STGEN: Deep Continuous-time Spatiotemporal Graph Generation

Chen Ling[1], Hengning Cao[2], and Liang Zhao[1]✉

[1] Emory University, Atlanta, GA, USA
[2] Cornell University, Ithaca, NY, USA
[1]{chen.ling, liang.zhao}@emory.edu,[2]hc2225@cornell.edu

Abstract. Spatiotemporal graph generation has realistic social significance since it unscrambles the underlying distribution of spatiotemporal graphs from another perspective and fuels substantial spatiotemporal data mining tasks. Generative models for temporal and spatial networks respectively cannot be easily generalized to spatiotemporal graph generation due to their incapability of capturing: 1) mutually influenced graph and spatiotemporal distribution, 2) spatiotemporal-validity constraints, and 3) characteristics of multi-modal spatiotemporal properties. To this end, we propose a generic and end-to-end framework for spatiotemporal graph generation (STGEN) that jointly captures the graph, temporal, and spatial distributions of spatiotemporal graphs. Particularly, STGEN learns the multi-modal distribution of spatiotemporal graphs via learning the distribution of spatiotemporal walks based on a new heterogeneous probabilistic sequential model. Auxiliary activation layers are proposed to retain the spatiotemporal validity of the generated graphs. In addition, a new boosted strategy for the ensemble of discriminators is proposed to distinguish the generated and real spatiotemporal walks from multi-dimensions and capture the combinatorial patterns among them. Finally, extensive experiments are conducted on both synthetic/real-world spatiotemporal graphs and demonstrated the efficacy of the proposed model.

Keywords: Deep Graph Generation · Spatiotemporal Graph · Deep Generative Model.

1 Introduction

Many complex systems can be modeled as graphs, which characterize the objects (i.e., nodes) and their interactions (i.e., edges) [31]. In many graph systems, the nodes and edges need to be embedded in space and evolve over time. The former is denoted as spatial network [4] while the latter is named temporal graphs [14], both of which are well-explored domains by network science models such as spatial small worlds model [29], optimal network [1], and epidemic temporal network [20]. These conventional methods propose to utilize prescribed structural assumptions (e.g., temporal exponentiality, network shortest distance) to characterize spatiotemporal graphs (STGs) through synthesizing them. However,

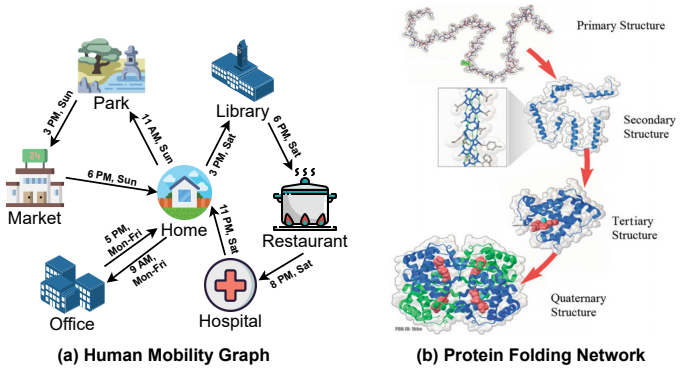


Fig. 1: The example of real world STGs. a) the human mobility graph describes one week's living trajectory between different locations of an adult with timestamps on directed edges. b) The protein folding process [21] with amino acids in different folding phases, which is a spatial graph evolves with time.

traditional methods are limited in modeling and interpreting STGs since the intrinsically complex spatiotemporal patterns are hard to be modeled only by prior knowledge. Such prior knowledge is not always available especially considering the limited information of human beings on many real-world complex networks such as brain network dynamics [23], the folding of protein structure [21], and catastrophic failures in power grids [27]. Therefore, it is desired to have a model with high expressiveness in learning the dynamics directly from data without detailed handcrafted rules.

Recently, there has been a surge of research efforts on deep generative models in the task of graph generation. For example, enormous works [7, 32, 25, 34, 35] have achieved promising performance in generating realistic static or temporal graphs or separately considering the spatial properties. On the other hand, there is also a fast-growing research body on discriminative learning for STG data and their applications, such as traffic prediction [33], emotion perception [5], and object recognition [22]. However, joint generative consideration of spatial, temporal, and graph aspects is still under-explored.

To fill this gap, we focus on the generic problem of STG generation, which cannot be easily handled by combining existing works because *1) Difficulties in jointly learning both graph and spatiotemporal distribution of STGs.* As shown in Fig. 1(a), the human mobility behavior follows the distribution characterized jointly by the spatial, temporal, and graph patterns. More important, these three patterns are strongly correlated. For example, the time "9 AM on workday" may correlate to the edge "going to work" and the location "traffic from home to downtown". Existing static and dynamic graph generative models cannot be combined to model it, which will discard the synergies among all the patterns simultaneously [10, 8]. *2) Difficulties in ensuring spatiotemporal validity in the generated STGs.* STGs need to follow spatial and temporal constraints. The

former means that the locations of the nodes and edges are confined in a specific geometric topology, while the latter means that the nodes and edges need to respect their temporal order. For example, in Fig. 1(a), for a path in the human mobility graph, the timestamp value of the first edge must be smaller than that for the second. In Fig. 1(b), a pair of amino acids with direct connections must be close to each other in space. Spatial and temporal constraints directly determine the validity of STGs. And because they are hard constraints, they cannot be intrinsically merged into the distribution of STGs, which are typically continuous according to the common statistical models. Therefore, establishing a model that can generate STGs while maintaining their validity is imperative yet challenging. *3) Difficulties in identifying the dependencies and independencies of spatial, temporal, and graph patterns.* A STG is composed of multi-modal components: graph information, temporal information, and spatial information. Some of the information are correlated while some are independent, which forms combinatorial among them into STG patterns, such as spatial and temporal graph patterns. Taking Fig. 1(a) and (b) as examples: in human mobility STG, spatial, temporal, and graph patterns are strongly correlated. In the folding process of protein (spatial graph of amino acids), the correlation between spatial and network patterns is even stronger than that between temporal and some graph properties (e.g., edge connections).

To address all the challenges, we propose a novel continuous-time STG generation framework, called *STGEN*, which coherently models both graph topology and spatiotemporal dynamics of observed STGs through a new generative adversarial model. Specifically, we propose to decompose STGs into spatiotemporal walks by developing a novel spatiotemporal walk generator to jointly capture the graph and spatiotemporal distribution. Novel STGs can be assembled through conditionally concatenating generated spatiotemporal walks with the guarantee of spatiotemporal validity. On top of that, we design a new discriminator which is an ensemble of multi-modal sub discriminators with different combinations of spatial, temporal, and graph patterns. We summarize contributions as follows:

- **The development of a new generative framework for STG generation**. We formally define the problem of STG generation and propose STGEN to tackle its unique challenges arising from real applications. It generates STGs with ensuring graph, temporal, and spatial validity.
- **The design of a novel spatiotemporal walk generator**. We develop a novel spatiotemporal walk generator with spatiotemporal information decoders to capture the underlying dynamics of observed STGs. Auxiliary activation layers are leveraged to ensure the validity of the generated STG.
- **The proposal of the ensemble of multi-modal sub discriminators for stronger STG adversarial training.** Multiple sub discriminators are designed and synergized by adaptively boosting extension in order to coherently examine the different combinations of spatial, temporal, and graph patterns. To extend WGAN by our ensemble discriminator, we propose a well-modularized learning objective and optimization algorithm.

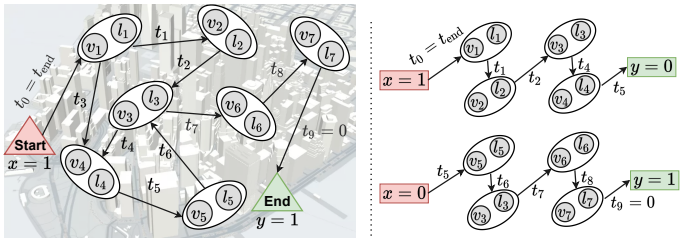
– **The conduct of extensive experiments to validate the effectiveness of the proposed model.** Extensive experiments and case analysis on both synthetic and real-world datasets demonstrate the capability of STGEN in generating the most realistic STGs compared to other baselines in terms of both temporal and spatial similarities.

2 Related Works

Spatial and Temporal Graphs. The graph is a mathematical subject that describes the interactions (edges) between a set of objects (nodes). Such a graph structure can also be employed as the infrastructure of real-world dynamic systems. However, for many dynamic systems, one might have more information than just about who interacts. Transportation and mobility networks [3], internet of things [18], and social networks [24], are all examples where time and space information are significant and where graph topology alone does not contain all the information. Nowadays, both spatial and temporal graphs have become indispensable extensions of static graphs and achieved success in various dynamic system modeling tasks [30, 33, 28]. We refer readers to recent surveys [14, 4] for more details of both spatial and temporal graphs.

Deep Graph Generation. A number of deep generative models for static graphs have emerged in the past few years [12]. Specifically, GraphVAE [25] try to generate new graphs' adjacency matrix in a variational auto-encoder way, while [32] treats graph generation as a node/edge sequence generation process based on LSTM. NetGAN and its variant [7, 19] generates new graphs through modeling the random walk distribution in observed graphs. Other than the static graph generation, the other line of works [34, 35, 11] have achieved success in generating temporal or spatial graphs. Both lines of works leverage temporal walks and spatial attribute to model the temporal and spatial properties of the observed graphs, respectively. However, none of the above works can be directly adapted to generate STGs since they neither can effectively decode the joint distribution of topology and spatiotemporal properties in STGs nor ensure the spatiotemporal validity (i.e., temporal ordering, semantics of spatial coordinates, and physical constraints of spatial properties) of the generated graph [14, 4]. Recently, STGD-VAE [9] is proposed to model deep generative processes of composing discrete spatiotemporal networks, which is a specific type of general spatiotemporal graphs that reduces time information into ordinal values. Hence it does not directly generate continuous-time spatiotemporal graphs, spatiotemporal validity constraints, and various spatiotemporal distributions. In this paper, we propose a generic framework to jointly model the distribution of multi-modal properties of observed STGs and generate novel ones with the spatiotemporal validity guarantee. To the best of our knowledge, STGEN is the first-of-its-kind deep generative model designed for continuous STGs with validity constraints.

Spatiotemporal Deep Learning. With the prevalence of deep learning techniques, various models [33, 16] have been proposed to model spatiotemporal data through decoding its underlying distribution in a discriminative way with-



(a) A Time-inversed Spatiotemporal Walk (b) Segmented Spatiotemporal Walks

Fig. 2: Fig. 2(a) illustrates a spatiotemporal walk, and Fig. 2(b) indicates the decomposition process of a spatiotemporal walk to multiple smaller-sized segmented spatiotemporal walk. For the sake of simplicity, we omit the turning angle ϕ_i and velocity ξ_i attached on each edge e_i in above figures.

out applying any hand-crafted rules. These approaches generally achieve promising results in plenty of predictive tasks through analyzing specific patterns (e.g., spatial proximity [16] and temporal correlations [33, 5]) of the spatiotemporal data. However, interpreting spatiotemporal data from the generation perspective has received less attention since it is a more challenging task and requires one to comprehensively capture the underlying dynamics among multi-modal spatiotemporal properties and their intricate and entangled dependencies. A few tries of utilizing deep generative models have been observed in spatiotemporal data generation. Additionally, [26] converts spatiotemporal data (e.g., trajectory) to images and applies GAN for the generation. Another work SVAE [15] utilizes VAE modules to learn variables from Gaussian distribution and generate novel human mobility accordingly. However, existing works still cannot explicitly consider the spatiotemporal validity during the generation. STGEN is the first-of-its-kind generative model that generates spatiotemporal data in the form of graphs with spatiotemporal validity.

3 Problem Setting

A continuous-time spatiotemporal graph (STG) is a directed graph $G = \{e_1, ..., e_i, ...\}$, where each edge $e_i = \left(u_i, v_i, t_i, l_u^{(i)}, l_v^{(i)}\right)$. $u_i, v_i \in V$ are two end nodes of e_i, $t_i \in [0, t_{\text{end}}]$ is the timestamp on the edge, and $[0, t_{\text{end}}]$ is the time span of the STG with $t_{\text{end}} \in \mathbb{R}^+$. Each node v_i in a STG is associated with a spatial attribute $l_v^{(i)}$. The spatial attribute $l_v^{(i)} = (l_{\text{lat}}^{v_i}, l_{\text{lon}}^{v_i})$ can be interpreted as a Global Positioning System (GPS) location with specific latitude and longitude on earth.

Definition 1 (*Spatiotemporal Walk*) *A spatiotemporal walk* $\mathbf{s} = \{t_0, e_1, ..., e_{L_\mathbf{s}}, t_{L_\mathbf{s}}\}$ *is defined as a sequence of spatiotemporal edges and a pair of initial time budget* t_0 *and end time budget* $t_{L_\mathbf{s}}$*, where* $L_\mathbf{s}$ *is the length of the spatiotemporal walk* $\mathbf{s}$ *and* $\forall e_i \in G$*. Specifically, a spatiotemporal edge is defined as:* $e_i =$

$\left(u_i, v_i, t_i, l_u^{(i)}, l_v^{(i)}\right) \in E$, *where each* $t_i \in [0, t_{end}], t_i < t_{i-1}$ *is called the* "time budget" *for* e_i *indicating the total timespan consumed in* e_i. *Intuitively, the initial time budget* $t_0 = t_{end}$ *and end time budget* $t_{L_\mathbf{s}} = 0$. *An example of the spatiotemporal walk is illustrated in Fig. 2(a).*

A continuous-time STG can be denoted as the union of all the spatiotemporal walks $G = \bigcup_{\mathbf{s} \sim P_r(\mathbf{s})} \mathbf{s}$, where the $P_r(\mathbf{s})$ is the distribution of all walks in graph G. It is straightforward that we can leverage fixed-length sequential models to learn the distribution of the spatiotemporal walks in order to capture the overall STG distribution. However, the nature of continuous-time STGs decides one cannot arbitrarily sample random walks as was done in static graphs [7] but needs to follow certain spatiotemporal orders. More specifically, the length of a spatiotemporal walk in STG is regulated by the starting and ending point based on the spatiotemporal information on edges. The length of the spatiotemporal walk varies that may easily reach a million-scale (especially when the time granularity is small), which cannot be learned effectively and efficiently by sequence learning methods. Therefore, the definition of spatiotemporal walk needs to be enriched into the following extension.

Definition 2 (*Segmented Spatiotemporal Walk*) *A segmented spatiotemporal walk is defined as a sequence* $\bar{\mathbf{s}} = \{(x, y, t_0), e_1, e_2, ..., e_L\}$ *with its profile informatiom* (x, y, t_0), *which is segmented from an originated spatiotemporal walk. Specifically, the* $L \leq L_\mathbf{s}$ *is the length of each segmented spatiotemporal walk. The profile information* (x, y, t_0) *includes* $x \in \{0, 1\}$ *and* $y \in \{0, 1\}$, *which denote whether* $\mathbf{s}$ *is the respective starting or ending segmentation (*$x = 1, y = 0$ *or* $x = 0, y = 1$*) or neither of them (*$x, y = 0$*) in its originated spatiotemporal walk. The whole process of decomposing a spatiotemporal walk to segmented spatiotemporal walks is elucidated in Fig. 2(b).*

With all the aforementioned notions, we can formalize the problem of STG generation as follows:

*Problem 1 (**Spatiotemporal Graph Generation**).* The problem of the STG generation is to learn an overall distribution $P_r(G)$ from the observed STGs, where each G is denoted as the union of all the spatiotemporal walks. Novel STGs $\hat{G}$ can be sampled from the distribution such that $\hat{G} \sim P_r(\cdot)$.

There are several challenges in solving the novel STG generation problem: *1)* It is difficult to capture the joint distribution of the multi-modal properties in $\bar{s}$ since its properties are characterized with both categorical distributions (e.g., x, y, and (u_i, v_i)) and continuous-value distribution (e.g., t_i and (u_i, v_i)). *2)* Correctly decoding all the information from the learned distribution poses another challenge. The validity of each graph component ((u_i, v_i), t_i, and the calculated spatial locations $(l_u^{(i)}, l_v^{(i)})$) requires extra attention since the generated STGs need to have realistic semantic meaning (i.e., $t_{L_\mathbf{s}} \leq t_i \leq t_0$ and $(l_u^{(i)}, l_v^{(i)})$ is valid in a prescribed spatial region). *3)* It is also difficult to characterize the dependency among multi-modal properties in different STGs due to the interplay among spatial, temporal, and graph information.

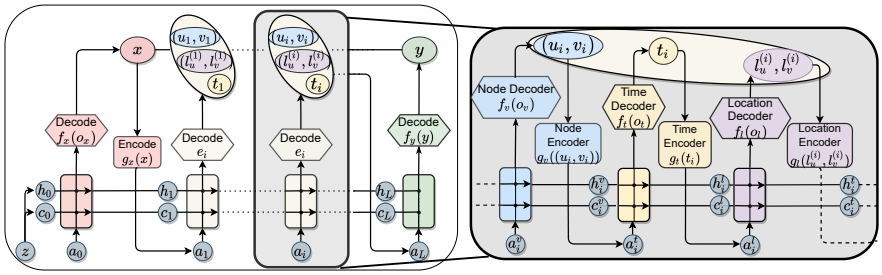


Fig. 3: Overview of spatiotemporal walk generator

4 Generative Model for Spatiotemporal Graph

In this section, we first introduce the backbone of our framework - STGEN for generating continuous-time STG. Then, we elaborate on each component of the generative framework, namely the spatiotemporal walk generator $\mathcal{G}$ and the boosted spatiotemporal walk discriminator $\mathcal{D}$.

4.1 Overall architecture

STGEN learns the distribution of STG through a generative adversarial architecture, which consists of two parts: a recurrent-structure-based spatiotemporal walk generator $\mathcal{G}$ (as shown in Fig. 3), and a boosted spatiotemporal walk discriminator $\mathcal{D}$ (as shown in Fig. 6). The training of both generator $\mathcal{G}$ and discriminator $\mathcal{D}$ are conducted under the framework of Wasserstein GAN (WGAN) [2] to maximize the discrepancy $W(P_r, P_\theta)$ between the real STG distribution P_r and the generated STG distribution P_θ such that:

$$W(P_r, P_\theta) = \max \left[\mathbb{E}_{\mathbf{s}_r \sim P_r}[\mathcal{D}(\mathbf{s}_r)] - \mathbb{E}_{\mathbf{s}_\theta \sim P_\theta}[\mathcal{D}(\mathbf{s}_\theta)] \right]$$
$$s.t.\ \mathbf{s}_\theta = \mathcal{G}(z) \sim P_\theta, \mathcal{T}(\mathbf{s}_\theta) \in \mathbb{T}, \mathcal{K}(\mathbf{s}_\theta) \in \mathbb{K} \tag{1}$$

where $\mathbf{s}_r$ and $\mathbf{s}_\theta$ are real and fake spatiotemporal walks sampled from P_r and P_θ, respectively. z is a latent noise sampled from the standard normal distribution. Specifically, the spatiotemporal walk generator $\mathcal{G}$ trains a fixed-length LSTM whose output $\mathbf{s}_\theta = \mathcal{G}(z)$ is a *segmented* spatiotemporal walk. The time budgets as well as the spatial attributes of these generated spatiotemporal walks $\mathbf{s}_\theta$ are regulated by a temporal activation layer $\mathcal{T}$ and a spatial activation layer $\mathcal{K}$, such that $\mathcal{T}(\mathbf{s}_\theta)$ and $\mathcal{K}(\mathbf{s}_\theta)$ are valid in respect of both temporal constraint $\mathbb{T}$ and spatial constraint $\mathbb{K}$. We introduce both the spatiotemporal walk generator and the corresponding STG assembler in Sec. 4.2. Moreover, due to the multi-modal nature of the spatiotemporal walk (i.e., node sequence, time budget on the edge, and the spatial attribute), a boosted discriminator is proposed for each combination of the spatiotemporal walk components to characterize dependencies among all components. We give further details of the boosted discriminator and theoretical analysis in Sec. 4.3.

Model Complexity. Existing static and dynamic graph generative models usually generate graphs through generating adjacency matrices or snapshots, and they require at least $O(N^2 \cdot M_s)$ time complexity, where N is the number of nodes, and M_s is the number of snapshots. For large-scale or dynamic graphs with a large timespan but small granularity, these models [32, 25, 9] would suffer from the scalability and information loss. However, since a continuous-time STG is composed of spatiotemporal walks and does not involve with an adjacency matrix, it makes the overall complexity of STGEN to be $O(L_{\mathbf{s}} \cdot M_e)$ since STGEN only needs to generate segmented spatiotemporal walks and assemble them as a whole, where $L_{\mathbf{s}}$ is the maximal length of all the spatiotemporal walks and M_e is the number of spatiotemporal walks needed to compose an STG.

4.2 Spatiotemporal Walk Generator with Validity Constraints.

Segmented spatiotemporal walk is a heterogeneous sequence where some elements are categorical signals (starting point x and ending point y), continuous-value scalars (t_i, $l_u^{(i)}, l_v^{(i)}$), and edges (u_i, v_i). To effectively characterize all the modalities in segmented spatiotemporal walks, we propose a novel heterogeneous recurrent-structured generator with various encoding/decoding functions for various modalities. In addition, we propose new activation functions to enforce spatiotemporal validity constraints on the generated spatiotemporal walk patterns. Finally, an STG assembler is proposed to compose final STGs by conditionally generating spatiotemporal walks given other spatiotemporal walks.

Segmented Spatiotemporal Walk Generation. The generator $\mathcal{G}$ defines an implicit probabilistic model for generating segmented spatiotemporal walks $\bar{\mathbf{s}}$ that are similar to the real spatiotemporal walks $\mathbf{s}$ sampled from P_r, and its overall architecture is summarized in Fig. 3. Specifically, the generator is modeled by a fixed-length LSTM model. Each LSTM unit keeps a hidden state h_i and cell state c_i as memory state, takes a_i as input, and returns o_i as output. The generator outputs the generated time budget t_i, node pair (u_i, v_i), and the spatial attributes $(l_u^{(i)}, l_v^{(i)})$ through decoding o_i in different decoding functions in a sequential order. The decoding functions can be divided into two categories: *discrete-value* and *continuous-value* decoding functions. *Discrete-value* decoding functions include a node decoding function: $f_v(o_{v_i})$ that outputs the node v_i, a starting point decoding function $f_x(o_x)$ that outputs the starting point x, and a ending point decoding function $f_y(o_y)$ that outputs the ending point y. *Continuous-value* decoding functions include a time decoding function $f_t(o_t)$ that outputs a residual time budget t_i on the generated edge, and the time budget t_i is regulated by the temporal activation layer $\mathcal{T}$ in order to meet the temporal constraint $\mathbb{T}$. In addition, STGEN also contains a location decoding function $f_l(o_l)$ that outputs spatial attributes $l_u^{(i)}$ and $l_v^{(i)}$ for both end nodes u_i and v_i on the generated edge. Likewise, the generated spatial attributes are also regulated by a spatial activation layer $\mathcal{K}$ to meet the spatial constraint $\mathbb{K}$. Other than the decoding functions, the generator $\mathcal{G}$ also uses different encoding functions to encode each generated components back to the next LSTM unit input a_{i+1}.

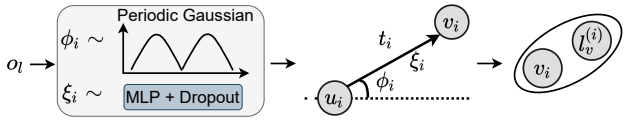


Fig. 4: Example of the sampling procedure of location decoder $f_l(\cdot)$.

We employ Multi-layer Perceptron (MLP) structures for all encoding functions. For discrete-value decoding functions, a Gumbel-softmax trick [17] is applied to make the sampling procedure differentiable. We describe the continuous-value decoding functions in more details in the following part. Due to space limit, we summarize the overall generative process in the supplementary material.

With Spatiotemporal Validity Constrained Multi-modal Decoding Functions. The time budget t_i as well as the spatial attributes $(l_u^{(i)}, l_v^{(i)})$ (e.g., GPS coordinates) of a spatiotemporal walk in many real world situations are typically irregular but following different underlying distributions, which makes both components cannot be trivially decoded by deterministic functions. Therefore, t_i and $(l_u^{(i)}, l_v^{(i)})$ are assumed to be sampled from a latent distribution. The sampling procedure is handled by both time decoder $f_t(\cdot)$ and location decoder $f_l(\cdot)$. Particularly, $f_t(\cdot)$ is an end-to-end sampling function that convert the latent representation o_i to parameters of an prescribed distribution (e.g., μ and σ in Gaussian distribution). Then, $t_i \sim f_t(o_t)$ could be sampled directly. In order to fulfill the temporal constraint, we apply a activation layer $\mathcal{T}$ to ensure the temporal validity of the generated segmented spatiotemporal walks. Specifically, we propose to impose a *Min-max Bounding* in the activation layer $\mathcal{T}$:

$$\mathcal{T}(t_i) = \begin{cases} t_i = t_i - \min(\{t_i\}), & \text{if } \min(\{t_i\}) \leq \epsilon \\ t_i = t_i / \max(\{t_i\}), & \text{if } \max(\{t_i\}) > 1 \\ t_i = t_i, & \text{otherwise} \end{cases}$$

where $\min(\{t_i\})$ and $\max(\{t_i\})$ are the min and max time budget in the generated mini-batch $\{t_i\}$, respectively. ϵ is a threshold with small value (i.e., $1e{-}3$) to prevent zero value for t_i. On the other side, spatial locations in many real-world situations have higher variance and could not to be typically described by known distributions. Instead of directly sampling exact locations, in this work, we sample the relative turning angle ϕ_i and the speed ξ in the spatiotemporal edge e_i for computing the node location $l_v^{(i)}$ and $l_u^{(i)}$. Specifically, we leverage MLP and dropout to mimic the sampling operation to obtain a continuous-valued speed ξ_i on spatiotemporal edge e_i from the LSTM unit's latent input o_l. Moreover, we model the distribution of turning angle ϕ_i to fit the Periodic Gaussian distribution [6]. Based on the sampled time budget t_i, speed ξ_i, and turning angle ϕ_i, the relative distance can be directly computed. By assigning the initial location for the first node in the generated spatiotemporal walk, the locations for the subsequent node can also be determined. We visualize the whole procedure in Fig. 4. To further regulate the generated node locations to have semantic meanings,

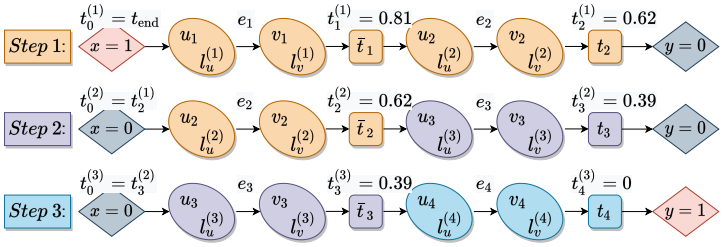


Fig. 5: Example of the spatiotemporal walk assembly. For step 1 (first line), the generator initially generates a segmented spatiotemporal walk with $x = 1$ and $y = 0, t_0^{(1)} = t_{\text{end}}$ containing two spatiotemporal edges (orange ovals) indicating it is the start of a spatiotemporal walk. At Step 2 and 3, the generator generates one additional edge e_3 (purple oval) and e_4 (blue oval) conditioned on the inputs of last edge, respectively. The final time budget $t_4^{(3)} = 0$ indicating the end of the spatiotemporal walk ($x = 0$ and $y = 1$) has reached.

we propose a *spatial activation layer* $\mathcal{K}$ with *Geographical Bounding*:

$$\mathcal{K}(l_v^{(i)}) = \begin{cases} l_v^{(i)} = l_{\xi_j}, & \text{if } \tau(l_v^{(i)}) \notin \bigcup_{j=1}^{H} \tau(\psi_j) \\ l_v^{(i)} = l_v^{(i)}, & \text{otherwise} \end{cases}$$

where $\{\psi_j\}$ is a set of prescribed geographical areas with size H. Particularly, for each generated geo-location $l_v^{(i)}$ generated in the set $\{l_v^{(i)}\}$, we project both $l_v^{(i)}$ and $\{\psi_j\}$ to the same space with a geographical projection function $\tau(\cdot)$ (e.g., Universal Transverse Mercator projection). If the projected $\tau(l_v^{(i)})$ belongs to any $\tau(\psi_j)$, this generated geo-location is valid and has realistic semantic meaning. Otherwise, this geo-location will be replaced by the closest point l_{ψ_j} on the prescribed area ψ_j that has the minimal distance to the original $l_v^{(i)}$.

Spatiotemporal Walk Assembler. The next step is to compose spatiotemporal walks from these segmented spatiotemporal walks generated from $\mathcal{G}$. In order to force the consistency of the underlying spatiotemporal diffusion pattern when we concatenate two segmented spatiotemporal walk, we may not chronologically concatenate walks purely based on their residual time budget t_i. Instead, we start by generating an initial segmented spatiotemporal walk $\bar{\mathbf{s}}_1 = (e_1^{(1)}, ..., e_L^{(1)})$ with $x = 1, y = 0$, and $t_0 = t_{\text{end}}$ that contains L spatiotemporal edges. The last spatiotemporal edge $e_L^{(1)}$ of $\bar{\mathbf{s}}_1$ is taken as the input to generate the next segmented spatiotemporal walk $\bar{\mathbf{s}}_2 = (e_L^{(1)}, e_2^{(2)}, ..., e_L^{(2)})$ starting from $e_L^{(2)}$. In this case, $\bar{\mathbf{s}}_2$ can be appended to $\bar{\mathbf{s}}_1$ with the guarantee of following the underlying diffusion pattern. We incrementally appending additional segmented spatiotemporal walks until we run out of the time budget (i.e., $t_i = 0$) to form one final spatiotemporal walk (with the ending flag $x = 0, y = 1$). The overall process of assembling a spatiotemporal walk is illustrated in Fig. 5.

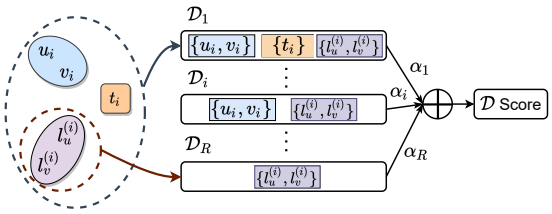


Fig. 6: The overview of the boosted discriminator. Each $\mathcal{D}_i$ is a weak classifier that takes a certain combination of spatiotemporal walk components.

4.3 Spatiotemporal Walk Discriminator

Discriminator $\mathcal{D}$ is employed to distinguish real and generated spatiotemporal walks, which is difficult due to their multi-modal nature. We aim to impose a stronger discriminator that can guide the generator characterizing both dependency and independency among the spatial, temporal, and graph modalities in STGs. Beyond a single discriminator that merely jointly considers spatial, temporal, and graph aspects of the walk, we propose to consider all the combinatorial of these aspects, such as spatial-temporal discriminator, temporal graph discriminator, spatial graph discriminator, etc. Our method is achieved by leveraging boosting strategy, which is well-recognized to enable the ensemble of models to outperform each individual model. Using a unified discriminator would also bring up the well-known training instability and potential mode collapse since the GAN may fall into recognizing only one of the generated modalities as the real sample while neglecting other modalities. Thus, we consider a multi-discriminator structure to better approximate Eq. (1) and constantly provide a harsher critique to the generator by considering the combinatorial of all modalities. A detailed architecture of the boosted discriminator is shown in Fig. 6.

Particularly, we adopt an adaptive boosted structure of the discriminator for discriminating each combination $\mathbf{s}^{(i)}$ of the spatiotemporal walk components, and there are a total of R combinations, such as spatiotemporal component (t_i and $(l_u^{(i)}, l_v^{(i)})$) and joint STG element ((u_i, v_i), $(l_u^{(i)}, l_v^{(i)})$, and t_i), etc. The boosted discriminator $\mathcal{D}$ takes the voting result over R sub discriminators $\mathcal{D}_i$ so that each sub discriminator $\mathcal{D}_i$ is performed as a weak-classifier. Such a boosted discriminator structure forces $\mathcal{G}$ to generate high fidelity samples that must hold up under the scrutiny of all R discriminators. The major voting strategy in our adaptive boosting can be induced into the WGAN objective and lead to a well-modularized objective function in Theorem 1.

Table 1: Dataset Description

	Syn_100	Syn_500	Syn_2000	Taxi	Check-in	Citation
Node	100	500	2,000	66	70	628
Temporal Edge	5,606	5,750	5,750	28,532	17,045	914
Temporal Samples	60	110	120	30	30	30

Theorem 1. *The aforementioned adaptive boosting strategy extends the objective function $W(P_r, P_\theta)$ in Eq.* (1) *of STGEN into the following:*

$$W(P_r, P_\theta) = \max \sum_{i=1}^{R} \alpha_i \Big[\mathbb{E}_{\mathbf{s}_r^{(i)} \sim P_r} [\mathcal{D}_i(\mathbf{s}_r^{(i)})] - \mathbb{E}_{\mathbf{s}_\theta^{(i)} \sim P_\theta} [\mathcal{D}_i(\mathbf{s}_\theta^{(i)})] \Big],$$
$$s.t., \ \mathbf{s}_\theta = \mathcal{G}(z) \sim P_\theta, \mathcal{G}(z) \in \{\mathbb{T}, \mathbb{K}\}, \sum_{i=1}^{R} \alpha_i = \frac{R}{4}.$$

With the objective of maximizing the above objective function, we can minimize the loss function of the generator $\mathcal{L}_G = -\sum_{i=1}^{R} \alpha_i \cdot \mathbb{E}_{\mathbf{s}_\theta^{(i)} \sim P_\theta} [\mathcal{D}_i(\mathbf{s}_\theta^{(i)})]$, where as well as the discriminator's overall loss function $\mathcal{L}_\mathcal{D} = \sum_{i=1}^{R} \alpha_i \cdot \mathcal{L}_{\mathcal{D}_i}$, where each sub discriminator's loss function is defined as: $\mathcal{L}_{\mathcal{D}_i} = \big[\mathbb{E}_{\mathbf{s}_\theta^{(i)} \sim P_\theta} [\mathcal{D}_i(\mathbf{s}_\theta^{(i)})] - \mathbb{E}_{\mathbf{s}_r^{(i)} \sim P_r} [\mathcal{D}_i(\mathbf{s}_r^{(i)})]\big]$. The proof of Theorem 1 can be found in Appendix. We illustrate the overall training framework in the supplementary material.

5 Experiment

In this section, we demonstrate the performance of our proposed STGEN framework across various synthetic and real world STGs. Basic experiment settings are illustrated here. Additional experiments (e.g., sensitivity analysis) are provided in the supplementary material. Code and data are made available[3].

Data. We performed experiments on three synthetic and three real-world STGs with different graph sizes and characteristics, where basic statistics are shown in Table 1. All graphs contain continuous timestamps as temporal information and geo-coordinates (latitude and longitude) as spatial information. Due to the space limit, details of all graphs can be found in the supplementary material.

Comparison Methods. Since there is no existing methods handling the STG generation problem, STGEN is compared with two categories of methods: *deep graph generation methods*: a) GraphRNN [32], b) NetGAN [7], c) TagGen [35], d) TG-GAN [34], and e) STGD-VAE [9]; and *spatial attribute generation methods*: a) LSTM [13], b) SVAE and its variant SVAE-γ [15], and c) IGMM-GAN [26]. Details of all baselines can be found in the supplementary material.

[3] github.com/lingchen0331/STGEN

Table 2: Performance comparison between real and generated graph samples in maximum mean discrepancy (MMD) (the lower the better), models with the best performance are marked with black.

Dataset	Method	AND	AGS	AGN	ACN	GD	Method	MCD	VCP
Syn_100	GraphRNN	4.2 e −3	7.12 e −3	0.175	7.2 e −2	7.12 e −2	LSTM	1.09 e −1	51.37%
	NetGAN	3.64 e −5	1.89 e −3	0.0613	2.77 e −2	5.17 e −3	SVAE-y	4.17 e −1	61.24%
	TagGen	4.14 e −4	2.72 e −3	0.0911	6.18 e −3	4.11 e −2	SVAE	2.41 e −1	50.19%
	TG-GAN	3.16 e −5	5.31 e −3	**0.0221**	3.06 e −3	4.17 e −3	IGMM-GAN	9.12 e −3	69.13%
	STGD-VAE	2.53 e −5	3.57 e −3	0.0367	5.15 e −3	7.92 e −3	–	–	–
	STGEN	**2.31 e −5**	**1.96 e −3**	0.0315	**2.14 e −3**	**3.89 e −3**	STGEN	**3.52 e −3**	**100%**
Syn_500	GraphRNN	7.64 e −4	3.73 e −5	0.0525	3.72 e −2	4.13 e −2	LSTM	4.56 e −1	57.15%
	NetGAN	2.19 e −4	6.12 e −4	0.023	7.71 e −3	7.18 e −3	SVAE-y	7.12 e −2	69.21%
	TagGen	3.24 e −5	1.23 e −2	0.3912	1.71 e −2	**5.86 e −3**	SVAE	6.12 e −2	59.19%
	TG-GAN	1.44 e −4	8.5 e −3	**0.0023**	1.24 e −2	5.12 e −2	IGMM-GAN	9.12 e −2	49.14%
	STGD-VAE	7.98 e −5	2.67 e −4	0.0024	3.97 e −3	8.43 e −3	–	–	–
	STGEN	**2.29 e −5**	**4.71 e −3**	0.009	**2.19 e −3**	7.41 e −3	STGEN	**9.18 e −3**	**100%**
Syn_2000	GraphRNN	2.79 e −4	7.4 e −2	0.042	2.88 e −2	2.15 e −2	LSTM	4.22 e −1	51.33%
	NetGAN	3.67 e −5	2.69 e −2	0.0472	6.92 e −3	6.22 e −3	SVAE-y	2.33 e −1	48.12%
	TagGen	2.67 e −4	1.98 e −2	0.031	5.57 e −3	5.66 e −3	SVAE	7.28 e −2	50.67%
	TG-GAN	2.66 e −5	**4.19 e −4**	0.012	4.95 e −3	**2.97 e −3**	IGMM-GAN	5.43 e −2	33.27%
	STGD-VAE	5.48 e −4	2.67 e −2	0.037	9.42 e −3	6.43 e −3	–	–	–
	STGEN	**2.56 e −5**	4.34 e −3	**0.009**	**4.49 e −3**	9.76 e −3	STGEN	**1.07 e −2**	**100%**
Taxi	GraphRNN	4.73 e −3	4.63 e −3	0.0226	3.2 e −3	2.57 e −2	LSTM	1.1852	17.23%
	NetGAN	8.29 e −1	5.25 e −6	0.0189	5.63 e −4	7.87 e −3	SVAE-y	2.37 e −2	21.44%
	TagGen	3.92 e −2	7.24 e −4	0.0221	3.58 e −4	3.91 e −3	SVAE	3.11 e −2	20.46%
	TG-GAN	6.69 e −4	8.87 e −6	**0.0132**	**1.06 e −5**	2.67 e −3	IGMM-GAN	1.67 e −1	9.54%
	STGD-VAE	**9.61 e −6**	3.61 e −4	0.0233	7.26 e −4	4.74 e −3	–	–	–
	STGEN	9.85 e −5	**3.09 e −6**	0.0165	1.17 e −5	**2.55 e −3**	STGEN	**3.39 e −3**	**100%**
Check-in	GraphRNN	3.5 e −3	2.89 e −2	0.0312	4.78 e −4	5.32 e −2	LSTM	0.0963	3.77%
	NetGAN	4.37 e −3	2.54 e −4	0.063	4.38 e −4	**2.38 e −3**	SVAE-y	2.67 e −2	23.11%
	TagGen	1.27 e −2	1.77 e −2	0.0292	3.58 e −4	3.91 e −3	SVAE	1.79 e −2	19.24%
	TG-GAN	7.69 e −4	**3.76 e −6**	0.0139	1.79 e −5	3.99 e −3	IGMM-GAN	2.37 e −1	15.23%
	STGD-VAE	1.33 e −4	6.71 e −3	0.0283	3.12 e −4	9.95 e −2	–	–	–
	STGEN	**9.17 e −5**	5.74 e −6	**0.0126**	**1.39 e −5**	3.57 e −3	STGEN	**1.58 e −4**	**100%**
Citation	GraphRNN	3.24 e −1	3.12 e −2	0.0465	1.98 e −3	4.21 e −2	LSTM	1.6857	45.18%
	NetGAN	5.2 e −1	3.77 e −3	0.0577	3.13 e −3	2.18 e −2	SVAE-y	7.41 e −1	8.33%
	TagGen	3.34 e −2	3.24 e −3	0.0561	7.71 e −5	3.67 e −3	SVAE	5.91 e −1	10.84%
	TG-GAN	1.24 e −2	1.78 e −5	0.0587	1.13 e −4	**1.38 e −4**	IGMM-GAN	1.47 e −1	3.21%
	STGD-VAE	3.75 e −1	3.64 e −3	0.0493	5.67 e −4	2.15 e −2	–	–	–
	STGEN	**3.04 e −3**	**1.23 e −5**	**0.0398**	**3.66 e −5**	2.77 e −3	STGEN	**3.77 e −2**	**100%**

Note that GraphRNN, NetGAN, TagGen, and STGD-VAE are discrete graph generative models, and they cannot generate continuous-time STGs. We instead modify them to generate multiple discrete-snapshot graphs and convert them into a continuous-time temporal graph. In addition, STGD-VAE cannot generate realistic spatial attributes (i.e., GPS locations) so that we only compare STGD-VAE in generating graph properties. **Evaluation Metrics.** A set of metrics, as elucidated in Table 2, are used to measure the similarity between the generated and real STGs in terms of both temporal and spatial graph attributes. For temporal attributes measurement, we adopt AND: Average Node Degree, AGS: Average Group Size, AGN: Average Group Number, ACN: Average Coordination Number, and GD: Group Duration. For spatial attributes measurement, we leverage MCD: Mean Coordinates Distribution and VCP: Valid Coordinates Percentage.

5.1 Quantitative Performance.

The overall performance comparisons are shown in Table 2. The proposed STGEN generally outperforms other methods in terms of both temporal and spatial attributes generation with only a few exceptions. Specifically, in terms of the similarity in temporal graph properties, STGEN performs better than static graph generation methods (i.e., GraphRNN and NetGAN) by, on average, two orders of magnitudes in several temporal graph properties (e.g., AND, AGS, and ACN). The main reason is that static graph generation methods generate dynamic graphs via generating a series of snap-shots, which may cause severe information loss [14]. STGEN also consistently achieves competitive results with dynamic graph generation methods (i.e., TagGen and TG-GAN) among all datasets since STGEN can effectively capture the underlying multi-modal distribution of STGs. Compared with the only discrete STG generation method - STGD-VAE, STGEN still exhibits an overall better performance in generating continuous STGs. In terms of the spatial graph properties generation, STGEN exceeds other approaches with an evident margin (two orders of magnitudes on average) in MCD while achieving a 100% validity rate of the generated spatial properties among all datasets. With the applied spatiotemporal constraints, the coordinates generated by STGEN can always be regulated in a valid semantic region and guarantee a 100% validity rate, while other methods can only achieve at most 70% validity rate. In other words, a large portion of the coordinates generated by baseline methods do not have valid semantic meanings.

5.2 Case Study.

Spatiotemporal walks in STGs are typically associated with various purposes, such as "wandering in attractions" and "picking-up people from the airport". In other words, each node in STGs has a semantic meaning that can be projected to a certain geographic area. We thus conduct a case study to evaluate the quality of the generated spatial information by projecting each of the generated GPS coordinates to a real-world map. Taking the Manhattan taxi trip STG

(a) LSTM (b) IGMM-GAN (c) SVAE (d) **STGEN** (e) Ground Truth

Fig. 7: The comparison of generated coordinates by each baseline.

Table 3: Average runtime comparison (in minutes).

	GraphRNN	NetGAN	TagGen	TG-GAN	STGD-VAE	STGEN
Syn_100	23.3561	0.8126	1.1682	0.9533	1.9781	0.9783
Syn_500	71.5622	1.9346	2.2653	1.4861	3.9923	1.6458
Syn_2000	337.7614	10.3302	8.5324	6.0182	10.8779	6.9633

as an example, we project a batch of generated coordinates for each method to the Manhattan map, and the results are shown in Fig. 7. Compared with the true coordinates (Fig. 7e), our proposed method STGEN (Fig. 7d) generates coordinates that all lie within the valid region by successfully characterizing the spatiotemporal properties. However, other deep methods like LSTM generate a large portion of coordinates scattered all over the New York City area since LSTM cannot effectively decode the spatial information from the multi-modal spatiotemporal distribution of STG. As can be seen from Fig. 7a to 7c, coordinates generated by comparison methods are disorganized (coordinates are not in the Manhattan area) and lack real semantic meaning (coordinates are projected on the sea) since they cannot consider any constraints during the model learning.

5.3 Model Scalability.

The runtime of graph generative methods is often composed of model training and graph assembling. Therefore, we record the average training time per epoch plus the graph assembling time, and the results are shown in Table 3. All the runtimes are shown with respect to the growth of graph size for all synthetic STGs. As can be seen from the table, TagGen, TG-GAN, and STGEN have linear growth regarding the graph size because these dynamic graph generation methods decode dynamic graphs into dynamic walks instead of utilizing snapshots, which makes both training and graph assembling processes in these methods are not sensitive to the overall graph size. Although NetGAN also utilizes random walk to learn the static graph distribution, but its random walk sampling process limits its capability in generating large dynamic graphs with many snapshots. Finally, the runtime growth of GraphRNN is exponential because of its quadratic complexity in modeling the whole graph as a sequence.

6 Conclusion

In this paper, we propose a novel generative framework for continuous-time STG generation, which can effectively model the underlying dynamics of STGs while maintaining the spatiotemporal validity. Our framework captures both graph and spatiotemporal distribution through utilizing a novel heterogeneous recurrent-structured generator to learn the distribution of sampled spatiotemporal walks. A novel boosted discriminator is proposed to characterize correlations between all modalities in STGs. Extensive experiments are conducted on generating both synthetic and real world STGs. Experimental results and analysis demonstrate the advantages of STGEN over existing deep graph generative models in terms of generating the most similar and realistic STGs.

Acknowledgement

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. Ahuja, R.K., Magnanti, T.L., Orlin, J.B.: Network flows (1988)
2. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein generative adversarial networks. In: ICML (2017)
3. Banavar, J.R., Maritan, A., Rinaldo, A.: Size and form in efficient transportation networks. Nature (1999)
4. Barthélemy, M.: Spatial networks. Physics Reports **499**(1-3) (2011)
5. Bhattacharya, U., Mittal, T., Chandra, R., Randhavane, T., Bera, A., Manocha, D.: Step: Spatial temporal graph convolutional networks for emotion perception from gaits. In: AAAI (2020)
6. Bishop, C.M., Nasrabadi, N.M.: Pattern recognition and machine learning (2006)
7. Bojchevski, A., Shchur, O., Zügner, D., Günnemann, S.: Netgan: Generating graphs via random walks. In: ICML (2018)
8. Cui, Z., Li, Z., Wu, S., Zhang, X., Liu, Q., Wang, L., Ai, M.: Dygcn: Dynamic graph embedding with graph convolutional network. arXiv (2021)
9. Du, Y., Guo, X., Cao, H., Ye, Y., Zhao, L.: Disentangled spatiotemporal graph generative models. In: AAAI (2022)
10. Goyal, P., Kamra, N., He, X., Liu, Y.: Dyngem: Deep embedding method for dynamic graphs. arXiv preprint arXiv:1805.11273 (2018)
11. Guo, X., Du, Y., Zhao, L.: Deep generative models for spatial networks. In: KDD. pp. 505–515 (2021)
12. Guo, X., Zhao, L.: A systematic survey on deep generative models for graph generation. arXiv (2020)
13. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8) (1997)
14. Holme, P.: Modern temporal network theory: a colloquium. Eur Phys J B (2015)

15. Huang, D., Song, X., Fan, Z., Jiang, R., Shibasaki, R., Zhang, Y., Wang, H., Kato, Y.: A variational autoencoder based generative model of urban human mobility. In: MIPR (2019)
16. Jain, A., Zamir, A.R., Savarese, S., Saxena, A.: Structural-rnn: Deep learning on spatio-temporal graphs. In: CVPR (2016)
17. Jang, E., Gu, S., Poole, B.: Categorical reparameterization with gumbel-softmax. arXiv (2016)
18. Li, S., Da Xu, L., Zhao, S.: The internet of things: a survey. Inf Syst Front (2015)
19. Ling, C., Yang, C., Zhao, L.: Deep generation of heterogeneous networks. In: ICDM. pp. 379–388 (2021)
20. Masuda, N., Holme, P.: Predicting and controlling infectious disease epidemics using temporal networks. F1000prime reports **5** (2013)
21. Melo-Vega, A., Frausto-Solís, J., Castilla-Valdez, G., Liñán-García, E., González-Barbosa, J.J., Terán-Villanueva, D.: Protein folding problem in the case of peptides solved by hybrid simulated annealing algorithms. In: Fuzzy Logic Augmentation of Neural and Optimization Algorithms: Theoretical Aspects and Real Applications (2018)
22. Sabirin, H., Kim, M.: Moving object detection and tracking using a spatio-temporal graph in h. 264/avc bitstreams for video surveillance. TOM (2012)
23. Schröter, M.S., Spoormaker, V.I., Schorer, A., Wohlschläger, A., Czisch, M., Kochs, E.F., Zimmer, C., Hemmer, B., Schneider, G., Jordan, D., et al.: Spatiotemporal reconfiguration of large-scale brain functional networks during propofol-induced loss of consciousness. Journal of Neuroscience (2012)
24. Scott, J.: Social network analysis. Sociology **22** (1988)
25. Simonovsky, M., Komodakis, N.: Graphvae: Towards generation of small graphs using variational autoencoders. In: ICANN (2018)
26. Smolyak, D., Gray, K., Badirli, S., Mohler, G.: Coupled igmm-gans with applications to anomaly detection in human mobility data. ACM TSAS (2020)
27. Solé, R.V., Rosas-Casals, M., Corominas-Murtra, B., Valverde, S.: Robustness of the european power grids under intentional attack. Physical Review E (2008)
28. Wang, S., Guo, X., Zhao, L.: Deep generative model for periodic graphs. arXiv preprint arXiv:2201.11932 (2022)
29. Watts, D.J., Strogatz, S.H.: Collective dynamics of 'small-world'networks. nature (1998)
30. Wiewel, S., Becher, M., Thuerey, N.: Latent space physics: Towards learning the temporal evolution of fluid flow. In: Computer graphics forum (2019)
31. Wu, L., Cui, P., Pei, J., Zhao, L., Song, L.: Graph neural networks. In: Graph Neural Networks: Foundations, Frontiers, and Applications (2022)
32. You, J., Ying, R., Ren, X., Hamilton, W., Leskovec, J.: Graphrnn: Generating realistic graphs with deep auto-regressive models. In: ICML. pp. 5708–5717 (2018)
33. Yu, B., Yin, H., Zhu, Z.: Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. In: IJCAI (2018)
34. Zhang, L., Zhao, L., Qin, S., Pfoser, D., Ling, C.: Tg-gan: Continuous-time temporal graph generation with deep generative models. In: WebConf (2020)
35. Zhou, D., Zheng, L., Han, J., He, J.: A data-driven graph generative model for temporal interaction networks. In: KDD (2020)