Automated Urban Planning for Reimagining City Configuration via Adversarial Learning: Quantification, Generation, and Evaluation

DONGJIE WANG, YANJIE FU, and KUNPENG LIU, University of Central Florida
FANGLAN CHEN, Virginia Tech
PENGYANG WANG, University of Macau
CHANG-TIEN LU, Virginia Tech

Urban planning refers to the efforts of designing land-use configurations given a region. However, to obtain effective urban plans, urban experts have to spend much time and effort analyzing sophisticated planning constraints based on domain knowledge and personal experiences. To alleviate the heavy burden of them and produce consistent urban plans, we want to ask that can AI accelerate the urban planning process, so that human planners only adjust generated configurations for specific needs? The recent advance of deep generative models provides a possible answer, which inspires us to automate urban planning from an adversarial learning perspective. However, three major challenges arise: (1) how to define a quantitative land-use configuration? (2) how to automate configuration planning? (3) how to evaluate the quality of a generated configuration? In this article, we systematically address the three challenges. Specifically, (1) We define a land-use configuration as a longitude-latitude-channel tensor. (2) We formulate the automated urban planning problem into a task of deep generative learning. The objective is to generate a configuration tensor given the surrounding contexts of a target region. In particular, we first construct spatial graphs using geographic and human mobility data crawled from websites to learn graph representations. We then combine each target area and its surrounding context representations as a tuple, and categorize all tuples into positive (well-planned areas) and negative samples (poorly-planned areas). Next, we develop an adversarial learning framework, in which a generator takes the surrounding context representations as input to generate a land-use configuration, and a discriminator learns to distinguish between positive and negative samples. (3) We provide quantitative evaluation metrics and conduct extensive experiments to demonstrate the effectiveness of our framework.

CCS Concepts: • **Social and professional topics** → *Professional topics*; *Computing and business*; Automation;

Additional Key Words and Phrases: Urban planning, representation learning, generative adversarial networks, graph neural networks

Authors' addresses: D. Wang, Y. Fu, and K. Liu, University of Central Florida, 4328 Scorpius Street, Orlando, FL 32816-2362; emails: wangdongjie@knights.ucf.edu, yanjie.fu@ucf.edu, kunpengliu@knights.ucf.edu; F. Chen, Virginia Tech, 7054 Haycock Rd, Falls Church; email: fanglanc@vt.edu; P. Wang, University of Macau, Taipa, Macau, China; email: pywang@um.edu.mo; C.-T. Lu, Virginia Tech, 7054 Haycock Rd, Falls Church, VA 22043; email: ctlu@vt.edu.

1 INTRODUCTION

We study the problem of machine learning for automated urban planning. Urban planning is an interdisciplinary and complex process that involves public policy, social science, engineering, architecture, landscape, and other related fields. In thisarticle, we refer urban planning to the efforts of designing land-use configurations of a target region, which is a reduced yet essential task of urban planning [36]. Effective urban planning can help to mitigate the operational and social vulnerability of an urban system, such as high tax, crimes, traffic congestion and accidents, pollution, depression, and anxiety [52].

This observation motivates us to rethink urban planning in the era of **artificial intelligence** (**AI**): What roles do deep learning play in urban planning? Can machines develop and learn at a human capability to automatically and quickly calculate land-use configurations? In this way, machines can be planning assistants and urban planning professionals can finally adjust machine-generated plans for specific needs.

Due to the high complexity and specificity of urban systems, urban planners need to consider and balance different various planning requirements, such as proximity metrics (e.g., distances to important places), access indexes (e.g., accessibility to food, recreation, goods, services, entertainment, transit, municipal services, mobility indexes), mobility indices (e.g., sidewalks, bike lanes, speed limits, crash rates), emergency responses (e.g., hospitals, fire departments) and, thus, planing highly replies on empirical experience and domain knowledge [25]. As a result, it is highly appealing to pursuit a fast, automated, and machine-assisted planning strategy. The recent advance of deep learning, particularly deep adversarial and generative learning, provide a great potential to teach a machine at a human capability to design and generate city configurations [10, 24, 55, 56]. This inspiration motivates us to rethink urban planning from the lens of deep learning: can AI automate the calculation of land-use configuration and the balancing of various planning factors, so professional planners can finally adjust machine-generated plans for specific needs?

All of the above evidences prompt us to develop a data-driven AI-enabled automated urban planner as shown in Figure 1. However, three unique challenges arise to achieve the goal: (1) How can we quantify a land-use configuration plan? (2) How can we develop a deep adversarial generative learning framework to learn the good and the bad of existing urban communities as data-driven knowledge, and, moreover, generate quality urban configuration? (3) How can we evaluate the quality of generated land-use configurations? Next, we will introduce our research insights and solutions for the three challenges.

First, as the objective is to teach a machine to generate the land-use configuration of a target region, it is naturally critical to define a machine- perceivable structure for the land-use configuration. In practice, the land-use configuration plan of a region can be geographically defined by a set of **Point of Interests (POIs)** and their corresponding locations (e.g., latitudes and longitudes) and urban functionality categories (e.g., shopping, banks, education, entertainment, residential). A close look can reveal that a land-use configuration is a high-dimensional indicator that precisely illustrates what, where, and how many we should build in a target region. After exploring large-scale land-use data, we observe that there is not just location-location statistical autocorrelation but also location-functionality statistical autocorrelation in a land-use configuration. To
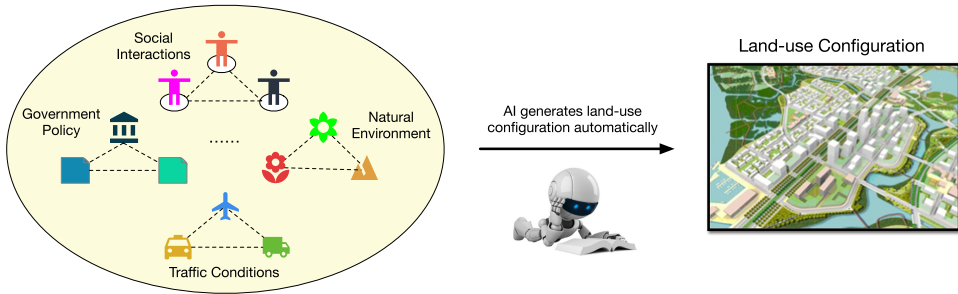
Fig. 1. Our expectation is to build an AI model to generate the land-use configuration automatically by considering the constraints that are used in traditional urban planning process.

preserve such statistical correlations, we propose to represent a land-use configuration as a latitude-longitude-channel tensor, where each channel is a specific category of POIs that are distributed across the target area, and the value of an entry in the tensor is the number of POIs. In this way, the tensor not just describes the location-location interaction, but also captures location-function interaction.

Second, after we quantitatively define the land-use configuration, the next question is that how to teach a machine to automatically generate a land-use configuration? We analyze large-scale urban residential community data, and find that: (1) an urban community can be viewed as an attributed node in a socioeconomic network (city as a community-community network), and this node proactively interacts with surrounding nodes (environments); (2) the coupling, interaction, and coordination of a community and surrounding environments significantly influence the livability, vibrancy, and quality of a community. Based on the above observations, we aim at developing a function that map the surrounding contexts to a well-planned configuration tensor. Recently, the development of deep generative and adversarial learning provides a great potential for solving this problem. We reformulate the task into an adversarial learning paradigm, in which: (1) A neural generator is analogized as a machine planner that generates a land-use configuration; (2) The generator generates a configuration in terms of the feature representation of surrounding spatial contexts; (3) The surrounding context feature representation is learned via self-supervised representation learning collectively from spatial graphs. (4) A neural discriminator is to classify whether the generated land-use configuration is well-planned (positive) or poorly-planned (negative). (5) A new mini-max loss function is constructed to guide the generator to learn the configuration patterns of well-planned areas, compared to poorly-planned areas.

Third, how can we evaluate the quality of a generated land-use configuration? This has been a long-standing challenging question. The most solid and sound validation is to collaborate with urban developers and city agencies to implement a machine-generated configuration into a target region to observe the development of the region in the following years. However, the validation method is not practical in reality. In this article, we design and develop three strategies to assess the generated configurations: (1) We leverage different distance measurements to measure the similarity between generated configurations and well-planned configurations. If the distance is small, it indicates that our generated configurations preserve the overarching distribution characteristics of well-planned configurations. (2) We develop a scoring model to score the quality of the generated configurations. Specifically, since we have collected a set of existing land-use configurations and 0–1 labels (1: well-planned 0: poorly-planned) as training data, we train a regression model to predict the quality score ranging from 0 to 1. After that, given a machine-generated configuration as testing data, we use the regression model to predict its

corresponding score. (3) We use a variety of visualization approaches to visualize the generated configurations, so domain experts can evaluate the generated quality and rationality.

Our preliminary work in [38] proposed a fundamental automated urban planning framework to automatically generate land-use configurations. The preliminary framework can be further improved to enhance its stability and efficiency from a computational perspective. For this purpose, in this journal version, we develop a new conditioning augmentation module adding to the preliminary framework to enhance its performance. Specifically, we first estimate the distribution of the embedding space of surrounding spatial contexts. Then, we sample embedding vectors from the estimated distribution of the embedding space to replace the embeddings of surrounding spatial contexts. The benefit is to use embedding space distribution estimation to augment data and overcome the sparsity of surrounding spatial context data. Later, we propose a new loss function that considers the embedding space regularization and standardization of surrounding spatial contexts. The new loss function can accelerate the convergence and improve the efficiency of learning. In addition, aside from prediction-based and visualization-based valuation approaches [38], in this article, we design a distance-based strategy to evaluate the quality of machine-generated configuration plans.

In summary, in both our preliminary work [38] and this extended version, we develop an adversarial learning framework to generate effective land-use configurations by learning from urban geography, human mobility, and socioeconomic data. Specifically, our contributions are: (1) We develop a latitude-longitude-channel tensor to quantify a land-use configuration plan. (2) We propose a socioeconomic interaction perspective to understand urban planning as a process of optimizing the coupling between a community and surrounding environments. (3) We reformulate the automated urban planning problem into an adversarial learning framework that maps surrounding spatial contexts into a configuration tensor. (4) We computationally enhance the efficiency and stability of the proposed framework by devising a conditioning augmentation module via leveraging a new sampling technique and a new optimization loss function. (5) We develop multiple strategies (i.e., distance-based, prediction-based, and visualization-based) to validate the effectiveness of our framework on real-world data.

2 PROBLEM STATEMENT AND FRAMEWORK OVERVIEW

2.1 Definitions

2.1.1 Target Area. Refers to a geographical area, where is centered on a geographical location (described by latitude and longitude), and the shape of the area is square.

2.1.2 Surrounding Contexts. Refer to the surrounding squares, which wraps the target area from different directions. The shape of each square in the surrounding contexts is same as the target area. In our research assumption, we have known the information such as demographic data, social activity, traffic volume, and so on, of the surrounding contexts. According to the geographical vicinity and the information of the surrounding contexts, we construct a spatial attributed graph, in which the vertices are the squares of the surrounding contexts and the attributes of each vertex are extracted from the information of each square. Figure 2 shows the geographical spatial relations between a target area and the surrounding contexts, in which different contexts have different urban utility and characteristics. Our framework aims to generate the land-use configuration of the target area based on the surrounding contexts.

2.2 Problem Statement

As mentioned before, we aim at building up an automated generation framework that generates land-use configuration of the target area based on the surrounding contexts. Formally, assuming

Fig. 2. The geographical spatial relations between a target area and the surrounding contexts.
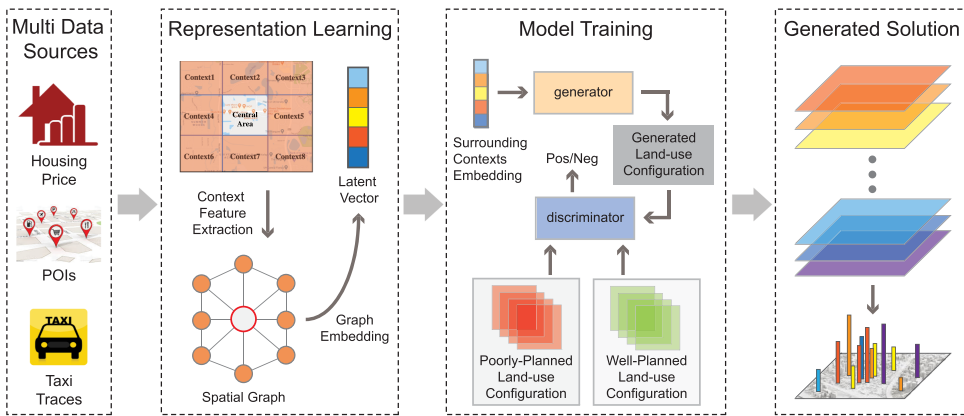

Fig. 3. An overview of the proposed framework. The proposed framework includes four steps: 1) we first collect multiple data sources such as urban community related data (housing prices), point of interests data, and human mobility data (taxicab GPS traces). We then utilize a spatial graph representation learning module to learn the representations of surrounding contexts. Finally, we develop an adversarial land-use configuration generation model to automate planning and generate recommended configurations.

a target area is R, the surrounding contexts of R are $[C_1 \sim C_K]$, and the land-use configuration for R is $\mathbf{M}$ that is a longitude-latitude-channel tensor. Given a spatial attributed graph G that is constructed by extracting explicit features such as traffic condition, economic development, and so on, from surrounding contexts $[C_1 \sim C_K]$, we aim at finding the mapping function $f : G \rightarrow \mathbf{M}$. The function takes the spatial attributed graph G as input, and outputs the land-use configuration $\mathbf{M}$. In this article, owing to the shape of the target area is square, the number of the squares in the surrounding contexts is determined as $K = 8$.

2.3 Framework Overview

Figure 3 shows an overview of our proposed method (LUCGAN). This framework has two main phases: (i) surrounding contexts embedding phase; (ii) land-use configuration generation phase. In the surrounding contexts embedding phase, we first extract explicit features of the surrounding contexts from multiple aspects, such as value-added space, POI distribution. Then, we model the eight squares of the surrounding contexts as eight vertices and map the explicit features to the vertices as the corresponding attribute to construct a spatial attributed graph. Next, we employ a graph

embedding model to preserve the information of the graph into an embedding vector. Through the above procedures, the final embedding vector represents the whole surrounding contexts. In the land-use configuration generation phase, we first input the embedding of the contexts into an extended **generative adversarial networks (GANs)**. Then, the GAN model learns to formulate the distribution of the well-planned land-use configurations instead of poorly-planned configurations gradually. Finally, when the model converges, the extended GAN can produce suitable and desired land-use configurations based on the embeddings of the surrounding contexts.

3 AUTOMATIC PLANNER FOR LAND-USE CONFIGURATION

In this section, we first introduce the strategy to represent surrounding contexts. Then, we detail how to quantify and evaluate the quality of land-use configurations. Finally, we develop an automated urban planner based on deep generative adversarial paradigm.

3.1 Extraction of Explicit Features of Surrounding Contexts

The surrounding contexts affect the land-use configuration of a target area. For instance, if the surrounding contexts own lots of recreational facilities, in order to avoid waste of resources, we will not plan lots of recreational buildings in the target area. Instead, we prefer to choose other kinds of buildings such as commercial or educational buildings to make the target area coexist with the surrounding contexts in harmony. Thus, based on the observation, during land-use configuration generation process, it is necessary to take surrounding contexts into consideration. In this article, we extract the explicit features of surrounding contexts from four aspects:

(1) **Value-Added Space.** Commonly, the variation of housing price reflects the value-added space of one area. Thus, we calculate the dynamically changing trend of housing price of the contexts $[C_1 \sim C_8]$ in continuous t months. Here, we take the context C_1 as an example to explain the calculation process. First, we obtain the housing price list among t months. Then, we calculate the changing trend of housing price by using the current housing price to subtract the previous housing price. So we get the changing trend of C_1 as $\mathbf{v}_1 = [v_1^1, v_1^2, \ldots, v_1^{t-1}]$, where v_1^i represents the value of the changing trend at ith month. Finally, we collect the housing price changing trend of all contexts together. The collected result is denoted as $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_8]$, where $\mathbf{V} \in \mathbb{R}^{8 \times (t-1)}$.

(2) **POI Ratio.** Since different POIs provide different services for residents, the ratio of different kinds of POIs can reflect the utility of one area. Therefore, we calculate the POI ratio of the contexts $[C_1 \sim C_8]$. Here, we take C_1 as an example to explain the calculation process. First, we count the total number of POI belonging to each POI category in C_1, respectively, to form a vector. Then, we divide each item in the vector by the number of all POIs in C_1 to obtain the POI ratio vector, denoted by $\mathbf{r}_1 = [r_1^1, r_1^2, \ldots, r_1^m]$, where r_1^i represents the ratio of ith POI category and m is the number of POI categories. Finally, we collect the POI ratio vector of all contexts together, denoted as $\mathbf{R} = [\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_8]$, where $\mathbf{R} \in \mathbb{R}^{8 \times m}$.

(3) **Public Transportation.** Public transportation (i.e., bus, subway) is one of the most important travel modes due to its convenience and economy. We need to consider the public transportation of the contexts $C_1 \sim C_8$. Here, we take C_1 as an example to show the calculation details. To capture the characteristics of public transportation, we extract features based on bus trajectory and bus station data from five perspectives: (1) the leaving volume of C_1 in one day, denoted by $\mathbf{o}_1^1$; (2) the arriving volume of C_1 in one day, denoted by $\mathbf{o}_1^2$; (3) the transition volume of C_1 in one day, denoted by $\mathbf{o}_1^3$; (4) the density of bus stop of C_1, denoted by $\mathbf{o}_1^4$; (5) the average balance of smart card of C_1, denoted by $\mathbf{o}_1^5$. Thus, the feature vector of C_1 can
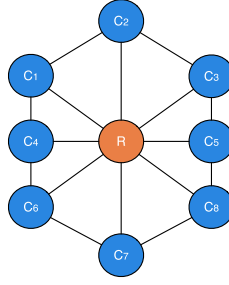
Fig. 4. The spatial structural relation between a target area and its surrounding contexts.

be denoted as $[o_1^1, o_1^2, \ldots, o_1^5]$. Finally, we collect the feature vectors of all contexts together. The collected result is denoted as $\mathbf{O} = [\mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_8]$, where $\mathbf{O} \in \mathbb{R}^{8 \times 5}$.

(4) **Private Transportation.** Private transportation (i.e., taxi, cab) is another important travel mode for individuals due to its flexibility. We extract the features of private transportation of the contexts $[C_1 \sim C_8]$ based on taxi trajectory data from five perspectives. Taking C_1 as an example, the definitions of the fie features are as follows: (1) the leaving volume of C_1 in one day, denoted by u_1^1; (2) the arriving volume of C_1 in one day, denoted by u_1^2; (3) the transition volume of C_1 in one day, denoted by u_1^3; (4) in C_1, the average driving velocity of taxis in one hour, denoted by u_1^4; (5) in C_1, the average commute distance of taxis in one hour, denoted by u_1^5; Then, the feature vector of private transportation is denoted as $[u_1^1, u_1^2, \ldots, u_1^5]$. Finally, we collect all context features together, denoted as $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_8]$, where $\mathbf{U} \in \mathbb{R}^{8 \times 5}$.

After that, we obtain an explicit feature set from the contexts $C_1 \sim C_8$. The set contains four kinds of features $[\mathbf{V}, \mathbf{R}, \mathbf{O}, \mathbf{U}]$, which describes the surrounding contexts from aforementioned perspectives.

3.2 Constructing Spatial Attributed Graphs with Explicit Features as Node Attributes

The surrounding contexts wrap the target area from different directions, resulting in spatial correlation among areas. To capture the spatial correlations among the areas, we construct a spatial attributed graph. Specifically, Figure 4 shows the graph structural relation between a target area and its surrounding contexts, where the blue vertices represent the surrounding contexts; the orange vertex indicates the target area; the edge between two vertices reflects the spatial connectivity between them.

Then, we map the explicit features $[\mathbf{V}, \mathbf{R}, \mathbf{O}, \mathbf{U}]$ to the spatial graph structure as the corresponding node attributes. Figure 5 expresses the mapping process. The final spatial attributed graph not only reflects the spatial correlation among different context squares but also depicts the utility characteristics of each square.

3.3 Learning Representation of Spatial Attributed Graphs

Figure 6 shows the spatial representation learning framework that preserves explicit features and spatial relations of the spatial attributed graph into a low-dimensional vector. Formally, we denote the spatial attributed graph as $G = (\mathbf{X}, \mathbf{A})$, where $\mathbf{A} \in \mathbb{R}^{8 \times 8}$ is the adjacency matrix that expresses the accessibility among different nodes; $X \in \mathbb{R}^{8 \times (t+m+9)}$ is the feature matrix of the graph, here, $\mathbf{X} = [\mathbf{V}, \mathbf{R}, \mathbf{O}, \mathbf{U}]$. In order to get the latent graph embedding $\mathbf{z}$, we minimize the reconstruction loss between the original graph G and the reconstructed graph $\widehat{G}$ through the encoding-decoding paradigm.
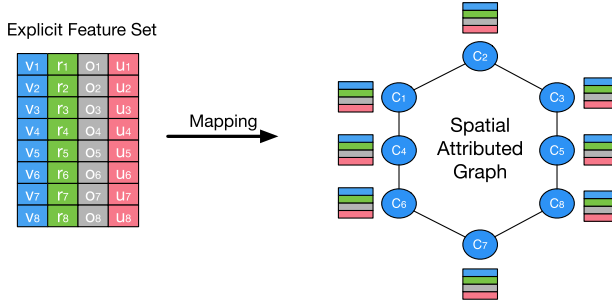
Fig. 5. The illustration of constructing a spatial attributed graph: Each feature vector is mapped to the corresponding vertex by a column-wise strategy.
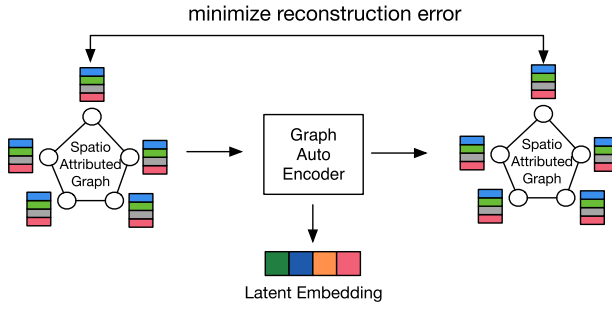

Fig. 6. The proposed representation learning module to obtain surrounding context representations by minimizing the reconstruction loss of spatial attributed graphs.

The encoding part has two GCN layers. The first GCN layer takes $\mathbf{X}$ and $\mathbf{A}$ as input and outputs the feature matrix of low-dimensional space $\widehat{\mathbf{X}}$. Thus, the encoding module can be formulated as

$$\widehat{\mathbf{X}} = GCN_1(\mathbf{X}, \mathbf{A}) = RELU\left(\widehat{\mathbf{D}}^{-\frac{1}{2}}\mathbf{A}\widehat{\mathbf{D}}^{-\frac{1}{2}}\mathbf{X}\mathbf{W}_1\right), \tag{1}$$

where $\widehat{\mathbf{D}} \in \mathbb{R}^{8 \times 8}$ is the diagonal degree matrix, $\mathbf{W}_1 \in \mathbb{R}^{8 \times M}$ is the weight matrix of the GCN_1 where M is the output dimension of the layer, and the whole layer is activated by $RELU$ function. The second GCN layer takes $\widehat{\mathbf{X}}$ and $\mathbf{A}$ as input and then outputs the mean value $\boldsymbol{\mu}$ and the variance value $\boldsymbol{\delta}^2$ of normal distribution. So the calculation process of the second GCN layer can be formulated as

$$\boldsymbol{\mu}, log(\boldsymbol{\delta}^2) = GCN_2(\widehat{\mathbf{X}}, \mathbf{A}) = \widehat{\mathbf{D}}^{-\frac{1}{2}}\mathbf{A}\widehat{\mathbf{D}}^{-\frac{1}{2}}\widehat{\mathbf{X}}\mathbf{W}_2, \tag{2}$$

where $\mathbf{W}_2 \in \mathbb{R}^{M \times H}$ is the weight matrix of GCN_2. Here, H is the output dimension of the GCN_2 layer. Next, we use the reparameterization trick to obtain the latent representation $\mathbf{z} \in \mathbb{R}^{8 \times H}$:

$$\mathbf{z} = \boldsymbol{\mu} + \boldsymbol{\delta} \times \epsilon, \tag{3}$$

where $\epsilon \sim \mathcal{N}(0, 1)$.

The decoding module takes the $\mathbf{z}$ as input and then outputs the reconstructed adjacent matrix $\widehat{\mathbf{A}}$. Hence, the decoding step can be formulated as

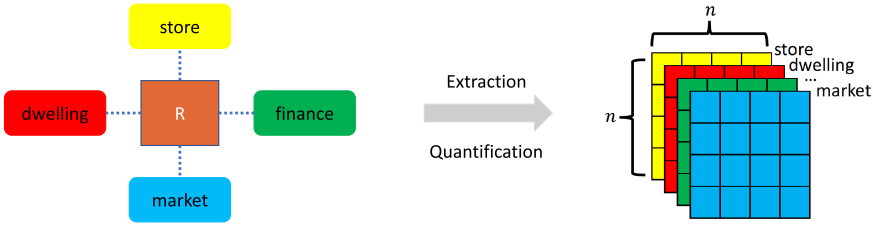$$\widehat{\mathbf{A}} = \sigma(\mathbf{z}\mathbf{z}^T), \tag{4}$$

Fig. 7. The construction of longitude-latitude-channel configuration tensor. Specifically, we first collect the information such as dwelling, market, and so on, of an area. Then, we extract and quantify the information of the area as a multi-channel tensor, where the value of each entry is the number of POIs with respect to a specific category in a specific latitude range and longitude range.

where σ represents the decoding layer activated by sigmoid function. Moreover, $\mathbf{z}\mathbf{z}^T$ can be converted to $\|\mathbf{z}\|\|\mathbf{z}^T\|\cos\theta$. The inner product operation is beneficial to capture the spatial correlation among different contexts.

During the training phase, we minimize the joint loss function $\mathcal{L}$, denoted as

$$\mathcal{L} = \sum_{i=1}^{N} \underbrace{KL[q(\mathbf{z}|\mathbf{X},\mathbf{A})||p(\mathbf{z})]}_{\text{KL Divergence between } q(.) \text{ and } p(.)} + \overbrace{\sum_{j=1}^{S}\left\|\mathbf{A}-\widehat{\mathbf{A}}\right\|^2}^{\text{Loss between } \mathbf{A} \text{ and } \widehat{\mathbf{A}}}, \tag{5}$$

where N is the dimension of $\mathbf{z}$; S is the total number of the vertices in $\mathbf{A}$; q represents the real distribution of $\mathbf{z}$; p represents the prior distribution of $\mathbf{z}$. $\mathcal{L}$ includes two parts, the first part is the **Kullback–Leibler (KL)** divergence between the standard prior distribution $\mathcal{N}(0,1)$ and the distribution of $\mathbf{z}$, and the second part is the squared error between $\mathbf{A}$ and $\widehat{\mathbf{A}}$. The training process try to make $\widehat{\mathbf{A}}$ get close to $\mathbf{A}$ and let the distribution of $\mathbf{z}$ get close to $\mathcal{N}(0,1)$. When the model converges, $\mathbf{z}$ contains all information of the surrounding contexts.

3.4 Land-Use Configuration Quantification and Quality Measurement

Land-use configuration indicates the location of different kinds of POIs in one area. To make a machine perceive and understand the configuration, we construct a longitude-latitude-channel tensor as the format of the configuration, where one channel denotes one POI category and the whole tensor represents the POI distribution in the area. Figure 7 shows the construction process of the longitude-latitude-channel configuration tensor. We first divide an unplanned target area into $n \times n$ squares. Then we count the number of POIs belonging to each POI category in each square entry and fill the number into the corresponding entry, respectively. In this way, we obtain the land-use configuration tensor. If we pick up one channel from the tensor, we can learn about the POI distribution of the corresponding POI category in the whole area.

Owing to we expect the generation framework can generate well-planned land-use configuration, the next big question is how to evaluate the quality of the configuration? In the classical urban planning domain, there are no general evaluation standards since the complexity of urban systems. To make our framework can produce the land-use configurations that people satisfied with, we provide a quality hyper-parameter Q to evaluate the quality of land-use configurations. In our experiment, Q is the combination of the POI diversity and the check-in frequency. Formally, we first count the total frequency number of mobile check-in events of an area, which reflect the social activity intensity, denoted by $freq$. Then, we calculate the total number of different POI categories of the area as the POI diversity, which depicts the completeness of urban functions,
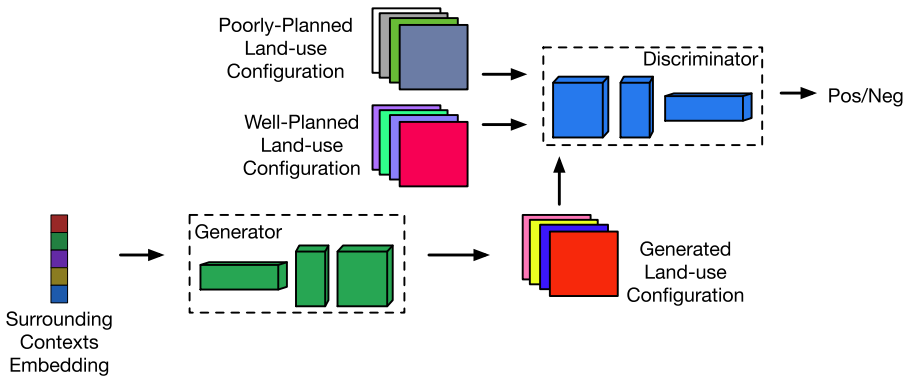
Fig. 8. The model structure of LUCGAN.

denoted by div. Next, we incorporate the two indicators together by the formula $Q = \frac{2 \times freq \times div}{freq + div}$ [51]. If $Q >$ threshold, the configuration of the area is regarded as a well-planned configuration, otherwise, it is justified as a poorly-planned configuration. Here, the value of *threshold* is determined by given requirements.

3.5 Land-Use Configuration Generative Adversarial Networks

Recently, GANs achieve tremendous achievements and reveal strong imaginative and generative abilities. It motivates us to formulate the land-use configuration generation task into the learning paradigm of GAN.

In our preliminary version [38], we propose a **land-use configuration GAN** (**LUCGAN**), and the network structure of LUCGAN as illustrated in Figure 8. In LUCGAN, the generator generates land-use configuration based on the embeddings of surrounding contexts. The discriminator provides feedback to the generator for generating configurations close to well-planned configurations instead of poorly-planned configurations.

The Algorithm 1 shows the training process of LUCGAN. Specifically, in one training iteration, we first update the parameters of the discriminator for κ times, then learn the parameters of the generator for one time based on the current discriminator. For the updating process of the discriminator, we sample m well-planned configurations, surrounding context embeddings, and poorly-planned configurations, respectively. We utilize them to maximize the loss function illustrated in line 10 of Algorithm 1. Intuitively, we expect the discriminator to provide positive feedback for well-planned configurations, and negative feedback for poorly-planned and generated configurations. In this way, the discriminator improves the distinguishing ability for land-use configurations. For the updating process of the generator, we sample m surrounding context embeddings firstly. Then, we minimize the loss function shown in line 14 of Algorithm 1. Intuitively, we aim at utilizing the discriminator to improve the generative ability of the generator for producing data structures similar to well-planned configurations.

However, the embeddings of the surrounding contexts come from a feature space constructed by spatial attributed graphs. Owing to the small number of graphs, the distribution of them in the feature space is sparse and discrete, which causes the learning process of the GAN model unstable. To overcome this limitation and improve model performance, we propose an enhanced framework, namely LUCGAN⁺, and the network structure as shown in Figure 9. Compared with Figure 8, we add a conditioning augmentation module [54] into our framework. Specifically, we first assume the prior distribution of the surrounding contexts embeddings is a normal distribution.
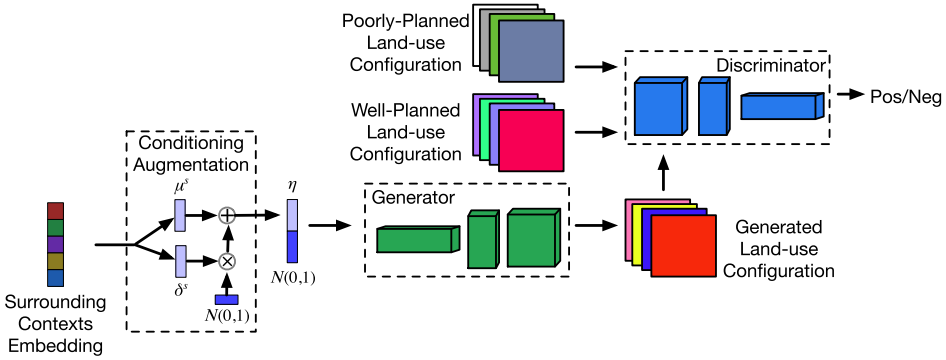
Fig. 9. The model structure of LUCGAN$^+$.

ALGORITHM 1: The training process of LUCGAN. Here, G denotes the generator, D denotes the discriminator. We adjust the hyperparameter κ to change the updating frequencies of the parameters of the discriminator.

---

1 // start training.

2 **for** *number of training iterations* **do**

3   // update discriminator firstly.

4   **for** κ *steps* **do**

5   Sample m well-planned land-use configuration samples $\{\mathbf{E}^1, \mathbf{E}^2, \dots, \mathbf{E}^m\}$.

6   Sample m surrounding context embeddings $\{\mathbf{z}^1, \mathbf{z}^2, \dots, \mathbf{z}^m\}$.

7   Generate land-use configurations by generator, $\{\mathbf{F}^1, \mathbf{F}^2, \dots, \mathbf{F}^m\}$, $\mathbf{F}^i = G(\mathbf{z}^i)$.

8   Sample m poorly-planned land-use configurations $\{\mathbf{T}^1, \mathbf{T}^2, \dots, \mathbf{T}^m\}$.

9   Update the discriminator by maximizing the following loss:

10          $\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} [log(D(\mathbf{E}^i)) + log(D(1 - \mathbf{F}^i)) + log(D(1 - \mathbf{T}^i))]$.

11   // update generator secondly.

12   Sample m context information embedding samples $\{\mathbf{z}^1, \mathbf{z}^2, \dots, \mathbf{z}^m\}$.

13   Update the generator by minimizing the following loss:

14      $\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} log(1 - D(G(\mathbf{z}^i)))$.

Then, we estimate the mean and variance of the distribution based on the original embeddings. Next, we sample a vector from the distribution and combine it with a vector sampled from standard normal distribution as the input vector of the model. This process improves the model performance because it mitigates the discreteness and sparsity of original graphs in the feature space.

In addition, owing to the differences of the model structure between LUCGAN and LUCGAN$^+$, we customize a new training algorithm for LUCGAN$^+$ as shown in Algorithm 2. Compared with Algorithm 1, there are two improvements: (1) conditioning augmentation module (line 4–7 in Algorithm 2); (2) loss function of the generator (line 19 in Algorithm 2). For the conditioning augmentation module, we calculate the mean $\boldsymbol{\mu}^s$ and the variance $\boldsymbol{\delta}^s$ based on original surrounding context embeddings, respectively. Then, we utilize reparametrization trick to sample a vector from normal distribution $\mathcal{N}(\boldsymbol{\mu}^s, \boldsymbol{\delta}^s)$, and concatenate the vector with a vector sampled from normal distribution $\mathcal{N}(0, 1)$ as the surrogate context embeddings $\boldsymbol{\eta}$. For the learning process of the discriminator, the main logic is the same as Algorithm 1, we only replace the surrounding context embeddings $\mathbf{z}$ with $\boldsymbol{\eta}$. For the loss of the generator, besides improving the generative capability of the generator, we

---

ALGORITHM 2: The training process of LUCGAN$^+$. Here, G denotes the generator, D denotes the discriminator. We adjust the hyperparameter κ to change the updating frequencies of the parameters of the discriminator.

---

1 // start training.
2 **for** *number of training iterations* **do**
3   // Conditioning Augmentation. μ^s, δ^s are mean and variance respectively. $\mathbf{W}_{()}, \mathbf{b}_{()}$ are weight and bias of the corresponding item respectively. $\epsilon^{()}$ indicates that sampling a vector from normal distribution $\mathcal{N}(0,1)$.
4   Sample m context information embedding samples $\mathbf{Z} = \left\{\mathbf{z}^1, \mathbf{z}^2, \ldots, \mathbf{z}^m\right\}$.
5      $\mu^s = RELU(\mathbf{W}_\mu \cdot \mathbf{Z} + \mathbf{b}_\mu)$;
6      $\delta^s = RELU(\mathbf{W}_\delta \cdot \mathbf{Z} + \mathbf{b}_\delta)$.
7      $\eta = Concatenate((\mu^s + \delta^s \cdot \epsilon^s), \epsilon^c)$.
8   // update discriminator firstly
9   **for** κ *steps* **do**
10   Sample m well-planned land-use configurations $\left\{\mathbf{E}^1, \mathbf{E}^2, \ldots, \mathbf{E}^m\right\}$.
11   Collect m vectors as context embeddings thorough line 6 $\left\{\eta^1, \eta^2, \ldots, \eta^m\right\}$.
12   Generate land-use configurations by generator, $\left\{\mathbf{F}^1, \mathbf{F}^2, \ldots, \mathbf{F}^m\right\}$, $\mathbf{F}^i = G(\eta^i)$.
13   Sample m poorly-planned land-use configurations $\left\{\mathbf{T}^1, \mathbf{T}^2, \ldots, \mathbf{T}^m\right\}$.
14   Update the discriminator by maximizing the following loss:
15         $\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^m [log(D(\mathbf{E}^i)) + log(D(1 - \mathbf{F}^i)) + log(D(1 - \mathbf{T}^i))]$.
16   // update generator secondly. KL means Kullback–Leibler divergence
17   Collect m vectors as contexts embeddings thorough line 6 $\left\{\eta^1, \eta^2, \ldots, \eta^m\right\}$.
18   Update the generator by minimizing the following loss:
19     $\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^m log(1 - D(G(\eta^i))) + KL(\mathcal{N}(\mu^s, \delta^s) \,||\, \mathcal{N}(0,1))$.

---

also minimize the KL divergence between $\mathcal{N}(\mu^s, \delta^s)$ and $\mathcal{N}(0,1)$, which enhances the smoothness of the surrounding context embeddings in the feature space and avoids overfitting.

4 EXPERIMENT RESULTS

In this section, we conduct extensive experiments and case studies to answer the following questions: **Q1.** Is our proposed automatic planner effective for generating land-use configurations? **Q2.** We split an area into $n \times n$ squares for quantifying land-use configurations. What is the influence of the square size for generating configurations? **Q3.** What are the differences between the contexts of well-planned configurations poorly-planned configurations? **Q4.** What are the differences of land-use configurations generated by our framework when facing with different planning goals? **Q5.** What does the generated result for each POI category look like in a generated land-use configuration?

4.1 Data Description

We use the following datasets for evaluation: **Residential Community:** The residential community dataset contains 2990 residential communities in Beijing.[1] Each community is centered by a geographic point (described by latitude and longitude). **POI:** The POI dataset includes 328668 POIs in Beijing.[2] Each POI item includes latitude, longitude, and the corresponding POI category.

Table 1. POI Category List

code	POI category	code	POI category
0	road	10	tourist attraction
1	car service	11	real estate
2	car repair	12	government place
3	motorbike service	13	education
4	food service	14	transportation
5	shopping	15	finance
6	daily life service	16	company
7	recreation service	17	road furniture
8	medical service	18	specific address
9	lodging	19	public service

Table 1 shows the detailed information of POI category. **Taxi Trajectories:** The taxi trajectories are collected from a Beijing taxi company.[3] Each trajectory contains trip ID, distance (m), travel time (s), average speed (km/h), pick-up and drop-off time, pick-up and drop-off point. **Public Transportation:** The public transportation dataset includes bus transactions in Beijing from 2012 to 2013, which contains 718 bus lines, 1,734,247 bus trips.[4] **Housing Price:** The housing price dataset is collected from a Chinese real estate website,[5] which contains the housing price of residential communities of Beijing from 2011 to 2012. **Check-In:** The check-in dataset contains the Weibo[6] check-in records in Beijing from 2011 to 2013. The data format of one record is: longitude, latitude, check-in time, and check-in place.

4.2 Evaluation Metrics

We aim at generating land-use configurations that are similar to well-planned configurations. To evaluate the generative performance, we calculate the difference between the distribution of well-planned configurations Y and the distribution of generated configurations $\widehat{Y}$. The less distribution difference is, the better generative performance will be.

(1) **Kullback–Leibler (KL) Divergence:** $KL(Y||\widehat{Y}) = \sum Y(x) \cdot \ln \frac{Y(x)}{\widehat{Y}(x)}$, where x is a test sample.

(2) **Jensen–Shannon (JS) Divergence:** $JS(Y||\widehat{Y}) = \frac{1}{2}KL(Y||\frac{Y+\widehat{Y}}{2}) + \frac{1}{2}KL(\widehat{Y}||\frac{Y+\widehat{Y}}{2})$.

(3) **Hellinger Distance (HD):** $HD(Y||\widehat{Y}) = \frac{1}{\sqrt{2}}||\sqrt{Y} - \sqrt{\widehat{Y}}||_2$.

(4) **Wasserstein Distance (WD):** $WD(Y||\widehat{Y}) = \inf_{\gamma \sim \Gamma(Y,\widehat{Y})} \mathbb{E}_{(x,y) \sim \gamma} ||x - y||$, where $\Gamma(Y, \widehat{Y})$ is a set of joint distribution between Y and $\widehat{Y}$; γ is a joint distribution of Γ; x, y are two samples sampled from γ; $\mathbb{E}||(.)||$ is the expectation of distances between any two samples.

4.3 Baseline Methods

We compare the performance of our journal version framework (LUCGAN$^+$) against the following baseline models:

(1) **DCGAN**: is an extension for traditional GAN, which utilizes convolutional layer and convolutional transpose layer in the generator and discriminator, respectively [32].

---

[3]https://www.microsoft.com/en-us/research/publication/t-drive-trajectory-data-sample/.
[4]https://www.beijingcitylab.com/data-released-1/data1-20/.
[5]http://www.soufun.com/.
[6]https://open.weibo.com/wiki/2/place/pois/add_checkin.

(2) **WGAN**: is a new GAN training framework, which improves the stability of learning and provides meaningful learning curve for debugging and hyperparameter adjustment [4].

(3) **WGAN^{GP}**: utilizes gradient penalty to replace clipping weights of WGAN, which enhances the performance of WGAN further [16].

(4) **LUCGAN**: is the conference version of our land-use configuration GAN, which is capable of generating the configurations based on the surrounding contexts [38].

To further study the generated land-use configurations, we adopt two new methods: **scoring model** and **visualization**. For the scoring model, we train a machine learning model to learn the scoring criteria that provide high score for well-planned configurations and low score for poorly-planned configurations. After we obtain all testing samples, the model can be used to evaluate the quality of generated results. For the visualization, we visualize the generated results in heat map, pie chart, 3d-bar chart for checking the POI distribution. We conduct all experiments on a x64 machine with Intel i9-9920 × 3.50 GHz CPU, 128 GB RAM, and Ubuntu 18.04.

4.4 Hyperparameters and Reproducibility

In our experiments, first, to obtain the embedding of surrounding contexts (Section 3.3), we employ a VGAE [20] composed of an encoder and a decoder. The encoder contains three graph convolutional neural layers. The decoder only has one reconstructed layer. We perform Adaptive Moment Estimation (Adam) to optimize the VGAE model with a learning rate of 0.005 for 300 epochs. The dimension of surrounding contexts' embedding is set to 100. Second, to quantify the quality of land-use configurations (Section 3.4), we set the value of the hyper-parameter Q to 0.5. Third, our planner LUCGAN$^+$ consists of a generator and a discriminator (Section 3.5). We optimize the generator by Adam with a learning rate of 0.0001. We perform **Stochastic Gradient Descent** (**SGD**) to optimize the discriminator with a learning rate of 0.0001 and a momentum of 0.95. The whole optimizing process continues for 50 epochs. To make other researchers easily reproduce our experiments, we release the code and data by Dropbox.[7]

4.5 Overall Performance (Q1)

To validate the effectiveness of our model, we evaluate the gap between the distribution of well-planned configurations and the distribution of generated configurations in terms of KL Divergence (KL), JS Divergence (JS), Hellinger Distance (HD), and Wasserstein Distance (WD). As Figure 10 shows, compared with the best performance of baseline models (WGAN, WGAN^{GP}, DCGAN), LUCGAN$^+$ improves 16.2%, 0.25%, 28.4%, 48.6% in terms of KL, JS, HD, and, WD, respectively. This observation indicates that LUCGAN$^+$ can capture more characteristics of the well-planned configurations compared with other baseline models. In addition, another interesting observation is that compared with LUCGAN, LUCGAN$^+$ increases 8.92%, 0.23%, 8.43%, 4.32% in terms of KL, JS, HD, and WD, respectively. A potential interpretation for the observation is that the conditioning augmentation module and the new training approach of LUCGAN$^+$ makes the learning process more stable and effective.

4.6 Study the Influence of the Square Size for Generating Land-Use Configurations (Q2)

To quantify the land-use configuration, we divide an area into $n \times n$ squares to collect the POI distribution information. To study the influence of the square size for generation, we vary $n = 5$, $n = 10$, $n = 25$, $n = 50$, $n = 100$ to conduct experiments. Here, the smaller value of n is, the larger

---

[7]https://www.dropbox.com/sh/16pk55efb9fzm2j/AACsosXxHtfQKXKjmL0NrOn1a?dl=0.

(a) KL Divergence

(b) JS Divergence

(c) Hellinger Distance
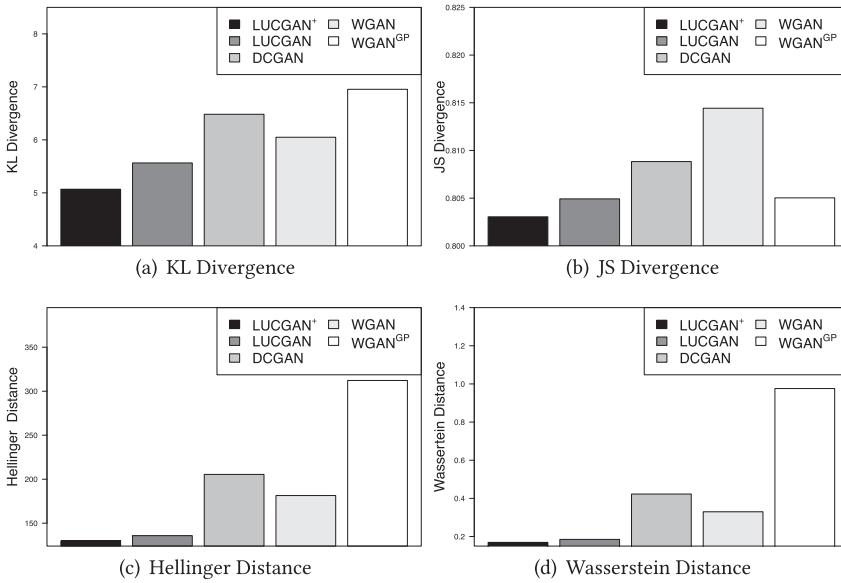
(d) Wasserstein Distance

Fig. 10. Overall performance for land-use configuration generation.

size of square is. Figure 11 shows the performance of all models when facing different square sizes in terms of KL Divergence, JS Divergence, HD, and WD. We find that with the increase of the square size, the value of all metrics decreases. A possible explanation for the observation is that when the square size is larger, the distribution of the land-use configurations becomes simpler. The generative models can capture the characteristics of the distribution of the configurations very easily, thus, the values of all metrics become smaller. However, the large square size loses much information about urban planning details of the land-use configuration. Another interesting observation is that LUCGAN+ outperforms other baseline models in terms of all evaluation metrics when $n = 100$. But for some smaller n values, LUCGAN + is slightly worse than LUCGAN. A potential reason for the observation is that LUCGAN is enough to capture the pattern of land-use configurations collected by smaller n values. Although the conditional augmentation module of LUCGAN+ can improve robustness, in this situation, such module may cause the model to a slightly underfitting. However, in reality, we should avoid collecting land-use configurations under small n values. Because such configurations lose many planning details, which is harmful to producing effective urban plans.

4.7   Study the Surrounding Contexts of Different Configurations (Q3)

Our framework generates land-use configurations based on the corresponding surrounding contexts. Thus, the surrounding contexts have strong influence on the generation of the land-use configuration. To observe the distribution of the surrounding contexts, we visualize the embeddings of the surrounding contexts on 2-dimensional space. Specifically, we first randomly choose 500 embeddings of the surrounding context of well-planned configurations and poorly-planned configurations, respectively. Then, we utilize T-SNE algorithm [37] to reduce the dimension of the embeddings into two. Next, we visualize the embeddings on 2-dimensional space, as illustrated in Figure 12. We find that the pattern of the well-planned configurations contexts is different from the pattern of the poorly-planned configurations contexts, which indicates that our research intuition, that generates the land-use configurations based on the surrounding contexts is reasonable.
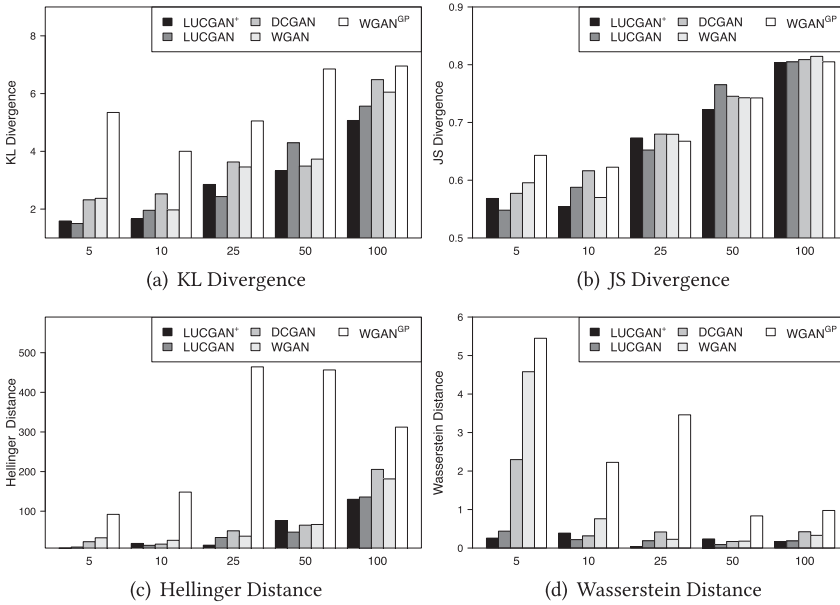
Fig. 11. The influence of square size for the generation of land-use configurations.
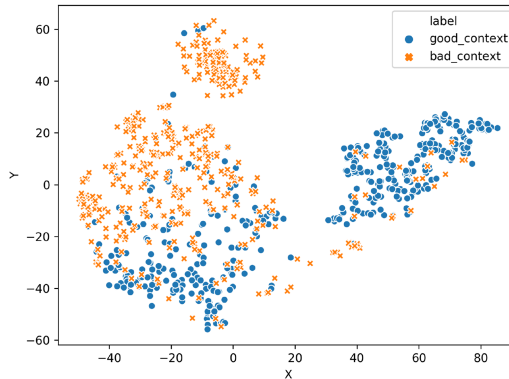

Fig. 12. Visualization for the surrounding contexts of well-planned and poorly-planned configurations.

4.8 Scoring Model Evaluation for Generated Land-Use Configurations (Q1)

To validate the effectiveness of LUCGAN$^+$ further, we build a scoring model. As illustrated in Figure 13, LUCGAN$^+$ owns the highest quality score compared with other baseline models, which indicates the superiority of LUCGAN$^+$. Meanwhile, it also shows that the scoring model can be regarded as an evaluation method for evaluating the generation of the land-use configurations.

4.9 Study the POI Ratio of Generated Configurations Under Different Q (Q4)

In our framework, we leverage a hyperparameter Q to evaluate whether a land-use configuration is well-planned or poorly-planned. When individuals have different urban planning goals, the meanings of Q are different. To validate the utility of Q, we conduct two generative tasks: (1) Q is used to determine whether a land-use configuration is vibrant; (2) Q is used to validate whether a
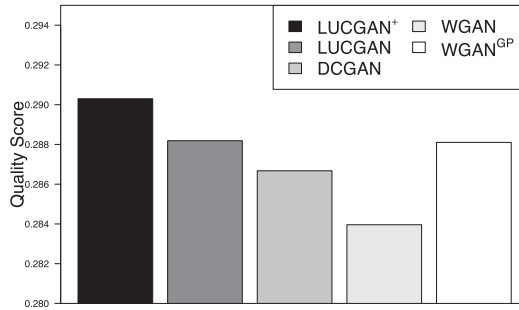
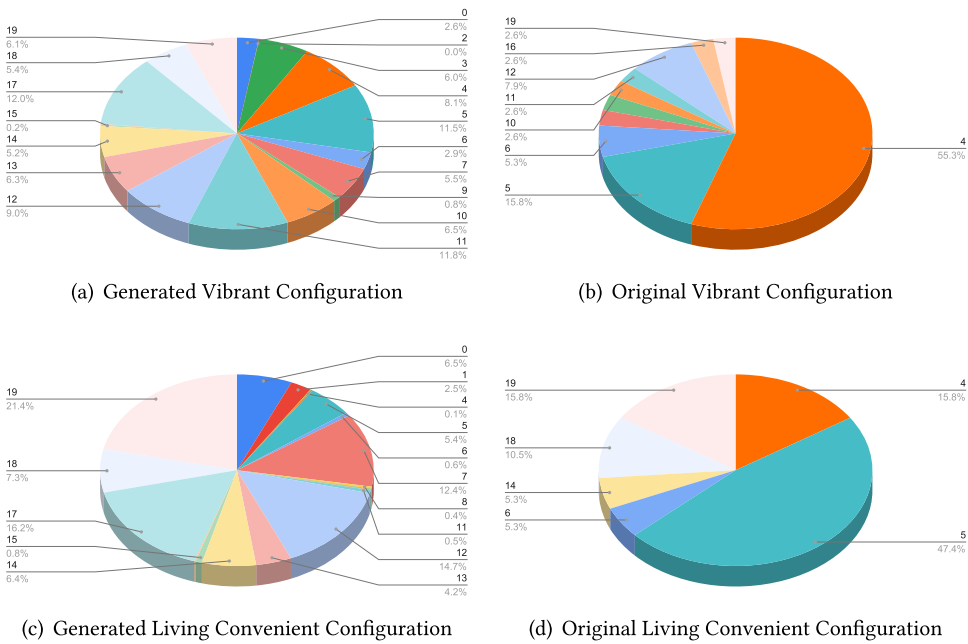Fig. 13. The quality score for different generated methods.

(a) Generated Vibrant Configuration

(b) Original Vibrant Configuration

(c) Generated Living Convenient Configuration

(d) Original Living Convenient Configuration

Fig. 14. Comparison of POI ratio between generated and original configurations under different Q.

land-use configuration is living convenient. We visualize the POI ratio of generated configurations and original configurations under the two Q settings as shown in Figure 14, in which the numbers 0 ∼ 19 denote different POI categories that are shown in Table 1, and the grey percentiles indicate the proportions of different POI categories in the corresponding configuration. Compared with Figures 14(a) and (c), we find that for the vibrant configuration, POI category 4 (food service), 5 (shopping), 7 (recreation service), and 11 (real estate) cover a large portion. This is reasonable because a vibrant configuration always owns many POIs related to economics and social activities; For the living convenient configuration, POI category 12 (government place), 17 (road furniture), and 19 (public service) occupy the majority. A reasonable explanation is that a configuration is living convenient when it contains many POIs related to public services and traffic conditions. The two observations validate that LUCGAN$^+$ can produce the customized land-use configuration utilizing Q according to people's requirements. In addition, compared with Figures 14(a), (b), (c),

(a) Generated Vibrant Configuration

(b) Original Vibrant Configuration

(c) Generated Living Convenient Configuration
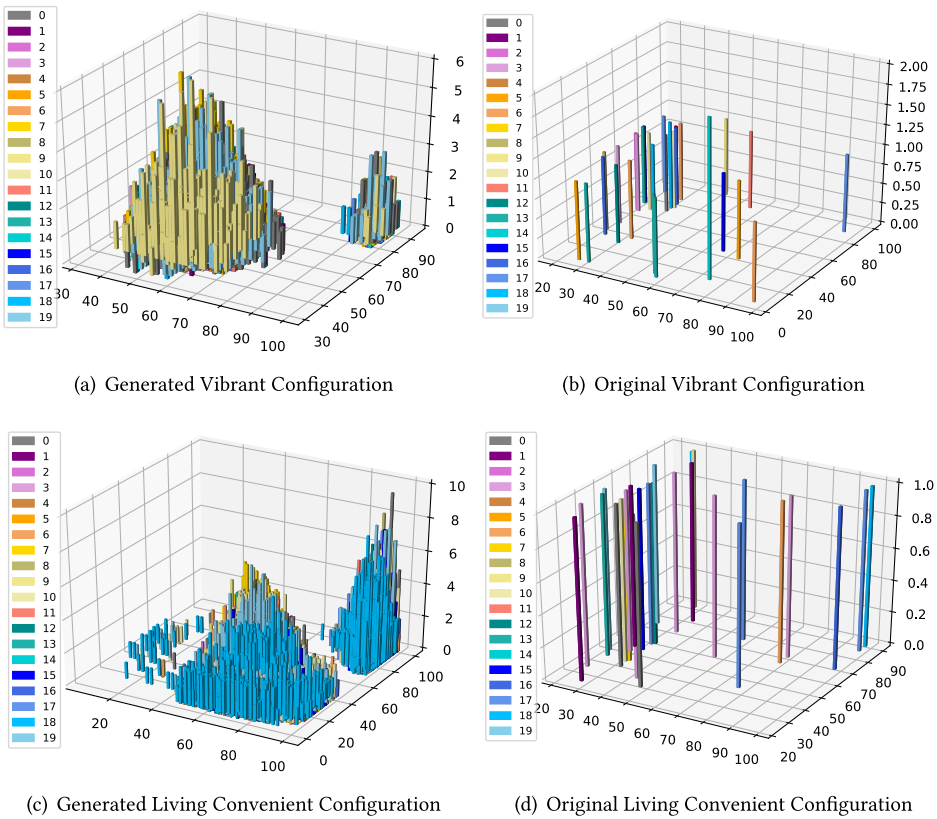
(d) Original Living Convenient Configuration

Fig. 15. Comparison of POI distribution between generated and original configurations under different Q.

and (d), another interesting observation is that the POI categories in generated configurations are more complete than the original configurations. A potential interpretation is that LUCGAN$^+$ not only captures the characteristics of the specific kind of land-use configuration but also includes new design elements into the generated configuration.

4.10 Study the POI Distribution of Generated Configurations Under Different Q (Q4)

To further understand the utility of Q and observe the differences between generated and original land-use configurations, we visualize the configurations into a 3-dimensional space as shown in Figure 15, in which the left color bar indicates the mapping relations between the number of POI categories and colors; the right part reflects the POI distribution of the configuration; the height of each bar indicates the number of POIs at the corresponding position. A careful inspection for Figure 15(a) and (c) shows that the generated configurations are organized and contain enough planning information for implementation in realistic. In addition, another interesting observation is that the generated configurations contain more dense POI distribution compared with original configurations. A potential interpretation is that LUCGAN$^+$ prefers to produce dense POI distribution, because it's easy to capture the correlation among different POIs.

4.11 Study the Generated Situation of Each Channel in Generated Configurations (Q5)

We quantify a land-use configuration as a longitude-latitude-channel tensor. So, what is the generated situation for each channel (POI category)? To check it, we visualize the POI distribution of
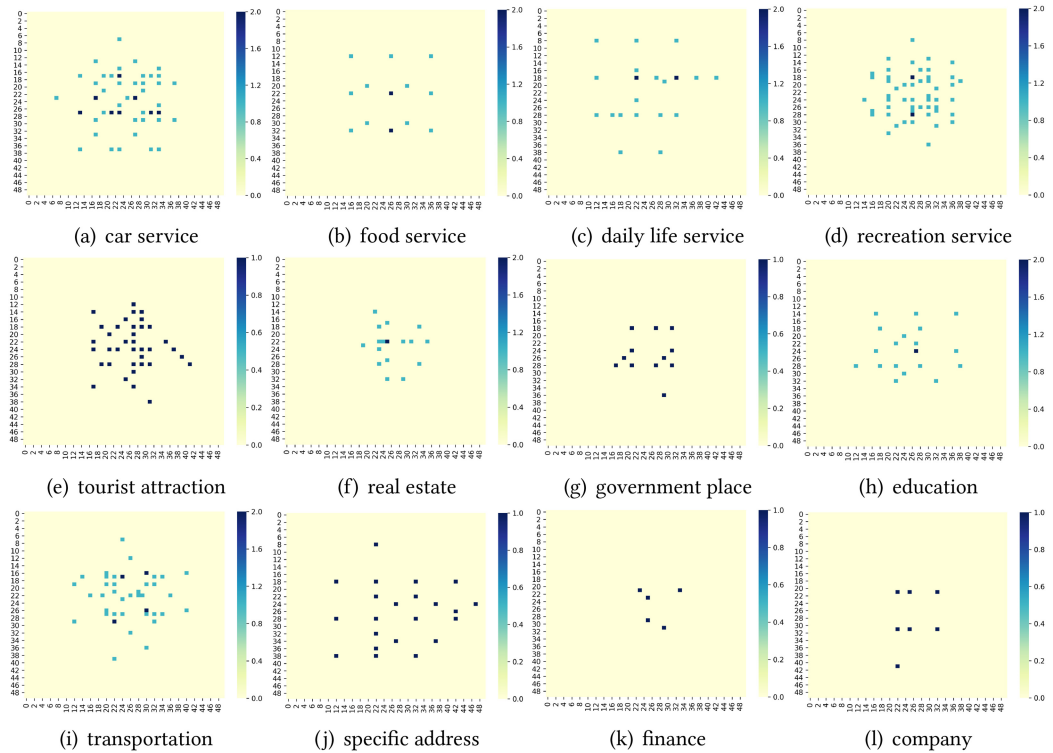
Fig. 16. Visualization for different POI categories of one generated configuration.

each channel. The visualization results are shown in Figure 16, in which the darker color of the block indicates the number of POI in the corresponding block is larger. An interesting observation standing out is that the POI distributions of different categories show their unique patterns. For example, transportation pots are more concentrated, while food service related POIs are more dispersed across the area; the distribution of car service spots is very similar to the recreation service, and the possible reason is that recreation service spots may occupy many parking lots which potentially attract car services. The observation shows that LUCGAN⁺ is capable of capturing characteristics of POI distribution of different categories at the same time. From another perspective, the observation also reflects that LUCGAN⁺ is able to capture the mutual interactions and constraints among different kinds of POIs. Thus, LUCGAN⁺ is superior and effective for generating land-use configurations automatically.

5 RELATED WORK

Spatio-Temporal Data Mining. Spatio-temporal data mining refers to the process of discovering the pattern and knowledge from the data related to space and time [5]. Owing to the spatio-temporal data is closely relevant to our real life, many researchers attempted to extract the patterns hidden behind the data for improving the urban life quality [22, 40–42, 45, 58]. For instance, Wang et al. employed deep learning approaches to forecast the travel demand of individuals based on the travel order data collected by car-hailing company [44]. Zhao et al. predicted the air quality index by considering spatio-temporal relatedness [57]. Wang et al. employed **reinforcement learning** (**RL**) and spatial knowledge graph to conduct mobile user profiling [50]. Yuan et al. utilized topic

model to discover urban functional zone based on POI data and taxi trajectories data [53]. Wang et al. used peer and temporal-aware representation learning to analyze the driving behavior based on GPS trajectory data [47]. Liu et al. studied the mobility patterns of traffic flows for bus routing optimization [23]. Du et al. provided a systematic study to capture the spatio-temporal dynamics of passenger transfers for crowdedness-aware route recommendations [11]. In this article, to reduce the heavy workload of urban planners and accelerate the urban planning process, we expect to utilize spatio-temporal data for the urban planning pattern discovering.

Representation Learning. The objective of representation learning is to preserve the information of original data into a low-dimensional feature space. In general, there are three types of representation learning models: (1) probabilistic graphical models; (2) manifold learning models; (3) auto-encoder models. The probabilistic graphical models build a complex Bayesian network system to learn the representation of uncertain knowledge buried in original data [31]. The manifold learning models infer low-dimensional manifold of original data based on neighborhood information by non-parametric approaches [59]. The auto-encoder models learn the latent representation by minimizing the reconstruction loss between original and reconstructed data [30]. In the spatio-temporal data mining domain, to capture the characteristics of spatial entity (i.e., city, geographical area), representation learning achieves great success [8, 13, 14, 46, 48]. For instance, to analyze the individual driving behaviors, Wang et al. utilized representation learning to mine the spatio-temporal characteristics of GPS trajectory data. [49]. Du et al. proposed a new spatial representation learning framework to capture the static and dynamic characteristics among the spatial entities for predicting housing price [12]. Wang et al. employed a spatio-temporal representation learning module to extract the features of cyber attack in a graph for cyber attack detection [43]. In this article, to incorporate the surrounding context characteristics into our framework, we employ representation learning to preserve the spatial attributed graphs constructed by the contexts into low-dimensional vectors.

Generative Adversarial Networks. Recently, GANs attract tremendous attention of researchers [55, 56]. GAN algorithms can be classified into three categories from the task-driven perspective. (1) Semi-supervised learning GANs. Usually, a complete labeled dataset is difficult to obtain, and the semi-supervised learning GANs can utilize unlabeled data or partially labeled data to train an excellent classifier [10, 24]. For instance, Akcay et al. designed a semi-supervised GAN anomaly detection framework that achieved good performance [2]. (2) Transfer learning GANs. Many researchers utilize the transfer learning GANs to transfer knowledge among different domains [18, 35]. For instance, Choi et al. built an unified GAN to translate the images among different style fields [9]. (3) RL GANs. RL is incorporated into GANs to improve the generative performance [34]. For instance, Ganin et al. combined reinforce learning and GAN to synthesize high-resolution images [15]. Aforementioned works indicate that GANs are capable of capturing the characteristics of the original data distribution and generate new data samples based on the distribution. Such observation motivates us to utilize the learning paradigm of GANs as the main framework of our automatic urban planner.

Urban Planning. Urbanplanning is a complex and interdisciplinary research domain [1]. Urban experts need to consider lots of factors such as government policy, environmental protection, and more for designing appropriate land-use configurations [1, 27, 29, 39]. Meanwhile, different areas have various planning goals. For example, Barton et al. focused on constructing an urban planning solution for human health and well-being [7]. John et al. discussed the relationship between urban planning and real estate development [33]. Indeed, it is difficult to generate a good urban planning solution objectively. Recently, with the development of AI, many researchers focus on making the process of urban planning become smart and automated. These methods always build up a GAN model to generate the layout of a space based on the realistic architecture

or designing image [21, 26, 28]. For instance, Albert et al. utilized GANs to generate the complex and spatial organizations observed in global urban patterns based on footprint data [3]. Bachl et al. proposed a new conditional GAN framework to learn the architecture features of major cities for generating the image of buildings, which do not exist before [6]. These works have had a lot of success, but they have one drawback: they require expert layout data in order to train AI models. In addition, some researchers use transfer learning to transfer spatial knowledge across many cities to increase the generalization of spatial AI models and learning efficiency [17, 19]. These works are capable of perceiving human mobility patterns, which gives a decent foundation for urban planning, but they are unable to immediately develop successful urban layouts. Compared to these works, our framework LUCGAN$^+$ has no strict condition for data collection. We focus on producing customized land-use configurations based on geographical spatial data such as POI, traffic data, economic data, demographic data, and so on. These data resources are always publicly available, which makes our framework have good flexibility and generalization.

6 CONCLUSION REMARKS

In order to generate a suitable and excellent land-use configuration solution objectively and reduce the heavy burden of urban planning specialists, we propose an automatic land-use configuration planner framework. This framework generates the land-use configuration based on the surrounding contexts. Specifically, we first collect a set of land-use configurations and corresponding surrounding contexts. Then, we construct spatial attributed graphs that contain explicit features such as value-added space, POI distribution, traffic conditions, and more of surrounding contexts, and preserve the information of the graphs into the surrounding embeddings. Next, we employ our proposed automatic urban planner model to generate well-planned land-use configurations based on the embeddings. Finally, through extensive experiments, we find that LUCGAN$^+$ is more effective and robust than other baseline models. In addition, different square sizes affect the generative ability of LUCGAN$^+$, so users should adopt a suitable segmentation scheme for land-use configurations based on their requirements. Moreover, LUCGAN$^+$ is capable of customizing land-use configurations based on hyperparameter Q. Furthermore, LUCGAN$^+$ not only generates the POI distribution of whole area but also provides the generation of each POI category.

REFERENCES

[1] David Adams. 1994. *Urban Planning and the Development Process*. Psychology Press.
[2] Samet Akcay, Amir Atapour-Abarghouei, and Toby P. Breckon. 2018. Ganomaly: Semi-supervised anomaly detection via adversarial training. In *Proceedings of the Asian Conference on Computer Vision*. Springer, 622–637.
[3] Adrian Albert, Emanuele Strano, Jasleen Kaur, and Marta González. 2018. Modeling urbanization patterns with generative adversarial networks. In *Proceedings of the IGARSS 2018 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2095–2098.
[4] Martin Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein generative adversarial networks. In *Proceedings of the 34th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 70)*, Doina Precup and Yee Whye Teh (Eds.), PMLR, International Convention Centre, 214–223. Retrieved from http://proceedings.mlr.press/v70/arjovsky17a.html.
[5] Gowtham Atluri, Anuj Karpatne, and Vipin Kumar. 2018. Spatio-temporal data mining: A survey of problems and methods. *ACM Computing Surveys (CSUR'18)* 51, 4 (2018), 1–41.
[6] Maximilian Bachl and Daniel C. Ferreira. 2019. City-GAN: Learning architectural styles using a custom conditional GAN architecture. arXiv:1907.05280. Retrieved from https://arxiv.org/abs/1907.05280.
[7] Hugh Barton and Catherine Tsourou. 2013. *Healthy Urban Planning*. Routledge.
[8] Dakshak Keerthi Chandra, Pengyang Wang, Jennifer Leopold, and Yanjie Fu. 2019. Collective representation learning on spatiotemporal heterogeneous information networks. In *Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. 319–328.
[9] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. 2018. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 8789–8797.

[10] Ming Ding, Jie Tang, and Jie Zhang. 2018. Semi-supervised learning on graphs with generative adversarial nets. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management.* 913–922.

[11] Bowen Du, Yifeng Cui, Yanjie Fu, Runxing Zhong, and Hui Xiong. 2018. SmartTransfer: Modeling the spatiotemporal dynamics of passenger transfers for crowdedness-aware route recommendations. *ACM Transactions on Intelligent Systems and Technology (TIST'18)* 9, 6 (2018), 1–26.

[12] J. Du, Y. Zhang, P. Wang, J. Leopold, and Y. Fu. 2019. Beyond geo-first law: Learning spatial representations via integrated autocorrelations and complementarity. In *Proceedings of the 2019 IEEE International Conference on Data Mining (ICDM).* 160–169.

[13] Yanjie Fu, Guannan Liu, Yong Ge, Pengyang Wang, Hengshu Zhu, Chunxiao Li, and Hui Xiong. 2018. Representing urban forms: A collective learning model with heterogeneous human mobility data. *IEEE Transactions on Knowledge and Data Engineering* 31, 3 (2018), 535–548.

[14] Yanjie Fu, Pengyang Wang, Jiadi Du, Le Wu, and Xiaolin Li. 2019. Efficient region embedding with multi-view spatial networks: A perspective of locality-constrained spatial autocorrelations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 906–913.

[15] Yaroslav Ganin, Tejas Kulkarni, Igor Babuschkin, S. M. Ali Eslami, and Oriol Vinyals. 2018. Synthesizing programs for images using reinforced adversarial learning. In *Proceedings of the International Conference on Machine Learning.* 1666–1675.

[16] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. 2017. Improved training of wasserstein GANs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems.* 5769–5779.

[17] Tianfu He, Jie Bao, Ruiyuan Li, Sijie Ruan, Yanhua Li, Li Song, Hui He, and Yu Zheng. 2020. What is the human mobility in a new city: Transfer mobility knowledge across cities. In *Proceedings of The Web Conference 2020* (Taipei, Taiwan) *(WWW'20).* Association for Computing Machinery, New York, NY, 1355–1365. DOI: https://doi.org/10.1145/3366423.3380210

[18] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. 2018. CyCADA: Cycle-consistent adversarial domain adaptation. In *Proceedings of the International Conference on Machine Learning.* 1989–1998.

[19] Renhe Jiang, Xuan Song, Zipei Fan, Tianqi Xia, Zhaonan Wang, Quanjun Chen, Zekun Cai, and Ryosuke Shibasaki. 2021. Transfer urban human mobility via POI embedding over multiple cities. *ACM Transactions on Data Science* 2, 1 (2021), 1–26.

[20] Thomas N. Kipf and Max Welling. 2016. Variational graph auto-encoders. arXiv:1611.07308. Retrieved from https://arxiv.org/abs/1611.07308.

[21] Jianan Li, Jimei Yang, Aaron Hertzmann, Jianming Zhang, and Tingfa Xu. 2018. LayoutGAN: Generating graphic layouts with wireframe discriminators. In *Proceedings of the International Conference on Learning Representations.*

[22] Kunpeng Liu, Pengyang Wang, Jiawei Zhang, Yanjie Fu, and Sajal K. Das. 2018. Modeling the interaction coupling of multi-view spatiotemporal contexts for destination prediction. In *Proceedings of the 2018 SIAM International Conference on Data Mining.* SIAM, 171–179.

[23] Yanchi Liu, Chuanren Liu, Nicholas Jing Yuan, Lian Duan, Yanjie Fu, Hui Xiong, Songhua Xu, and Junjie Wu. 2017. Intelligent bus routing with heterogeneous human mobility patterns. *Knowledge and Information Systems* 50, 2 (2017), 383–415.

[24] Zhiyue Liu, Jiahai Wang, and Zhiwei Liang. 2020. CatGAN: Category-aware generative adversarial networks with hierarchical evolutionary learning for category text generation. In *Proceedings of the AAAI Conference on Artificial Intelligence.* 8425–8432.

[25] Basant Maheshwari, Vijay P. Singh, and Bhadranie Thoradeniya. 2016. Balanced urban development: Is it a myth or reality? In *Proceedings of the Balanced Urban Development: Options and Strategies for Liveable Cities.* Springer, Cham, 3–13.

[26] Nelson Nauata, Kai-Hung Chang, Chin-Yi Cheng, Greg Mori, and Yasutaka Furukawa. 2020. House-gan: Relational generative adversarial networks for graph-constrained house layout generation. In *Proceedings of the European Conference on Computer Vision.* Springer, 162–177.

[27] Jari Niemelä. 1999. Ecology and urban planning. *Biodiversity & Conservation* 8, 1 (1999), 119–131.

[28] Ariel Noyman and Kent Larson. 2020. A deep image of the city: Generative urban-design visualization. In *Proceedings of the 11th Annual Symposium on Simulation for Architecture and Urban Design*, 1–8.

[29] Vitor Oliveira and Paulo Pinho. 2010. Evaluation in urban planning: Advances and prospects. *Journal of Planning Literature* 24, 4 (2010), 343–361.

[30] Samuel E. Otto and Clarence W. Rowley. 2019. Linearly recurrent autoencoder networks for learning dynamics. *SIAM Journal on Applied Dynamical Systems* 18, 1 (2019), 558–593.

[31] Yu-Ting Qiang, Yan-Wei Fu, Xiao Yu, Yan-Wen Guo, Zhi-Hua Zhou, and Leonid Sigal. 2019. Learning to generate posters of scientific papers by probabilistic graphical models. *Journal of Computer Science and Technology* 34, 1 (2019), 155–169.

[32] Alec Radford, Luke Metz, and Soumith Chintala. 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv:1511.06434. Retrieved from https://arxiv.org/abs/1511.06434.

[33] John Ratcliffe, Michael Stubbs, and Mark Shepherd. 2004. *Urban Planning and Real Estate Development.* Vol. 8. Taylor & Francis.

[34] Muhammad Sarmad, Hyunjoo Jenny Lee, and Young Min Kim. 2019. Rl-gan-net: A reinforcement learning agent controlled gan network for real-time point cloud shape completion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 5898–5907.

[35] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. 2017. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 7167–7176.

[36] Kristof Van Assche, Raoul Beunen, Martijn Duineveld, and Harro de Jong. 2013. Co-evolutions of planning and design: Risks and benefits of design perspectives in planning systems. *Planning Theory* 12, 2 (2013), 177–198.

[37] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research* 9, 11 (2008), 11.

[38] Dongjie Wang, Yanjie Fu, Pengyang Wang, Bo Huang, and Chang-Tien Lu. 2020. Reimagining city configuration: Automated urban planning via adversarial learning. In *Proceedings of the 28th International Conference on Advances in Geographic Information Systems.* 497–506.

[39] Dongjie Wang, Kunpeng Liu, Pauline Johnson, Leilei Sun, Bowen Du, and Yanjie Fu. 2021. Deep human-guided conditional variational generative modeling for automated urban planning. In *Proceeding of the IEEE International Conference on Data Mining (ICDM'21),* IEEE, 679–688.

[40] Dongjie Wang, Kunpeng Liu, David Mohaisen, Pengyang Wang, Chang-Tien Lu, and Yanjie Fu. 2021. Automated feature-topic pairing: Aligning semantic and embedding spaces in spatial representation learning. In *Proceedings of the 29th International Conference on Advances in Geographic Information Systems.* 450–453.

[41] Dongjie Wang, Kunpeng Liu, David Mohaisen, Pengyang Wang, Chang-Tien Lu, and Yanjie Fu. 2021. Towards semantically-rich spatial network representation learning via automated feature topic pairing. *Frontiers in Big Data* 4, 762899 (2021), DOI : 10.3389/fdata.2021.762899

[42] Dongjie Wang, Pengyang Wang, Kunpeng Liu, Yuanchun Zhou, Charles E. Hughes, and Yanjie Fu. 2021. Reinforced imitative graph representation learning for mobile user profiling: An adversarial training perspective. *Proceedings of the AAAI Conference on Artificial Intelligence* 35, 5 (2021), 4410–4417.

[43] D. Wang, P. Wang, J. Zhou, L. Sun, B. Du, and Y. Fu. 2020. Defending water treatment networks: Exploiting spatio-temporal effects for cyber attack detection. In *Proceedings of the 2020 IEEE International Conference on Data Mining (ICDM'20).* 32–41. DOI : https://doi.org/10.1109/ICDM50108.2020.00012

[44] D. Wang, Y. Yang, and S. Ning. 2018. DeepSTCL: A deep spatio-temporal ConvLSTM for travel demand prediction. In *Proceedings of the 2018 International Joint Conference on Neural Networks (IJCNN'18).* 1–8. DOI : https://doi.org/10.1109/IJCNN.2018.8489530

[45] Junshan Wang and Guojie Song. 2018. A deep spatial-temporal ensemble model for air quality prediction. *Neurocomputing* 314, 2018 (2018), 198–206.

[46] Pengyang Wang, Yanjie Fu, Jiawei Zhang, Xiaolin Li, and Dan Lin. 2018. Learning urban community structures: A collective embedding perspective with periodic spatial-temporal mobility graphs. *ACM Transactions on Intelligent Systems and Technology (TIST'18)* 9, 6 (2018), 1–28.

[47] Pengyang Wang, Yanjie Fu, Jiawei Zhang, Pengfei Wang, Yu Zheng, and Charu Aggarwal. 2018. You are how you drive: Peer and temporal-aware representation learning for driving behavior analysis. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining.* 2457–2466.

[48] Pengyang Wang, Yanjie Fu, Yuanchun Zhou, Kunpeng Liu, Xiaolin Li, and Kien Hua. 2020. Exploiting mutual information for substructure-aware graph representation learning. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence (IJCAI'20).* 3415–3421.

[49] Pengyang Wang, Xiaolin Li, Yu Zheng, Charu Aggarwal, and Yanjie Fu. 2019. Spatiotemporal representation learning for driving behavior analysis: A joint perspective of peer and temporal dependencies. *IEEE Transactions on Knowledge and Data Engineering* 33, 2 (2019), 728–741.

[50] Pengyang Wang, Kunpeng Liu, Lu Jiang, Xiaolin Li, and Yanjie Fu. 2020. Incremental mobile user profiling: Reinforcement learning with spatial knowledge graph for modeling event streams. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining.* 853–861.

[51] Pengyang Wang, Jiawei Zhang, Guannan Liu, Yanjie Fu, and Charu Aggarwal. 2018. Ensemble-spotting: Ranking urban vibrancy via poi embedding with multi-view spatial graphs. In *Proceedings of the 2018 SIAM International Conference on Data Mining.* SIAM, 351–359.

[52] Oren Yiftachel. 1989. Towards a new typology of urban planning theories. *Environment and Planning B: Planning and Design* 16, 1 (1989), 23–39.

[53] Nicholas Jing Yuan, Yu Zheng, Xing Xie, Yingzi Wang, Kai Zheng, and Hui Xiong. 2014. Discovering urban functional zones using latent activity trajectories. *IEEE Transactions on Knowledge and Data Engineering* 27, 3 (2014), 712–725.

[54] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N. Metaxas. 2017. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*. 5907–5915.

[55] Yingxue Zhang, Yanhua Li, Xun Zhou, Xiangnan Kong, and Jun Luo. 2019. TrafficGAN: Off-deployment traffic estimation with traffic generative adversarial networks. In *Proceedings of the 2019 IEEE International Conference on Data Mining (ICDM'19)*. IEEE, 1474–1479.

[56] Yingxue Zhang, Yanhua Li, Xun Zhou, Xiangnan Kong, and Jun Luo. 2020. Curb-GAN: Conditional urban traffic estimation through spatio-temporal generative adversarial networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 842–852.

[57] Xiangyu Zhao, Tong Xu, Yanjie Fu, Enhong Chen, and Hao Guo. 2017. Incorporating spatio-temporal smoothness for air quality inference. In *Proceedings of the 2017 IEEE International Conference on Data Mining (ICDM'17)*. IEEE, 1177–1182.

[58] Wei Zhou, Yan Yang, Yiling Zhang, Dongjie Wang, and Xiaobo Zhang. 2020. Deep flexible structured spatial–temporal model for taxi capacity prediction. *Knowledge-Based Systems* 205, 2020 (2020), 106286.

[59] Bo Zhu, Jeremiah Z. Liu, Stephen F. Cauley, Bruce R. Rosen, and Matthew S. Rosen. 2018. Image reconstruction by domain-transform manifold learning. *Nature* 555, 7697 (2018), 487–492.