Automated Urban Planning & Urban Mobility

Washif Uddin Ahmed

School of Data and Sciences

Brac University

Dhaka, Bangladesh

washif.uddin.ahmed@g.bracu.ac.bd

Abstract*- Land-use configuration design is referred to as urban planning. Effective urban design can lessen the operational and societal risks that an urban system faces, such as excessive taxes, criminal activity, traffic congestion and accidents, pollution, depressive and anxious states. Such duties are typically carried out by professional planners since urban systems are so complicated. Human planners, however, require more time. In order to do this, the automated urban planning problem is formulated as a challenge of learning to design land-uses given the surrounding spatial contexts. In order to set up the task, we define a land-use configuration as a longitude-latitude-channel tensor, with each channel representing a category of POIs and each entry representing the number of POIs. The next step is to suggest a framework for adversarial learning that can produce such tensors for an unanticipated area automatically. By learning representations from spatial graphs utilizing geographic and human mobility data, we in particular first characterize the surroundings of a nearby unplanned area. In the second step, we create tuples out of each unplanned region and its surrounding context representation, classifying each tuple into positive (well-planned areas) and negative samples (poorly-planned areas). Third, we create an adversarial land-use configuration technique, in which the representation of the surrounding context is input into a generator to produce a land-use configuration, and a discriminator learns to differentiate between positive and negative samples. Finally, we develop two new metrics to assess the quality of land-use arrangements and show detailed experiment and visualization findings to illustrate the viability of our approach. Also, we will suggest a traffic system using CityFLow in a planned urban area.

I. LITERATURE REVIEW

For this research we have taken reference of several research papers about this problem. In the papers related automated urban planning the authors explained that, the automated urban planning problem is formulated into a task of learning to configure land-uses, given the surrounding spatial contexts, and an adversarial learning framework is proposed that can automatically generate a land-use configuration for an unplanned area. In another paper from the same authors, they have developed an adversarial land-use configuration approach, where the surrounding context representation is fed into a generator to generate a land-use configuration, and a discriminator learns to distinguish among positive and negative samples. Again, in another paper they have worked on the AI by which the heavy burden of the panners, engineers alleviate and produce consistent urban plans, they wanted to ask that can AI accelerate the urban planning process, so that human planners only adjust generated configurations for specific needs. Furthermore, in a paper from Massachusetts Institute of Technology, Cambridge the authors actually proposed a method to simulate hyper-realistic urban pattern by Generative Adversarial Networks. In the other hand, for an organized planning some authors proposed BlockPlanner to tackle the problem of city block generation, accompanied by a newly collected NYC-Block Dataset in a taken research paper. However, for the traffic system of the planned urban area there are papers taken in which the authors showed the use of Eclipse Sumo. In the paper from SUMO User Conference 2019 the authors gave a model where, waiting traffic units create a pressure at the traffic control agency (TCA) and when this pressure reaches a critical value, TCA terminates the current phase and switches to another. They calculated crossing speed of pedestrians for different scenarios and compared the results having the simulation software VISSIM. By using this model, one can achieve low computational times. It is shown that, by using adaptive signal controllers, total time loss in the system can be reduced for both pedestrians and vehicles. In another paper from 2018 International Electronics Symposium, the authors discussed realistic vehicle mobility simulation based on traffic surveys of a city using a microscopic model simulation by using Simulation of Urban Mobility (SUMO). It can be known the pattern the traffic distribution. Also, in a paper named "CityFlow" they used a new method named CityFlow rather than using the most used simulator named SUMO. It is also in some extend faster in the comparison with SUMO according to them.

II. DATASET

Here, we have used assumed data of a city. The used data are following,

- Residential Community: the data is about around 3,000 people.
- POI: The POI dataset includes assumed POI records, where each POI item includes its latitude, longitude and the category. The POI information is shown in Table 1.

Table 1

Code	POI	Code	POI
0	road	10	Tourist attraction
1	Car service	11	Real estate
2	Car repair	12	Government place
3	Motorbike	13	Education
4	Food service	14	Transportation
5	Shopping	15	Finance
6	Daily life service	16	Company
7	Recreation	17	Road furniture
8	Medical service	18	Specific address
9	Lodging	19	public service

- Public Transportation.
- Check-in: The check-in records of an area of people.
- House Price: The area wise current house pricing data.

- Traffic data: vehicle and driver's maximum acceleration, road speed limit, collision free following speed, headway time following speed, intersection related speed. For some limitations here we are considering the collision free speed computation.

III. Methodology

A. Generative adversarial networks (GANs)

Unsupervised machine learning models can be trained using Generative adversarial networks, an unique paradigm that involves training two networks against one another to learn representations of the input data. According to the original concept, a generator G takes a random noise vector z as input and modifies it deterministically to produce a sample called x(fake) = G. (z). The discriminator D accepts an input x (which can be either real, from an empirical dataset, or synthetically generated by G), and outputs the source probability P(o|x) = D(x) that x is either sampled from the true distribution (o = real), or produced by G (o = fake). When G performs best, x(fake) is implicitly taken as a sample from the data distribution that G tries to mimic. Deep convolutional neural networks G and D both have weight vectors G and D that define their parameters. By alternately minimizing the following loss functions, these weights are discovered through back-propagation:

$$\theta_D : L_D = E_{x\sim px} [\log D(x)] + E_{z\sim pz} [\log(1 - D(G(z)))]$$

$$\theta_G : L_G = E_{z\sim pz} [\log (1 - D(G(z)))]$$

B. Generating Land-Use Configuration by GAN

The two types of genuine land-use layout are excellent and terrible. The automated planner's goal is to produce a top-notch land-use configuration plan based on context embedding. In order to construct the land-use configuration solution, we formally enter the context embedding into the generator. The discriminator labels the good plans as positive and the bad ones as negative in order to enhance the generating potential. The detailed information regarding the training phase is shown in Algorithm 1. In algorithm 1, we first adjust the generator module's parameters and update the discriminator module's settings. The discriminator module is then fed the top-notch, dreadful, and created land-use configuration samples. The classification outcome that was activated by the sigmoid function is then output by the discriminator. Excellent samples receive greater classification scores than poor and produced samples. The discriminator module is then fixed, and the generator module's parameter is updated. The generator receives the context embedding vectors and outputs land-use configuration solutions. The created solutions are then sent into a discriminator to determine their quality. To enhance the created ability of the generator module, we adjust the parameter. The justification output of the discriminator module serves as the source of the update gradient. When the GAN model converges, we finally have a single automatic land-use configuration planner. The discriminator can create a fantastic land-use configuration for the unplanned region if we are able to extract the context embedding.

Algorithm: Minibatch adaptive moment estimation training of automatic land-use configuration model. We adjust one hyperparameter f to change the update frequencies of the weight of the discriminator.

// start training.
for *number of training iterations* **do**

// update discriminator firstly.
for *n steps* **do**
Sample minibatch of m excellent land-use configuration samples $\{E^1, E^2, ..., E^m\}$.
Sample minibatch of m context information embedding samples $\{z^1, z^2, ..., z^m\}$.
Generate land-use configuration samples by generator, $\{F^1, F^2, ..., F^m\}$. Here, F^i= G (z^i)
Sample minibatch of m terrible land-use configuration samples $\{T^1, T^2,...,T^m\}$.
Update the discriminator by ascending its gradient:

$$\nabla\theta_d \frac{1}{m}\sum_{i=1}^{m}[log(\mathrm{D}(E^i)) + log(\mathrm{D}(1 - F^i)) + log(\mathrm{D}(1 - T^i))]$$

// update generator secondly.
Sample minibatch of m context information embedding samp1es $\{z^1, z^2,...,z^m\}$.
Update the generator using descending gradient:

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

C. Traffic Control using CityFlow

Here, the system can be designed by some parameters. Like,

1. Road network: this is actually the study of the road connection of an area. Which is actually the basic data for it.
2. Car Following Model: it is actually the study of the vehicles; their speed, acceleration, leading car etc. For the limitations here, we have used collision free data. For determining speed without collision, we have used the equation,

$$s = (-b \pm \sqrt{(b\text{^}2 - 4ac)})/2a$$

3. Intersection Logic: it is actually the intersections of the roads. To avoid the collision, vehicles should obey two rules, 1) fully stop on red lights, stop if possible on yellow lights. 2) turning vehicles should yield to straight vehicles. Here, the cross points are precomputed and when a vehicle approaches it will notify all cross points.
4. Lane Change: A means must be found for a vehicle to alert other moving vehicles when it chooses to change lanes. In this case, we employ SUMO's equivalent technique. The simulation engine will send a shadow vehicle, a copy of the moving vehicle, to the new lane when it changes lanes. In the car-following model, a shadow vehicle can take the place of a regular vehicle and perform the same duties. The simulation engine ensures consistency in the movement of the vehicle and its shadow by applying speed limitations to each other. The simulation engine will simply delete the real vehicle after the lane shift is complete and allow its shadow vehicle to take its place.

Here, the Python interface would be like this to have the count of every data,

```
import engine
eng = engine . Engine ( c o n f i g _ f i l e )
phase = [ . . . ] #the traffic signal phase of
each timestep
for step in range (3600) :
eng.set_tl_phase("intersection_1_1",
phase[step]
eng.next_step( )
eng.get_current_time( )
eng.get_lane_vehicle_count( )
eng.get_lane_waiting_vehicle_count ( )
```

```
eng.get_lane_vehicles( )
eng.get_vehicle_speed( )
#do something
```

IV. Result And Comparison

A. Evaluation and Baseline Methods for GAN

There is no set measurement because determining the quality of the urban land-use pattern is an unresolved issue. In order to demonstrate the efficacy of our framework, we analyze the quality of the planning solution generated in this study from a variety of perspectives:

1)Scoring Model: Using the great and bad land-use configuration plans as a foundation, we construct a random forest model. The algorithm can provide greater ratings for superior land-use configuration designs and lower points for bad proposals. The scoring model can be used to quantify the quality of the created land-use configuration solutions once they are developed.

2)Visualization: We choose one typical sample to visually portray from many perspectives in order to explore the generated solutions. In this method, we may instantly see the solutions. Understanding how our planner differs from other baselines is useful. Here, we are comparing our method to the following methods:

1. VAE: An algorithm using the encoder-decoder paradigm is VAE. The encoder transforms image data into latent embedding, while the decoder recovers the original data from the embedding. To discover the distribution of outstanding solutions by reducing the reconstruction loss, we used good land-use configuration in this experiment. When VAE converges, we use the decoder to provide the solution based on the context environment embedding.
2. AVG: By computing the mean value of all superb land-use plans, which reflects the average quality of all excellent examples, AVG creates the land-use configuration. However, this approach is unable to offer a tailored answer depending on various context situations.
3. MAX: By performing the max operation on all superb land-use designs, MAX provides the land-use configuration solutions. This method's output indicates the POI categories that are most prevalent in each geographic block. Similar to AVG, MAX is unable to produce a unique answer depending on various context situations.

The MAX method's superior ranking over other approaches is an intriguing phenomenon. The scoring model only accurately represents the distribution of first good plans, which is one explanation. The produced solution reflects the dominating POI categories of each geographical block, which are also inherent in the original data distribution, as the MAX technique incorporates all great plans through max operation. The MAX technique thus receives the highest rating from the scoring model. Despite coming in first, the MAX technique is not necessarily the best. This is true because the MAX approach, regardless of the context setting, only generates one type of planning solution. But this approach allows for the customization of the solutions based on various context and embedding environments. Additionally, the GAN's score is high, indicating that it well captures the fundamental principle governing the distribution of outstanding plans. To generate a land-use arrangement, the GAN is a practical and adaptable technique.

B. Evaluation and Comparison for CityFlow

We assess CityFlow's performance by contrasting it to SUMO because SUMO is an established traffic simulator with proven effectiveness. Under various traffic volume levels, we compare the average vehicle duration (the amount of time it takes for a vehicle to enter and exit the road network). The discrepancy is within a fair range, as the table illustrates.

Vehicles/Hour	100	200	300	400	500
SUMO	40.76	41.57	42.75	44.08	45.93
CityFlow	40.79	41.58	42.62	43.84	45.45
Difference	0.07%	0.04%	0.30%	0.54%	1.06%

V. Discussion And Conclusion

Here, we demonstrated in this work how current generative machine learning models, like GANs, may be utilized to accurately recreate urban patterns. Though the results are amazing, this is only the beginning, and there are still a number of significant unanswered questions. As is usual for deep learning (DL) models, the majority of them are related to the black-box nature of deep neural networks, which are currently lacking in full human interpretability and fine-tuned control. However, we think that this constraint, which most definitely merits discussion in the DL literature, shouldn't prevent further investigation into their promise to improve existing models utilizing widely accessible remote sensing data. In addition, we propose CityFlow, an efficient, multi-agent reinforcement learning environment for large scale city traffic scenarios. Researchers can use it as a testbed for traffic signal control problems and conduct research on urban mobility. We will demonstrate the usage and some results of the RL-controlled traffic signal plan. Also, we are actively developing the project and plan to support more RL scenarios like dynamic vehicle routing, policy of reversible lane or limited lane as well as open source the project in the near future.

VI. Future Works

Generative Adversarial Networks: In recent years, the study of Generative Adversarial Networks has gained popularity. From a task-driven perspective, the GAN algorithms can be divided into three types. (1) GANs with semi-supervised learning. The semi-supervised learning GANs can use unlabeled data or partially labeled data to train an outstanding classifier even if it is typically impossible to obtain a complete labeled data set. (2) GANs with transfer learning. The transfer learning GANs are widely used by researchers to transfer knowledge between various fields. (3) GANs for reinforcement learning. Reinforcement learning (RL) is implemented into generative models to enhance generative performance.

Urban Planning: Urban planning is a challenging area of study. To create proper plans for the layout of land use, experts must take into account a variety of aspects, including environmental protection and governmental policy. For instance, some scholars work to develop an urban design strategy that promotes human health and wellbeing. Several academics simultaneously design a land-use configuration strategy based on the growth of real estate. As a result, it is challenging to come up with an excellent and appropriate urban planning solution. We first suggest an automatic land-use configuration planner to address this issue, to the best of our knowledge.

References

1. Wang, D., Liu, K., Johnson, P., Sun, L., Du, B., & Fu, Y. (2021). Deep Human-guided Conditional

Variational Generative Modeling for Automated Urban Planning. 2021 IEEE International Conference on Data Mining (ICDM). https://doi.org/10.1109/icdm51629.2021.00079
2. Xu, L., Xiangli, Y., Rao, A., Zhao, N., Dai, B., Liu, Z., & Lin, D. (2021). BlockPlanner: City Block Generation with Vectorized Graph Representation. 2021 IEEE/CVF International Conference on Computer Vision (ICCV). https://doi.org/10.1109/iccv48922.2021.00503
3. Wang, D. (2021, December 26). Automated Urban Planning for Reimagining City Configuration via Adversarial Learning: Quantification, Generation, and Evaluation. https://arxiv.org/abs/2112.14699
4. Wang, D., Fu, Y., Wang, P., Huang, B., & Lu, C. T. (2020). Reimagining City Configuration. Proceedings of the 28th International Conference on Advances in Geographic Information Systems, 497–506. https://doi.org/10.1145/3397536.3422268
5. Albert, A., Strano, E., Kaur, J., & Gonzalez, M. (2018). Modeling Urbanization Patterns with Generative Adversarial Networks. IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium. https://doi.org/10.1109/igarss.2018.8518032
6. Dian Khumara, M. A., Fauziyyah, L., & Kristalina, P. (2018). Estimation of Urban Traffic State Using Simulation of Urban Mobility (SUMO) to Optimize Intelligent Transport System in Smart City. 2018 International Electronics Symposium on Engineering Technology and Applications (IES-ETA). https://doi.org/10.1109/elecsym.2018.8615508
7. Lopez, P. A., Wiessner, E., Behrisch, M., Bieker-Walz, L., Erdmann, J., Flotterod, Y. P., Hilbrich, R., Lucken, L., Rummel, J., & Wagner, P. (2018). Microscopic Traffic Simulation using SUMO. 2018 21st International Conference on Intelligent Transportation Systems (ITSC). https://doi.org/10.1109/itsc.2018.8569938
8. Akyol, G. (n.d.). Pedestrian-Friendly Traffic Signal Control Using Eclipse SUMO | Semantic Scholar. https://www.semanticscholar.org/paper/Pedestrian-Friendly-Traffic-Signal-Control-Using-Akyol-Silgu/03836970589fd42d13b64ac306e57b6423f9e2af
9. Zhang, H., Feng, S., Liu, C., Ding, Y., Zhu, Y., Zhou, Z., Zhang, W., Yu, Y., Jin, H., & Li, Z. (2019). CityFlow: A Multi-Agent Reinforcement Learning Environment for Large Scale City Traffic Scenario. The World Wide Web Conference. https://doi.org/10.1145/3308558.3314139