z_s, z_{sg})] \\ \text{s.t.} \quad & \mathbb{E}_{S\sim D}[D_{KL}(q_\phi(z_s|S)||p(z_s)] < I_s. \\ & \mathbb{E}_{G\sim D}[D_{KL}(q_\phi(z_g|G)||p(z_g)] < I_g. \\ & \mathbb{E}_{S,G\sim D}[D_{KL}(q_\phi(z_{sg}|S, G)||p(z_{sg})] < I_{sg}, \end{aligned}$$

where we decompose I into three separate parts of the information capacity to control each group of latent variables, so that the variables inside each group of latent variables are disentangled.

As stated in the problem formulation, the latent z_s should capture just the independent spatial factors and z_{sg} should capture just the correlated spatial/graph factors. However, the above objective only ensures that z_{sg} captures all the correlated spatial/graph factors, and cannot enforce z_s captures all the independent spatial factors, which means that there is a chance that some of the independent spatial factors can also be captured by z_{sg}. Similarly, there is a chance that some of independent graph factors can also be captured by z_{sg}.

To address this issue, we first interpret the constraints based on the information bottleneck theory, as stated by Burgess et al.[8]. The posterior distribution $q_\phi(z_s|S)$ and $q_\phi(z_{sg}|S, G)$ are interpreted as an information bottleneck for the reconstruction task $\mathbb{E}_{q_\phi(Z|X)} log p_\theta(S|z_s, z_{sg})$. Similarly, $q_\phi(z_g|G)$ and $q_\phi(z_{sg}|S, G)$ are interpreted as the information bottleneck for the reconstruction task $\mathbb{E}_{q_\phi(Z|X)} log p_\theta(G|z_g, z_{sg})$. We then propose that, by constraining the information flow through z_{sg} to be less than the maximum information (entropy) C_{sg} of the correlated factors, namely $I_{sg} \leq C_{sg}$, the latent z_{sg} will only capture information on correlated factors when well-optimized. Thus, the final objective is expressed as:

$$\begin{aligned} \max_{\theta,\phi} \quad & \mathbb{E}_{S,G\sim D}\mathbb{E}_{q_\phi(Z|G)}[log p_\theta(G|z_g, z_{sg}) + log p_\theta(S|z_s, z_{sg})] \\ \text{s.t.} \quad & \mathbb{E}_{S\sim D}[D_{KL}(q_\phi(z_s|S)||p(z_s)] < I_s, \\ & \mathbb{E}_{G\sim D}[D_{KL}(q_\phi(z_g|G)||p(z_g)] < I_g, \\ & \mathbb{E}_{SG\sim D}[D_{KL}(q_\phi(z_{sg}|S, G)||p(z_{sg})] < I_{sg}, \\ & I_{sg} \leq C_{sg} \end{aligned} \tag{4}$$

The above objective is derived based on the concept that by constraining the information flow through z_{sg} to be less than the maximum information (entropy) of the correlated semantic factors, the latent z_{sg} will only capture information on correlated factors when well-optimized. Thus, the independent semantic factors of spatial and network will be forced into z_s and z_g, as the following theorem which is proved in Appendix A.

Theorem 1. *Given that (1) I_s and I_g are large enough to contain the information on the independent graph and spatial factors, and (2) $I_{sg} \leq C_{sg}$, hence to achieve the maximum objective, the information captured by z_{sg} needs to be all on the correlated semantic factors.*

Algorithm 1 Two-loop Optimization for SND-VAE

Input: The initialized parameter set $\mathcal{W}$; the initialized $I_{sg} = 0$ ($I_{sg} \notin \mathcal{W}$); the increase step γ for optimizing I_{sg}; the max value C_{max} as stop criterion; the number of epochs P of optimization for each updated I_{sg}.
Output: The optimized parameter set $\mathcal{W}$.

```
while I_sg ≤ C_max do
  for epoch = 1 : P do
    Compute the gradient of W via Back Propagation.
    Update W based on gradient with I_sg fixed.
  end for
  I_sg := I_sg + γ
end while
```

3.3 Optimization Strategy

To optimize the overall objective, we transform the inequality constraint into an tractable formulation. Given that I_s and I_g are constants, the first two constraints in Eq. 4 are rewritten based on the Lagrangian algorithm under KKT condition [35] as:

$$\mathcal{R}_1 = \beta_1 D_{KL}(q_\phi(z_s|S)||p(z_s) + \beta_2 D_{KL}(q_\phi(z_g|G)||p(z_g), \tag{5}$$

where the Lagrangian multipliers β_1 and β_2 is the regularization coefficients that constrains the capacity of the latent information channels z_s and z_g, respectively, and puts implicit independence pressure on the learned posterior.

In the third constraint, I_{sg} is a trainable parameter since the fourth constraint requires that $I_{sg} < C_{sg}$. Thus, it can be rewritten as a Lagrangian under the KKT condition as:

$$\mathcal{R}_2 = \beta_3(D_{KL}(q_\phi(z_{sg}|S, G)||p(z_{sg})) - I_{sg}). \tag{6}$$

Thus, the overall objective is formalized as:

$$\begin{aligned} & \max_{\theta,\phi} \mathbb{E}_{S,G\sim D}[\mathbb{E}_{q_\phi(Z|G)}[\log p_\theta(G|z_g, z_{sg}) + \log p_\theta(S|z_s, z_{sg})] - \mathcal{R}_1 - \mathcal{R}_2 \\ & \text{s.t.} \quad I_{sg} < C_{sg} \end{aligned}$$

Since C_{sg} is unknown, it is hard to optimize this objective. To deal with this, we introduce a novel optimization strategy that utilizes a

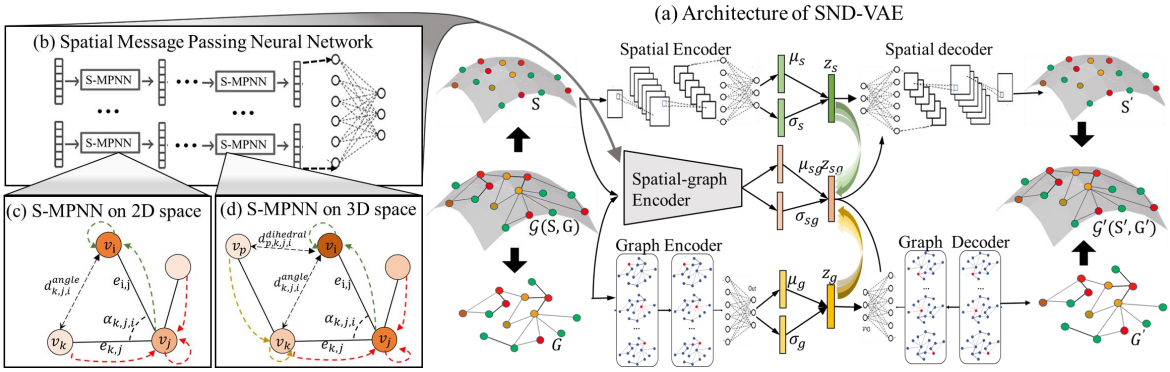

Figure 3: The proposed SND-VAE: (a) The overall architecture, which consists of a spatial encoder, a network encoder and a spatial network encoder, as well as a spatial decoder and a network decoder; (b) The S-MPNN; (c) the message passing operations for 2D S-MPNN in the spatial-network encoder; (d) the message passing operations for 3D S-MPNN in the spatial-network encoder.

two-loop optimization strategy: one loop is for the optimization of I_{sg}, and the other loop is for the optimization of the model, as shown in Algorithm 1. We propose to gradually increase I_{sg} by γ every P epoch, until it reaches C_{max}, where C_{max} is a hyper-parameter.

The most important advantage of the proposed model is that the optimization result is not sensitive to C_{max} for the following reasons: (1) if $C_{max} \leq C_{sg}$, we have $I_{sg} < C_{sg}$, where the constraint is satisfied; and (2) if $C_{max} > C_{sg}$, during the optimization process where $I_{sg} \leq C_{sg}$, all the correlated spatial and network information will flow into z_s and z_g, respectively, based on Theorem 1. During the optimization process where $I_{sg} > C_{sg}$, though the condition of Theorem 1 is no longer met, it is proved that further increasing the value of I_{sg} will not change the assignment of information on each latent representation, as defined in Theorem 2, which is proved in Appendix C.

Theorem 2. *During training process, if (1) z_s and z_g have captured the information of all the independent semantic factors of spatial and network respectively, and (2) z_{sg} captured all the correlated semantic factors of spatial and network, increasing the value of I_{sg} will not change the information assignment of independent semantic factors of spatial and network, and correlated semantic factors of spatial network to the z_s, z_g and z_{sg}.*

3.4 Spatial Network Encoders and Decoders

Based on the above inference for the objective, we are proposing our new Spatial-Network Disentangled VAE model (SND-VAE). In addition, to capture the correlated semantic factors within the spatial and network information, we propose a novel spatial network message passing neural network (S-MPNN) as one component in SND-VAE. Both will be described in detail in this section.

3.4.1 Architecture of SND-VAE. The architecture of the proposed model is shown in Fig. 3. The overall framework is based on a conventional VAE, where encoders learn the mean and standard deviation of the latent representation of the input and the decoder decodes the sampled latent representation vector to reconstruct the input. The proposed framework has three encoders, each of which models one of the distributions $q_\phi(z_s|S)$, $q_\phi(z_g|G)$, and $q_\phi(z_{sg}|S, G)$; and two novel decoders to model $p_\theta(G|z_g, z_{sg})$ and $p_\theta(S|z_s, z_{sg})$, that jointly generate the graph and spatial based on the three types of latent representations. Each type of representations is sampled using its own inferenced mean and standard derivation. For example, the representation vectors z_s are sampled as $z_s = {}_s + {}_s \cdot \epsilon$, where ϵ follows a standard normal distribution.

There are three encoders and two decoders in the overall architecture (shown in Fig. 3(a)). Specifically, for the spatial encoder, we utilize a convolution neural network. For the graph encoder, we utilize the typical graph convolution neural network [27]. For the spatial-network encoder, we propose a novel Spatial-Network Message Passing Neural Network (S-MPNN) (shown in Fig. 3(b)) which is detailed in the following. For the spatial decoder, we utilize the typical convolution neural network. For the graph decoder, we utilize a similar graph decoder to that proposed in NED-VAE [19]. The details of all the encoders and decoders are provided in Appendix D.

3.4.2 S-MPNN for 2D Graphs. In this section, we introduce the S-MPNN for the 2-D spatial network by first introducing a novel expression of geometry information of spatial network and then the two-order message passing layers of S-MPNN.

Expression of 2D geometry information. Normally, the geometry of 2-D graphs is specified in terms of the Cartesian coordinates of nodes, but doing so means that the specification depends on the (arbitrary) choice of origin and is thus too general for specifying a geometry that is invariant to both the rotation and translations in the graph. Thus, we propose to representing the spatial information by the distances between all pairs of nodes and the angles between all pair of edges. Specifically, we first define the edge distance as the distance between two nodes connected together and the angle as the angle formed between three nodes across two edges, as illustrated by $\alpha_{k,j,i}$ in Fig. 3(c). To adopts a unified scheme (distance) and reflects pairwise node interactions and their generally local nature, we introduce the angle distances to represent angles in the spatial network. The angle distance (e.g., $d_{k,j,i}$) is the distance between the end nodes of an angle (e.g., $\alpha_{k,j,i}$).

Two-order Message Passing. The key point for message passing operation is to define which nodes will influence and can pass messages to the target node. For each node in 2D graphs, its geometry information will be featured or determined not only by its first order neighbors, but also by its second order neighbors, as well as

the angle distance between its first and the relevant second order neighbors. For example, as shown in Fig. 3 (c), the connectivity and geometry of target node v_i can be described at least by its first-order neighboring node v_j, second-order neighboring nodes v_k as well as the angle distances $d^{angle}_{k,j,i}$. Thus, the message passing process at each layer for each node in 2D spatial network involves three steps: (1) the second-order nodes transmit the message into the first-order nodes carrying the angle distances information; (2) the first-order nodes collect all the received messages and transmit them to the target node; and (3) the embedding of target node is updated based on the messages. The detailed operations are shown as follows.

First, featured by its relevant angle distance, each second-order message (e.g., $m^{(l+1)}_{k,j,i}$) is flown from a second-order neighbor (e.g., node v_k) to its relevant first-order neighbor (e.g., node v_j) regarding the target node (e.g., v_i) at the $(l+1)$-th layer as:

$$m^{(l+1)}_{k,j,i} = M(h^l_i, h^l_j, h^l_k, , d^{edge}_{j,k}, d^{angle}_{k,j,i}), \tag{7}$$

where h^l_i refers to the latent embedding of node i at the l-th layer, $E_{i,j}$ refers to the edge weights (if applicable) of edge $e_{i,j}$. $d^{edge}_{k,j}$ refers to the distance between node v_k and v_j.

Next, based on the messages from all the second neighbors, the first-order message (e.g., $m^{(l+1)}_{i,j}$) is flown from the first-order neighbor (e.g., node v_j) to the target node (e.g., v_i) as:

$$o^{(l+1)}_{i,j} = O(h^l_i, h^l_j, d^{edge}_{i,j}, \sum\nolimits_{k\in\mathcal{N}(j)} m^{(l+1)}_{k,j,i}). \tag{8}$$

At last, after calculating the first-order messages passing onto the target node, the embedding of target node v_i is updated as:

$$h^{(l+1)}_i = U(h^l_i, \sum\nolimits_{j\in\mathcal{N}(i)} o^{(l+1)}_{i,j}). \tag{9}$$

The functions $M(\cdot)$, $O(\cdot)$ and $U(\cdot)$ can be implemented by the Multi-Later Perceptions (MLPs).

3.4.3 S-MPNN for 3D Graphs. Here we introduce the the S-MPNN for the 3-D spatial network by first introducing a novel expression of geometry information of 3D spatial network and then a three-order message passing layer.

Expression of 3D geometry information Compared to 2D spatial network, the geometry of a 3D graph is fully specified not only with edge and angles distances, but also dihedral distance. A dihedral is the angle between the plane formed by the target node v_i, its first-order neighbor v_j and its second-order neighbor v_k, and the plane formed by its first-order neighbor v_j, the second-order neighbor v_k and the third-order neighbor v_p. Thus, the dihedral distance is to represent dihedral in the spatial graph and denotes the distance between the target node v_i and its third-order neighbor v_p, as illustrated by $d^{dihedral}_{p,k,j,i}$ in Fig. 3 (d).

Three-order Message Passing. For each target node v_i in 3D spatial networks, its connectivity and geometry will be featured not only by its first and second order neighbors, but also the relevant angle and dihedral distances, as shown in Fig. 3 (d). Thus, the message passing process of 3D spatial network involves four steps.

First, featured by its relevant dihedral distances, the third-order message $m^{(l+1)}_{p,k,j,i}$ is flown from a third-order neighbor v_p to its associated second-order neighbor v_k regarding the target node v_i and first order node v_j at the $l+1$-th layer as:

$$t^{(l+1)}_{p,k,j,i} = T(h^l_i, h^l_j, h^l_k, h^l_p, d^{edge}_{k,p}, d^{dihedral}_{p,k,j,i}). \tag{10}$$

Next, given the messages from the third-order neighbors, featured by the angle distance, the second-order message $m^{(l+1)}_{k,j,i}$ is flown from the second-order neighbor to the first-order neighbor as:

$$m^{(l+1)}_{k,j,i} = M(h^l_i, h^l_j, h^l_k, d^{edge}_{j,k}, d^{angle}_{k,j,i}, \sum\nolimits_{k\in\mathcal{N}(k)} t^{l+1}_{p,k,j,i}), \tag{11}$$

Then, given the messages from the second-order neighbors, featured by the edge distance, the first-order message $o^{(l+1)}_{k,j,i}$ is flown from the second-order neighbor to the first-order neighbor as:

$$o^{(l+1)}_{i,j} = O(h^l_i, h^l_j, d^{edge}_{i,j}, \sum\nolimits_{k\in\mathcal{N}(j)} m^{l+1}_{k,j,i}). \tag{12}$$

At last, the node embedding of the target node v_i is updated as:

$$h^{l+1}_i = U(h^l_i, \sum\nolimits_{j\in\mathcal{N}(i)} o^{l+1}_{i,j}). \tag{13}$$

The functions $T(\cdot)$ can also be implemented by MLPs.

4 EXPERIMENT

This section reports the results of both qualitative and quantitative experiments that are carried out to test the performance of SND-VAE and its extensions on two synthetic and one real-world datasets. All experiments are conducted on a 64-bit machine with an NVIDIA GPU (GTX 1070, 1683 MHz, 16 GB GDDR5)[1].

4.1 Dataset

Waxman graphs. The Waxman random graph model places n nodes uniformly at random in a rectangular domain [45]. There are three types of factors. The independent graph factor b (controlling node attributes), the independent spatial factor p (controlling the overall node positions) and the graph-spatial correlated factor s (controlling both graph and spatial density). There are $80,000$ samples for training and $80,000$ for testing.

Random geometric graph. The random geometric graph model places n nodes uniformly at random in a rectangular domain [7]. There are three types of factors. The independent graph factor b (controlling node attributes), the independent spatial factor p (controlling the overall node positions) and the graph-spatial correlated factor s (controlling both graph and spatial density). There are $80,000$ samples for training and $80,000$ for testing.

Protein Structure dataset. Protein structures can be formulated as graph structured data where each amino acid is a node and the geo-spatial distances between them are edges. The density of graphs (contact maps) and the folding degree of protein (reflected by locations of amino acids) are correlated graph-spatial factors. There are $38,000$ samples for training and $38,000$ samples for testing.

4.2 Comparison Methods

The comparison methods can be divided into three categories as:

- To validate the significance of the proposed disentanglement objective and the optimization strategy, the proposed model is compared with (1) *beta-VAE* [23]; (2) *beta-TC-VAE* [10]; (3) *DIP-VAE* [31]; and (4) *NED-IPVAE* [22], where the overall architectures

[1]The code and details of datasets are available at: https://github.com/xguo7/SGD-VAE

Table 1: The evaluation results for the generated spatial graphs for different dataset (*kld_cls* refers to the KLD of graph clustering coefficient. kld_connect refers to for KLD of node connectivity. kld_dense refers to for KLD of graph density.

Dataset	Method	Node_MSE	Spatial_MSE	Edge_ACC	kld_cls	kld_dense	kld_connect	avgMI
Waxman graph	beta-VAE	0.22	3.01	66.83%	0.67	1.23	1.61	1.44
	beta-TCVAE	0.84	4.80	61.62%	0.40	1.12	1.56	1.85
	NED-IPVAE	2.28	1.80	66.73%	1.33	2.00	2.68	1.56
	SGD-VAE(geo-GCN)	6.12	31.20	64.59%	2.70	2.83	2.88	1.65
	SGD-VAE(pos-GCN)	6.84	34.80	64.61%	2.92	3.16	3.25	1.66
	SGD-VAE (single)	0.24	34.80	64.67%	**0.20**	**0.32**	**0.29**	N/A
	SGD-VAE	**0.12**	**0.18**	**67.40%**	0.39	0.50	0.53	**1.10**
Random Geometry graph	beta-VAE	6.84	34.80	71.29%	2.87	3.27	3.40	1.65
	beta-TCVAE	6.96	34.21	59.76%	1.55	2.31	2.37	1.73
	NED-IPVAE	1.44	1.80	76.65%	0.67	0.65	1.15	1.42
	SGD-VAE(geo-GCN)	6.80	31.20	71.32%	3.04	3.10	3.57	1.65
	SGD-VAE(pos-GCN)	6.75	33.21	71.27%	3.06	3.84	3.34	1.64
	SGD-VAE (single)	**0.36**	0.22	79.90%	**0.28**	**0.46**	**0.61**	N/A
	SGD-VAE	**0.36**	**0.19**	**80.80%**	0.79	1.48	1.85	**0.89**
Protein structure	beta-VAE	N/A	0.06	99.76%	2.09	2.91	3.69	0.93
	beta-TCVAE	N/A	0.78	91.40%	3.05	3.27	4.81	0.97
	NED-IPVAE	N/A	0.25	99.54%	2.31	2.36	4.01	0.91
	SGD-VAE(geo-GCN)	N/A	0.08	99.24%	1.78	2.37	3.49	1.02
	SGD-VAE(pos-GCN)	N/A	0.07	99.25%	1.97	1.86	3.26	1.01
	SGD-VAE (single)	N/A	**0.01**	99.63%	1.78	1.51	**2.39**	N/A
	SGD-VAE	N/A	0.06	**99.95%**	**1.58**	**1.46**	2.71	**0.77**

of comparison models are the same to the proposed SND-VAE, except for the disentanglement objective.

- To validate the superiority of proposed spatial message passing neural netwotk (S-MPNN), the proposed model is compared with two existing spatial graph convolution network: (1) geo-GCN [13] (2) pos-GCN [25] by replacing the spatial-network joint encoder with these two networks respectively.
- A baseline model (named as SND-VAE (single)), which has the same decoders to those of SND-VAE but with only one encoder (i.e. the proposed S-MPNN) is utilized to validate the necessity of structured latent representation for spatial network generation.

4.3 Evaluation on Spatial Network Generation

To evaluate the reconstruction performance of different generation models on both datasets, we calculate the MSE (mean squared error) between the generated and real node attributes or spatial locations, and calculate the accuracy of edge prediction. To evaluate the generation performance of the different models, we calculate the Kullback–Leibler divergence (KLD) between the generated and real spatial graphs to measure the similarity of their distributions in terms of: (1) density; (2) average clustering coefficient; and (3) the average node connectivity of networks.

4.3.1 Evaluation for Waxman graphs. The evaluation results of different models on generating Waxman graphs are shown in Table 1. The proposed SND-VAE outperforms the beta-VAE, beta-TCVAE and NED-VAE by about 65% in terms of reconstruction performance and about 47.6% in terms of generation performance. This validates the superiority of the proposed objective for disentangled structured latent representation as well as the effectiveness of the proposed optimization algorithm. The proposed SND-VAE outperforms the SND-VAE (geo-GCN) and SND-VAE (pos-GCN) by about 66.93% in terms of reconstruction performance and about 82.6% in terms of generation performance, showing the big advantage of the S-MPNN over the comparison spatial graph neural networks.

4.3.2 Evaluation results for Random Geometric graphs. The evaluation results of different models on generating Random Geometric graphs are shown in Table 1. The proposed SND-VAE outperforms the beta-VAE, beta-TCVAE and NED-VAE by about 68.97% in terms of reconstruction performance. This validate the superiority of the proposed objective for disentangled structured latent representation as well as the effectiveness of the proposed optimization algorithm. The proposed SND-VAE outperforms the SND-VAE (geo-GCN) and SND-VAE (pos-GCN) by about 94.7% in terms of reconstruction performance and about 3.9% in terms of generation performance. This validates the proposed S-MPNN is better at captuing the interaction patterns of spatial and networks over the comparison spatial graph convolution neural network.

4.3.3 Evaluation results for Protein structure generation. The evaluation results of different models on protein structure dataset are shown in Table 1. The proposed SND-VAE outperforms the beta-VAE, beta-TCVAE and NED-VAE by about 42.8% in terms of reconstruction performance and about 38.8% in terms of generation performance. This validate the superiority of the proposed objective for disentangled structured latent representation as well as the effectiveness of the proposed optimization algorithm. The proposed SND-VAE outperforms the SND-VAE (geo-GCN) and SND-VAE (pos-GCN) by about 10.5% in terms of reconstruction performance and about 22.1% in terms of generation performance. This validates the superiority of the proposed SGCN over the comparison spatial graph convolution neural network.

4.4 Evaluation on Disentangled Representations

We evaluate the proposed models and comparison models in the task of disentangled representation learning and provide both the quantitative evaluation and qualitative evaluation results.

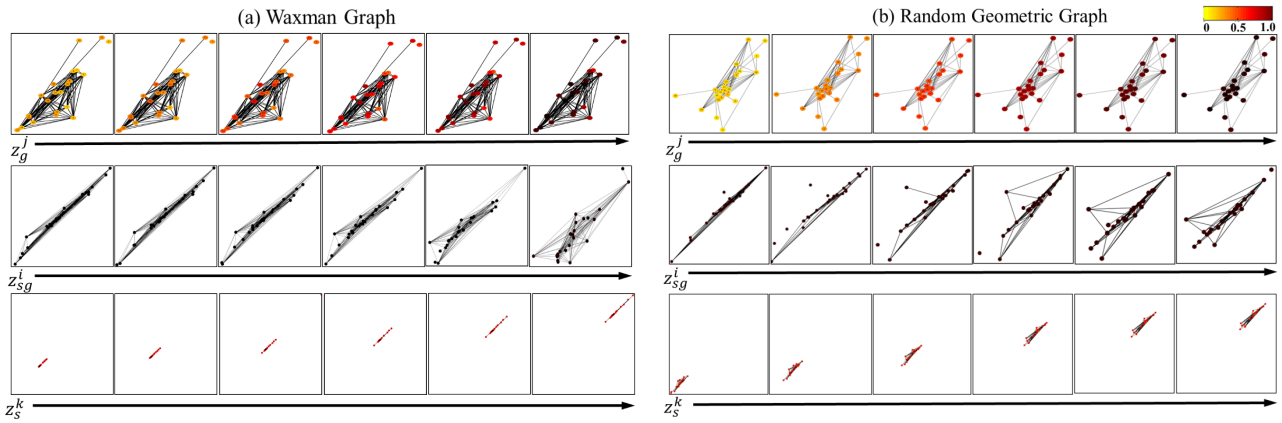

Figure 4: Visualizing the variations of generated spatial network regarding three groups of semantic factors on (1) Waxman graphs and (2) Random geometry graphs.

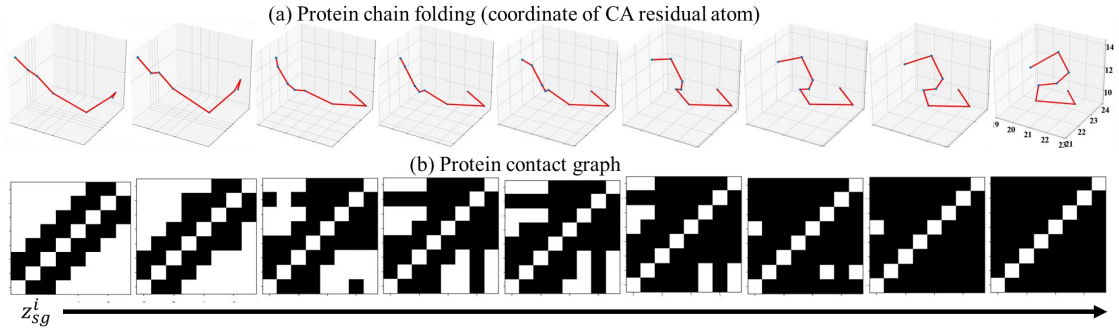

Figure 5: Visualizing the variations of generated protein structures in terms of the joint related semantic factors of (a) protein chain folding and (b) the density of contact graphs. The more black blanks in a contact graph, the higher density it has.

4.4.1 Quantitative Evaluation. As defined in the problem formulation, the main target of disentangled representation learning is to disentangle and capture the spatial-independent, network-independent and spatial-network correlated semantic factors by the structured latent representation z_s, z_g and z_{sg}. Thus, if the goal is fully satisfied, the standard mutual information matrix between three groups of semantic factors and three groups of latent representation will be a unit diagonal matrix (ground truth). Thus, we utilize avgMI [33] as metric which denotes to the distance between the real standard mutual information matrix and the ground truth matrix. The last column in Table 1 shows the avgMI evaluated on different models regarding different datasets that have the ground truth semantic factors. As shown in the results, the proposed SND-VAE achieves the best performance in disentangling the three groups of semantic factors into three pre-defined latent representation with the smallest avgMI. Specifically, the proposed SND-VAE have smaller avgMI than all the comparison methods by about 32.6%, 44.7%, and 20.5% on controlling the Waxman graphs, geometry graphs and protein structures, respectively. This is because the architecture of the proposed SND-VAE naturally enforce the disentangled z_s and z_g to capture the spatial and network semantic factors, respectively. Moreover, the proposed objective enforce the information of correlated spatial and network semantic factors flows into the latent representation z_{sg}.

4.4.2 Qualitative Evaluation. To measure the level of disentanglement achieved by different models, we search to qualitatively demonstrate that our proposed SND-VAE model consistently discover more latent factors and disentangles them in a cleaner fashion. As the same to the conventional qualitative evaluation in disentanglement representation learning [10, 23], by changing the value of one variable continuously while fixing the remaining variables, we can visualize the variation of the corresponding semantic factors in the generated spatial networks.

Fig. 4 shows the generated Waxman graphs and random geometry graphs when traversing the relevant latent variables in z_s, z_g and z_{sg}. The values of the latent variables range in $[-2, 2]$. The first line shows the variation of graph-related semantic factors (i.e., mean of node feature b), as reflected by the color of nodes. There are clear variation of the color of nodes in both generated waxman and random geomrty graphs. The second line shows the variation of the spatial-network joint related semantic factors. It can be easily observed that both the mutual distances between nodes and density of networks of the generated Waxman and random geometry graphs decrease when traversing one of latent variables in z_{sg}. To highlight the variation of the absolute locations of the whole spatial network, a larger coordinate system is utilized, as shown in the bottom line of Fig. 4. The overall location of the generated Waxmanx and random geometry graphs continously change from the left-bottom to the upper right corner. These qualitative evaluation results validate the effectiveness of the proposed SND-VAE in learning a structured latent representation, each of which has successful captured the relevant semantic factors of spatial network.

Fig. 5 shows the generated protein structure and contact maps when traversing the relevant latent in z_sg. The values of the latent variables range in $[-2, 2]$. As shown in Fig. 5, while traversing the values of latent variable, the folding degree of the protein structure increases and the density of the contact maps also increase accordingly. Thus, the proposed SND-VAE shows great capabilities in discovering the correlated semantic factors of spatial and network information in the protein structure data.

5 CONCLUSION

We have introduced SND-VAE, a novel and the first method for disentangling on spatial networks as far as we know. Moreover, we propose a generic framework and objectives to learn a structured latent representation, which explicitly disentangle the independent and correlated spatial and network semantic factors. The derived objective is analyzed from the perspective of information bottleneck and optimized by a novel optimization algorithm. Comprehensive experiments are conducted on the tasks of data generation and disentangled representation learning qualitatively and quantitatively. The comparison with five comparison models and a baseline model validates the effectiveness of the spatial network disentanglement architecture and the necessities of separately learning three types of latent representations.

ACKNOWLEDGMENTS

This work was supported by the National Science Foundation (NSF) Grant No. 1755850, No. 1841520, No. 2007716, No. 2007976, No. 1942594, No. 1907805, a Jeffress Memorial Trust Award, Amazon Research Award, NVIDIA GPU Grant, and Design Knowledge Company (subcontract number: 10827.002.120.04).

REFERENCES

[1] Farras Abdelnour, Michael Dayan, and Orrin et al. Devinsky. 2018. Functional brain connectivity is predictable from anatomic network's Laplacian eigenstructure. *NeuroImage* 172 (2018), 728–739.
[2] Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. 2016. Deep variational information bottleneck. *arXiv preprint arXiv:1612.00410* (2016).
[3] Namrata Anand and Po-Ssu Huang. 2018. Generative modeling for protein structures. In *NeurIPS*. 7505–7516.
[4] Tristan Aumentado-Armstrong, Stavros Tsogkas, Allan Jepson, and Sven Dickinson. 2019. Geometric disentanglement for generative latent shape models. In *ICCV*. 8181–8190.
[5] Marc Barthélemy. 2011. Spatial networks. *Physics Reports* 499, 1-3 (2011), 1–101.
[6] Yoshua Bengio, Aaron Courville, and Pascal Vincent. 2013. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence* 35, 8 (2013), 1798–1828.
[7] Milan Bradonjić, Aric Hagberg, and Allon G Percus. 2007. Giant component and connectivity in geographical threshold graphs. In *International Workshop on Algorithms and Models for the Web-Graph*. Springer, 209–216.
[8] Christopher P Burgess, Irina Higgins, and Arka et al. Pal. 2018. Understanding disentangling in β-VAE. *arXiv preprint arXiv:1804.03599* (2018).
[9] Daniel T Chang. 2020. Geometric Graph Representations and Geometric Graph Convolutions for Deep Learning on Three-Dimensional (3D) Graphs. *arXiv preprint arXiv:2006.01785* (2020).
[10] Ricky TQ Chen, Xuechen Li, Roger B Grosse, and David K Duvenaud. 2018. Isolating sources of disentanglement in variational autoencoders. In *NeurIPS*. 2610–2620.
[11] Hyeoncheol Cho and Insung S Choi. 2018. Three-dimensionally embedded graph convolutional network (3dgcn) for molecule interpretation. *arXiv preprint arXiv:1811.09794* (2018).
[12] Christopher B Choy, Danfei Xu, and JunYoung et al. Gwak. 2016. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *European conference on computer vision*. Springer, 628–644.
[13] Tomasz Danel, Przemysław Spurek, and Jacek et al Tabor. 2019. Spatial Graph Convolutional Networks. *arXiv preprint arXiv:1909.05310* (2019).
[14] Yuanqi Du, Xiaojie Guo, Amarda Shehu, and Liang Zhao. 2020. Interpretable Molecule Generation via Disentanglement Learning. In *Proceedings of the 11th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*. 1–8.
[15] Anastasia Dubrovina, Fei Xia, and Panos et al. Achlioptas. 2019. Composite shape modeling via latent space factorization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 8140–8149.
[16] Aditya Grover, Aaron Zweig, and Stefano Ermon. 2019. Graphite: Iterative Generative Modeling of Graphs. In *ICML*. 2434–2444.
[17] Xiaojie Guo, Yuanqi Du, and Liang Zhao. 2021. Property Controllable Variational Auto-encoder via Invertible Mutual Dependence. In *ICLR*.
[18] Xiaojie Guo, Sivani Tadepalli, Liang Zhao, and Amarda Shehu. 2020. Generating tertiary protein structures via an interpretative variational autoencoder. *arXiv preprint arXiv:2004.07119* (2020).
[19] Xiaojie Guo, Lingfei Wu, and Liang Zhao. 2018. Deep graph translation. *arXiv preprint arXiv:1805.09980* (2018).
[20] Xiaojie Guo and Liang Zhao. 2020. A systematic survey on deep generative models for graph generation. *arXiv preprint arXiv:2007.06686* (2020).
[21] Xiaojie Guo, Liang Zhao, and et al. 2019. Deep Multi-attributed Graph Translation with Node-Edge Co-evolution. In *ICDM*.
[22] Xiaojie Guo, Liang Zhao, and Zhao et al Qin. 2020. Interpretable Deep Graph Generation with Node-Edge Co-Disentanglement. In *KDD*.
[23] Irina Higgins, Loic Matthey, and Arka et al. Pal. 2017. beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework. *ICLR* 2, 5 (2017), 6.
[24] Dou Huang, Xuan Song, and Zipei et al. Fan. 2019. A variational autoencoder based generative model of urban human mobility. In *MIPR*. IEEE, 425–430.
[25] John Ingraham, Vikas Garg, and Regina et al Barzilay. 2019. Generative models for graph-based protein design. In *NeurIPS*. 15820–15831.
[26] Hyunjik Kim and Andriy Mnih. 2018. Disentangling by factorising. *ICML* (2018).
[27] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *ICLR* (2016).
[28] Jon Kleinberg. 2000. The small-world phenomenon: An algorithmic perspective. In *Proceedings of the thirty-second annual ACM symposium on Theory of computing*. 163–170.
[29] Konstantin Klemmer, Adriano Koshiyama, and Sebastian Flennerhag. 2019. Augmenting correlation structures in spatial data using deep generative models. *arXiv preprint arXiv:1905.09796* (2019).
[30] Johannes Klicpera, Janek Groß, and Stephan Günnemann. 2020. Directional message passing for molecular graphs. *arXiv preprint arXiv:2003.03123* (2020).
[31] Abhishek Kumar, Prasanna Sattigeri, and Avinash Balakrishnan. 2018. Variational Inference of Disentangled Latent Concepts from Unlabeled Observations. In *ICLR*.
[32] Jake Levinson, Avneesh Sud, and Ameesh Makadia. 2019. Latent feature disentanglement for 3D meshes. *arXiv preprint arXiv:1906.03281* (2019).
[33] Francesco Locatello, Stefan Bauer, and Mario et al. Lucic. 2019. Challenging Common Assumptions in the Unsupervised Learning of Disentangled Representations. In *ICML*. 4114–4124.
[34] Jianxin Ma, Peng Cui, and Kun et al. Kuang. 2019. Disentangled graph convolutional networks. In *ICML*. 4212–4221.
[35] Olvi L Mangasarian. 1994. *Nonlinear programming*. SIAM.
[36] Charlie Nash and Christopher KI Williams. 2017. The shape variational autoencoder: A deep generative model of part-segmented 3D objects. In *Computer Graphics Forum*, Vol. 36. Wiley Online Library, 1–12.
[37] Arun Pandey, Michael Fanuel, and Joachim et al. Schreurs. 2020. Disentangled Representation Learning and Generation with Manifold Optimization. *arXiv preprint arXiv:2006.07046* (2020).
[38] Mathew Penrose et al. 2003. *Random geometric graphs*. Vol. 5. Oxford university press.
[39] Taseef Rahman, Yuanqi Du, Liang Zhao, and Amarda Shehu. 2021. Generative Adversarial Learning of Protein Tertiary Structures. *Molecules* 26, 5 (2021), 1209.
[40] Scott Reed, Kihyuk Sohn, and Yuting et al Zhang. 2014. Learning to disentangle factors of variation with manifold interaction. In *ICML*. PMLR, 1431–1439.
[41] Martin Simonovsky and Nikos Komodakis. 2018. Graphvae: Towards generation of small graphs using variational autoencoders. In *ICANN*. Springer, 412–422.
[42] Qingyang Tan, Lin Gao, Yu-Kun Lai, and Shihong Xia. 2018. Variational autoencoders for deforming 3d mesh models. In *CVPR*. 5841–5850.
[43] Diego Valsesia, Giulia Fracastoro, and Enrico Magli. 2018. Learning localized generative models for 3d point clouds via graph convolution. In *ICLR*.
[44] Xiaoyang Wang, Yao Ma, and Yiqiet al. Wang. 2020. Traffic flow prediction via spatial temporal graph neural network. In *Proceedings of The Web Conference 2020*. 1082–1092.
[45] Bernard M Waxman. 1988. Routing of multipoint connections. *IEEE journal on selected areas in communications* 6, 9 (1988), 1617–1622.
[46] Sijie Yan, Yuanjun Xiong, and Dahua Lin. 2018. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *AAAI*, Vol. 32.
[47] Jiaxuan You, Rex Ying, and et al. 2018. GraphRNN: Generating Realistic Graphs with Deep Auto-regressive Models. In *ICML*. 5708–5717.
[48] Liang Zhao. 2021. Event Prediction in the Big Data Era: A Systematic Survey. *ACM Comput. Surv.* 54, 5, Article 94 (May 2021), 37 pages.

A PROOF OF THEOREM 1

Proof. To assist the proof, we introduce four groups of semantic factors. The spatial data is assumed to be simulated via two types of semantic factors as $S = \mathbf{Sim}(s^+, s^-)$ and the network data is also assumed to be simulated via two parts of semantic factors as $G = \mathbf{Sim}(g^+, g^-)$, which follows the conventional definition in the domain of disentangled representation learning [10, 23, 26]. Here $s^+ \perp s^-, g^+ \perp g^-, s^+ \perp g^+$, and $s^- \not\perp g^-$. That is, $s^+ \perp$ and $g^+ \perp$ refers to the independent semantic factors of spatial and network data, respectively. $s^- \perp$ and $g^- \perp$ refers to the correlated semantic factors of spatial and network data.

First, the objective can be rewrite based on the information bottleneck principle(see the derivation process in Appendix 2) as :

$$\begin{aligned} \max_{\theta,\phi} \quad & I(z_s, z_{sg}; S) + I(z_g, z_{sg}; G) \\ \text{s.t.} \quad & I(S; z_s) \le I_s, \\ \text{s.t.} \quad & I(G; z_g) \le I_g, \\ \text{s.t.} \quad & I(S; z_{sg}) + I(G; z_{sg}) \le I_{sg}, \end{aligned} \tag{14}$$

where $I(z_s, z_{sg}; S)$ refers to the mutual information between $p(z_s, z_{sg})$ and $p(S)$ and $I(z_g, z_{sg}; G)$ refers to the mutual information between $p(z_g, z_{sg})$ and $p(G)$. Considering $z_s \perp z_{sg}$ and $z_g \perp z_{sg}$, we have $I(z_s, z_{sg}; S) = I(z_s; S) + I(z_{sg}; S)$ and $I(z_g, z_{sg}; G) = I(z_g; G) + I(z_{sg}; G)$. Considering the graph G and spatial information S are generated based on four categories of semantic factors, namely independent spatial factors s^+, independent graph factors g^+, and the correlated spatial and graph factors g^- and s^-, we have $p(S) = p(s^+)p(s^-)$ and $p(G) = p(g^+)p(g^-)$. We also have $s^+ \perp g^+$. Thus, we can have

$$\begin{aligned} I(z_s, S) + I(z_{sg}, S) &= I(z_s, s^-) + I(z_s, s^+) + I(z_{sg}, s^+) + I(z_{sg}, s^-) \\ I(z_g, G) + I(z_{sg}, G) &= I(z_g, g^-) + I(z_g, g^+) + I(z_{sg}, g^+) + I(z_{sg}, g^-) \end{aligned}$$

Since $z_s \perp z_g$ and $s^- \not\perp g^-$, we have $I(z_s, s^-) = 0$ and $I(z_g, g^-) = 0$, thus, the objective can be rewritten as:

$$\begin{aligned} \max \quad & I(z_s, s^+) + I(z_{sg}, s^+) + I(z_{sg}, s^-) + I(z_g, g^+) + I(z_{sg}, g^+) + I(z_{sg}, g^-) \\ \text{s.t.} \quad & I(s^+; z_s) \le I_s \\ \text{s.t.} \quad & I(S; z_{sg}) + I(G; z_{sg}) \le I_{sg} \\ \text{s.t.} \quad & I(g^+; z_g) \le I_g \end{aligned} \tag{15}$$

since $z_s \perp z_{sg}$ and $z_g \perp s_{sg}$, the information of s^+ captured by z_s has no intersection with the information of s^+ captured by z_{sg}, thus we can have:

$$\begin{aligned} I(z_s; s^+) + I(z_{sg}; s^+) \le I(s^+; s^+) = H(s^+) \\ I(z_g; g^+) + I(z_{sg}; g^+) \le I(g^+; g^+) = H(g^+). \end{aligned} \tag{16}$$

Based on the second constrain and $s^+ \perp s^-$ and $g^+ \perp g^-$, we also have

$$I(z_{sg}; s^+) + I(z_{sg}; s^-) + I(z_{sg}; g^+) + I(z_{sg}; g^-) \le C_{sg}. \tag{17}$$

By summarizing the inequalities Eq 16 and Eq. 17, we have:

$$\begin{aligned} I(z_s; s^+) + 2 * I(z_{sg}; s^+) + I(z_{sg}; s^-) + \\ + I(z_s; g^+) + 2 * I(z_{sg}; g^+) + I(z_{sg}; g^-) \\ \le H(s^+) + C_{sg} + H(g^+), \end{aligned} \tag{18}$$

which can be further written as:

$$\begin{aligned} I(z_s; s^+) + I(z_{sg}; s^+) + I(z_{sg}; s^-) + \\ + I(z_s; g^+) + I(z_{sg}; g^+) + I(z_{sg}; g^-) \\ \le H(s^+) + C_{sg} + H(g^+) - I(z_{sg}; s^+) - I(z_{sg}; g^+). \end{aligned} \tag{19}$$

Since $I(z_{sg}; s^+) \ge 0$ and $I(z_{sg}; g^+) \ge 0$, and $H(s^+)$ and $H(g^+)$ are constants, the left side of Inequality (13) achieves its maximum when $I(z_{sg}; s^+) = 0$ and $I(z_{sg}; g^+) = 0$. Thus, to achieve the most optimal objective, z_{sg} need to only capture the information from correlated semantic factors s^+ and g^+. □

B THE ASSISTANT DERIVATION OF THEOREM 1

In this section, we derive the process how the initial objective for spatial graph generation (as shown in Eq. (4)) can be written as the information bottleneck format (as shown in Eq. (15)).

Specifically, for the first part of Eq. (4), we have:

$$\begin{aligned} & \mathbb{E}_{S,G\sim D}\mathbb{E}_{q(Z|S,G)}[\log p(G|z_g, z_{sg}) + \log p(S|z_s, z_{sg})] \\ &= \mathbb{E}_{p(Z,S,G)}[\log p(G|z_g, z_{sg}) + \log p(S|z_s, z_{sg})] \\ &= \mathbb{E}_{p(z_g,z_{sg},G)}[\log \frac{p(G|z_g, z_{sg})}{p(G)}] + \mathbb{E}_{p(G)} \log p(G)] \\ &+ \mathbb{E}_{p(z_g,z_{sg},S)} \log \frac{p(S|z_s, z_{sg})}{p(S)}] + \mathbb{E}_{p(S)} \log p(S)] \\ &= I(z_s, z_{sg}; S) + I(z_g, z_{sg}; G) \end{aligned} \tag{20}$$

For the first constraint, we have:

$$\begin{aligned} & \mathbb{E}_{S\sim D}[D_{KL}(q(z_s|S)||p(z_s)] \\ &= \mathbb{E}_{S\sim D}\mathbb{E}_{q(z_s|S)}[\log \frac{q(z_s|S)}{p(z_s)}] \\ &= \mathbb{E}_{S\sim D}\mathbb{E}_{q(z_s|S)}[\log \frac{q(z_s|S)}{q(z_s)} \frac{q(z_s)}{p(z_s)}] \\ &= \mathbb{E}_{S\sim D}\mathbb{E}_{q(z_s|S)}[\log \frac{q(z_s|S)}{q(z_s)} + \log \frac{q(z_s)}{p(z_s)}] \\ &= \mathbb{E}_{S\sim D}[D_{KL}(q(z_s|S)||q(z_s)] + \mathbb{E}_{S\sim D}\mathbb{E}_{q(z_s|S)}[\log \frac{q(z_s)}{p(z_s)}] \\ &= I(z_s|S) + \mathbb{E}_{q(z_s)}[\log \frac{q(z_s)}{p(z_s)}] \\ &= I(z_s|S) + D_{KL}(q(z_s)||p(z_s) \end{aligned} \tag{21}$$

Considering $D_{KL}(p(z_s)||q(z_s)$ is a constant that has nothing to do with the parameters θ and ϕ, thus the first constrain is rewritten as: $I(S; z_s) \le I_s^I$. Since I_s^I is a hyper-parameter which is select before training, for simplicity, we still use I_s as the right side of the constraint. We can have the same derivation for the second constraints in Eq.5.

Table 2: Encoders and decoders architectures (Each layers is expressed in the format as *<filter_size><layer type><Num_channel><Activation function><stride size>*. *FC* refers to the fully connected layers). *c-deconv* and *c-conv* refers to the cross edge deconvolution and convolution respectively. The activation functions after each layer are all ReLU except the last layers.

Spatial Encoder	Joint Encoder	Network encoder	Network decoder(for edge)	Network decoder(for node)	Spatial Decoder
Input: $L \in \mathbb{R}^{25\times 2}$	Input: E, L	Input: $E \in \mathbb{R}^{25\times 25}$,$F \in \mathbb{R}^{25}$	Input:$z_g \in \mathbb{R}^{100}$,$z_{sg} \in \mathbb{R}^{200}$	Input:$z_g \in \mathbb{R}^{100}$,$z_{sg} \in \mathbb{R}^{200}$	Input:$z_s \in \mathbb{R}^{100}$,$z_{sg} \in \mathbb{R}^{200}$
5 conv1D.10. stride 1	S-MPNN.20	GCN.10	FC.500	FC.500	FC.500
5 conv1D.10. stride 1	S-MPNN.50	GCN.20	5×5 deconv.50. stride 1	5 conv1D.50. stride 1	5 conv1D.50. stride 1
5 conv1D.20. stride 1	FC.200.	FC.100.	5×5 deconv.20. stride 1	5 conv1D.20. stride 1	5 conv1D.20. stride 1
FC.100.	FC.200	FC.100	FC.1	FC.1	5 conv1D.10. stride 1
FC.100					FC.2

Next, we consider the third constraint in Eq. (4). Given $S \perp G|z_{sg}$, We can have:

$$
\begin{aligned}
&\mathbb{E}_{S,G\sim D}[D_{KL}(q(z_{sg}|S,G)||p(z_{sg})] \qquad (22)\\
&= \mathbb{E}_{S,G\sim D}\mathbb{E}_{q(z_{sg}|S,G)} \log \frac{q(z_{sg}|S,G)q(S,G)q(z_{sg})}{p(z_{sg})q(S,G)q(z_{sg})}\\
&= \mathbb{E}_{S,G\sim D}\mathbb{E}_{q(z_{sg}|S,G)} \log \frac{q(S,G|z_{sg})}{q(S,G)} + \mathbb{E}_{q(z_{sg})} \log \frac{q(z_{sg})}{p(z_{sg})}\\
&= \mathbb{E}_{S,G\sim D}\mathbb{E}_{q(z_{sg}|S,G)} \log \frac{q(S,G|z_{sg})}{q(S)q(G)} - \mathbb{E}_{S,G\sim D} \log \frac{q(S,G)}{q(S)q(G)}\\
&+ \mathbb{E}_{q(z_{sg})} \log \frac{q(z_{sg})}{p(z_{sg})}\\
&= \mathbb{E}_{S,G\sim D}\mathbb{E}_{q(z_{sg}|S,G)} \log \frac{q(S|z_{sg})q(G|z_{sg})}{q(S)q(G)} - I(S;G)\\
&- D_{KL}(p(z_{sg})||q(z_{sg}))\\
&= I(S,z_{sg}) + I(G;z_{sg}) - I(S;G) - D_{KL}(p(z_{sg})||q(z_{sg}))
\end{aligned}
$$

Considering $D_{KL}(p(z_s)||q(z_s)$ and $I(S;G)$ is a constant that has nothing to do with the parameters θ and ϕ, thus the third constrain is rewritten as: $I(S;z_{sg}) + I(G;z_{sg}) \leq I^I_{sg}$.

C PROOF OF THEOREM 2

Proof. To assist the proof, we introduce four groups of semantic factors. The spatial data is assumed to be simulated via two types of semantic factors as $S = \mathbf{Sim}(s^+, s^-)$ and the network data is also assumed to be simulated via two parts of semantic factors as $G = \mathbf{Sim}(g^+, g^-)$, which follows the conventional definition in the domain of disentangled representation learning [10, 23, 26]. Here $s^+ \perp s^-, g^+ \perp g^-, s^+ \perp g^+$, and $s^- \not\perp g^-$. That is, $s^+ \perp$ and $g^+ \perp$ refers to the independent semantic factors of spatial and network data, respectively. $s^- \perp$ and $g^- \perp$ refers to the correlated semantic factors of spatial and network data.

(1) Given the situation that z_s and z_g have already captured all the independent semantic factors s^+ and g^+, we have $I(z_s, s^+) = I(s^+, s^+)$ and $I(z_g, g^+) = I(g^+, g^+)$. Given the situation that z_{sg} captured all the dependent semantic factors s^- and g^-, we can have $I(z_{sg}, s^+) = 0$ and $I(z_{sg}, g^+) = 0$. We also can have $I(z_{sg}, s^-) = I(s^-, s^-)$ and $I(z_{sg}, g^-) = I(g^-, g^-)$. Thus, the value of the current loss is equal to: $I(s^+, s^+) + I(s^-, s^-) + I(g^+, g^+) + I(g^-, g^-)$.

(2) Next, we come back to the original objective function, which is expressed as:

$$
\begin{aligned}
\max I(z_s, s^+) + I(z_{sg}, s^+) + I(z_{sg}, s^-) + I(z_g, g^+)\\
+I(z_{sg}, g^+) + I(z_{sg}, g^-). \qquad (23)
\end{aligned}
$$

Since $z_s \perp z_{sg}$ and $z_g \perp z_{sg}$, we have:

$$I(z_s, s^+) + I(z_{sg}, s^+) \leq I(s^+, s^+) \qquad (24)$$

$$I(z_g, g^+) + I(z_{sg}, g^+) \leq I(g^+, g^+). \qquad (25)$$

We also have $I(z_{sg}, s^-) \leq I(s^-, s^-)$ and $I(z_{sg}, g^-) \leq I(g^-, g^-)$. Thus, the value of the most optimal loss is $I(s^+, s^+) + I(s^-, s^-) + I(g^+, g^+) + I(g^-, g^-)$.

As a summary, given the situation that z_s and z_g have already captured all the independent semantic factors s^+ and g^+ and z_{sg} captured all the dependent semantic factors s^- and g^-, the loss has already achieved the optimal one. As the I_{sg} increases, though the constraint is removed, the loss can not be maximized anymore no matter how the assignment of information change through z_s, z_g and z_{sg}. Thus, training while increasing C_{sg} will not change the current status of information flow. □

D ARCHITECTURE AND HYPER-PARAMETERS

The detailed setting of the encoders and decoders in the model for the experiment are provided in Table 2.

The network decoder have two parts: one is for nodes and one is for edges, which are detailed as follows. The nodes feature/labels are generated by a set of conventional 1D convolution layers. The edge weights/adjacent matrix are generated based on a set of edge deconvolution layers and fully connected layers. The input is the concatenation of both the network representation z_g and the spatial network representation z_{sg}. First, the input vector is mapped into a node-level feature vector through a fully connected layer and is converted into a matrix by being replicated. The same node assignment vector S is also concatenated to this feature matrix. The hidden edge latent representation matrices are then generated by the node-to-edge deconvolution layer [19] by decoding each of the node-level representations, where the principle is that each node's representation can make contributions to the generation of its related edges latent representation. Thirdly, the edge weights oradjacent matrix E is generated through the edge-edge deconvolution layer, where the principle is that each hidden edge feature can contribute to the generation of its adjacent edges.