This is the author's accepted version of the article. The final version published by IEEE is K. Xu, R. Singh, H. Bilen, M. Fiore, M.K. Marina and Y. Wang, "CartaGenie: Context-Driven Synthesis of City-Scale Mobile Network Traffic Snapshots," IEEE PerCom 2022 - IEEE International Conference on Pervasive Computing and Communications, 2022, pp. TBD, doi: TBD.

CartaGenie: Context-Driven Synthesis of City-Scale Mobile Network Traffic Snapshots

Kai Xu[*,†], Rajkarn Singh[*,†], Hakan Bilen[†], Marco Fiore[‡], Mahesh K. Marina[†], Yue Wang[◇]
{kai.xu, r.singh, h.bilen}@ed.ac.uk, marco.fiore@imdea.org, mahesh@ed.ac.uk, yue2.wang@samsung.com
The University of Edinburgh[†], IMDEA Networks Institute[‡], Samsung[◇]

***Abstract*—Mobile network traffic data offers unprecedented opportunities for innovative studies within and beyond networking. However, progress is hindered by the very limited access that the research community at large has to the real-world mobile network data that is needed to develop and dependably test mobile traffic data-driven solutions. As a contribution to overcome this barrier, we propose CartaGenie, a generator of realistic mobile traffic snapshots at city scale. Taking a deep generative modeling approach and through a tailored conditional generator design, CartaGenie can synthesize high-fidelity and artifact-free spatial traffic snapshots using only contextual information about the target geographical region that is easily found in public repositories. Hence, CartaGenie allows researchers to create their own realistic datasets of spatial traffic from open data about their region of interest. Experiments with real-world mobile traffic measurements collected in multiple metropolitan areas show that CartaGenie can produce dependable network traffic loads for areas where no prior traffic information is available, significantly outperforming a comprehensive set of benchmarks. Moreover, tests with practical case studies demonstrate that the synthetic data generated by CartaGenie is as good as real data in supporting diverse research-oriented mobile traffic data-driven applications.**

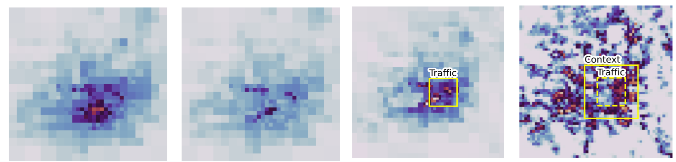
***Index Terms*—Mobile network traffic data, Traffic snapshots, Synthetic data generation, Deep generative models.**

I. Introduction

Network traffic generated by mobile devices is an unprecedented, rich source of information on human activity and movement [1], [2]. It has enabled studies on modelling mobility and migration [3], [4], inferring commuting patterns, crowding and inequities [5]–[7], monitoring demographic dynamics [8], [9], detecting functional uses of urban areas [10], monitoring road traffic [11], or planning transportation systems [12]. Networking use cases are also supported by traffic data, *e.g.*, for planning infrastructure deployment [13], improving energy efficiency [14], or optimizing novel paradigms [15].

Despite the breadth and importance of these applications, the research community currently has very limited access to mobile network traffic data. When access is granted, restrictive non-disclosure agreements (NDAs) limit the analyses to narrowly defined scopes and prevent further circulation of the data.

In this paper, we look at synthetic data generation as an effective way to lower the access barrier to mobile traffic data. In particular, we aim at synthetically generating realistic (and open) *snapshots* of mobile traffic, *i.e.*, datasets reporting the spatial distribution of the traffic generated by all users in a target geographical region where the information is aggregated according to a specific tessellation of space. This type of data,



(a) Weekday (b) Weekend (c) Traffic patch (d) Context patch

Figure 1: (a), (b) Examples of mobile traffic snapshots, also showing the spatiotemporal variation of the demand in City A from our dataset. (c) Example of a mobile traffic patch, for which the generation process is handled independently. (d) The outer square outlines the wider context patch for the traffic patch in (c), for one specific context attribute (population density).

exemplified in Figures 1a and 1b for two moments in time in the same urban area, permits multiple aforementioned applications.

Importantly, we aim at ensuring that mobile traffic snapshots can be generated for any given region from *contextual information* that is easily obtained via public repositories, such as demographics and land use databases. This allows researchers to create their own realistic datasets of spatial mobile traffic from open context data about a region of their interest.

Attaining the above goal, however, poses significant challenges. First, mobile traffic data exhibits hard-to-model characteristics such as spatial correlations and skewness [16] and traffic pattern fluctuations over time [17], as also seen in Figures 1a and 1b. Second, the available contextual attributes (interchangeably referred to as *conditions* in the rest of the paper) can only partially explain the traffic in the corresponding region, which is also driven by additional unobserved latent factors. Third, the dependence of mobile traffic on conditions is not deterministic, but inherently stochastic. Fourth, regions of interest for traffic generation (*e.g.*, cities) typically have different spatial dimensions, which gives rise to the challenge of synthesizing varied size traffic snapshots. We are unaware of prior work that address the aforementioned challenges, as more broadly discussed in §VII.

As a very first step towards solving the problem at hand, we present CartaGenie (§II), a novel method rooted in deep generative modelling for high-fidelity and generalizable synthesis of city-scale mobile traffic snapshots from contextual input. CartaGenie casts spatial traffic synthesis as a strongly conditioned generation problem, and solves it via a tailored deep convolutional neural network (CNN) architecture design.

Experiments based on multi-city mobile traffic measurements from a major European operator, augmented with contextual

[*] These authors contributed equally to this work.

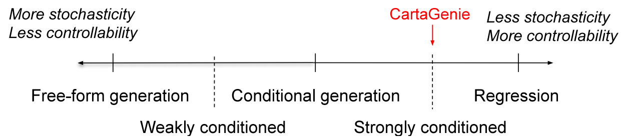


Figure 2: Spectrum of controllable data synthesis methods.

attributes from public sources (§III) lead to the following results (§IV and §V): (i) CartaGenie successfully achieves the goal of synthesizing traffic snapshots for new unseen cities solely based on contextual inputs; (ii) in doing so, it achieves significantly superior city-level fidelity compared to a range of benchmarks; and, (iii) CartaGenie-generated synthetic traffic data provides nearly indistinguishable results from those derived with real data in practical application use cases, which demonstrates the potential utility for research studies.

Upon publication of this work, we will make a synthetic mobile traffic dataset generated using CartaGenie available to the research community at https://github.com/netsys-edinburgh/CartaGenie/. Specifically, this dataset will consist of weekday/weekend daily and peak hour traffic snapshots for five diverse sized cities in Germany, obtained with context data for those cities (retrieved from public sources) as input to a pre-trained CartaGenie model. We will also strive to make CartaGenie available as a tool to the research community, subject to clearance from the project sponsor and the operator.

II. Method Design

A. Problem Statement

Let $\mathcal{X} = \{X^1, X^2, \dots, X^N\}$ be a mobile network traffic dataset that contains N sets of traffic snapshots, such that each set is collected in a different geographical region*. The data in region $n \in \{1, \dots, N\}$ includes observations over time T^n, hence $X^n = \{\boldsymbol{x}_1^n, \boldsymbol{x}_2^n, \dots, \boldsymbol{x}_{T^n}^n\}$. Each observation $\boldsymbol{x}_t^n \in \Re^{H_x^n \times W_x^n}$ is modeled as a single channel image, whose pixels values correspond to the network traffic load recorded at each location in n at time t; H_x^n and W_x^n are the height and width of the city n in pixels, which may differ between cities.

Each observation $\boldsymbol{x}_t^n$ is associated with a set of A auxiliary observations (or *conditions*), *i.e.*, attributes that may partly explain the volume of traffic generated by mobile users (*e.g.*, population distribution in the region, land use characteristics, or presence of points of interest). We denote the set of auxiliary observations for each city n as its *context*, and represent it as the set $C^n = \{\boldsymbol{c}_1^n, \boldsymbol{c}_2^n, \dots, \boldsymbol{c}_{T^n}^n\}$. Here, $\boldsymbol{c}_t^n = (\boldsymbol{v}^n, w_t)$, where $\boldsymbol{v}^n \in \Re^{A \times H_c^n \times W_c^n}$ is a multi-channel image with one channel per attribute, and H_c^n and W_c^n are the height and width of the context in pixels, respectively; whereas, w_t is a discrete time context attribute that captures the time dimension t of the $\boldsymbol{c}_t^n$.

Our goal is to design a model to *generate* synthetic network traffic data $\boldsymbol{s}_t^m \in \Re^{H_s^m \times W_s^m}$ for a previously *unseen* region m at a specified time* t and given auxiliary observations $\boldsymbol{c}_t^m$, in a way that synthetic samples $\boldsymbol{s}_t^m$ exhibit similar characteristics as the real training data $\mathcal{X}$ and are compatible with input $\boldsymbol{c}_t^m$.

*In this work, we will consider whole cities as the regions of interest, and we will use the terms region and city interchangeably. However, the approach we propose is general, and applies to other definitions of region.

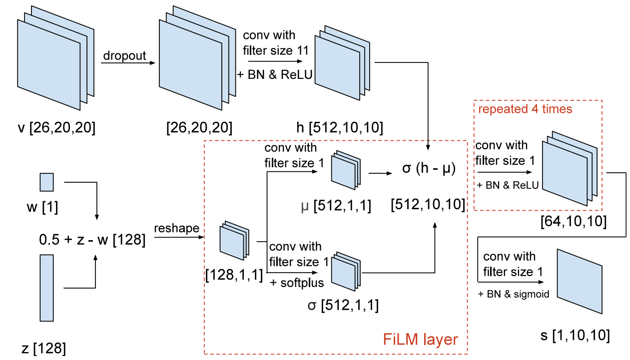


Figure 3: Schematic of CartaGenie neural network architecture. Tensor shapes are annotated in brackets, *e.g.*, [26, 20, 20] means a 3-dimensional tensor with its corresponding dimension sizes.

B. Proposed Solution: CartaGenie

The design of CartaGenie stems from the consideration that the available conditions can only partially explain the data to be generated, and there are unexplained parts, either due to missing conditions or stochasticity, that we want our synthesizer to model. From a data generation perspective, and as shown in Figure 2, this calls for a *strongly conditioned* model. We argue that alternative approaches are not well suited for the problem of mobile traffic synthesis. On the one hand, the complex and partly stochastic nature of the data $\mathcal{X}$ (used for training) cannot be fully explained by conditions $\mathcal{C}$: hence, classical regression that lies at the right end of the spectrum performs poorly, as we also experimentally show in §IV. On the other hand, the free-form generation approach that ignores conditions and tries to generate real data distributions from only noise input lacks controllability and discards valuable information: in §IV we show that popular methods for 'unconditioned' generation at the left end of the spectrum, such as Generative Adversarial Networks (GANs) [18] and Variational Autoencoders (VAEs) [19], [20], also yield lower traffic synthesis quality than CartaGenie.

CartaGenie is a conditional deep generator $\mathcal{G}_\theta : \boldsymbol{c}, z \rightarrow \boldsymbol{x}$, with a tailored Convolutional Neural Network (CNN) architecture, as depicted in Figure 3. It takes conditions $\boldsymbol{c}$ and a latent variable z as input. Specifically, $\boldsymbol{c}$ is a combination of spatial ($\boldsymbol{v}$) and non-spatial (*i.e.*, temporal, w) data; the time context attribute w allows generating traffic snapshots for specified time periods with different traffic patterns (*e.g.*, compare Figures 1a and 1b). On the other hand, the latent (or noise) variable $z \in \Re^{D_z}$ captures unobserved conditions beyond $\boldsymbol{c}$ and stochasticity, and is also non-spatial.

The non-spatial inputs z and w are processed via a specialized FiLM conditioning layer [21]. As detailed in §II-C, this

*We will focus our evaluation on two relevant definitions of time, *i.e.*, daily aggregates and peak traffic hours. Our model, however, is capable of generating spatial mobile traffic snapshots at any given time granularity (weeks, days, hours, etc.), only limited by the time-resolution of the training data.

design avoids naive conditioning on the latent variable (*i.e.*, simply concatenating $\boldsymbol{c}$ and z), which weakens the conditioning and is known to result in the neural network completely ignoring stochasticity [22], [23]; our approach also provides an elegant way of integrating the time context as a noise modulator. The result of the separate convolutions along spatial and non-spatial inputs are then merged via an affine transformation into a hidden representation, which is then processed by stacked convolution layers with size-1 kernels and batch normalization (BN) layers in between, to produce the final sample $\boldsymbol{s}$.

To allow generation of mobile traffic snapshots for cities with varying spatial dimensions via a single generator model, CartaGenie operates at a smaller *patch* level, an example of which is in Figure 1c. Doing so has additional advantages in terms of efficient training and avoiding overfitting, as elaborated in the next section, although care needs to be taken to avoid artifacts when sewing up generated patches to make up the city level traffic map. To this end, CartaGenie ensures for each traffic patch the corresponding context patch is sufficiently wider, as illustrated in Figure 1d, to enable conditionally independent pixel level generation; artifact-free generation then naturally results from the Markov blanket property [24] (§II-D).

The learning process during model training is via minimizing the expected reconstruction error between the real and synthesized traffic maps, computed as the L1 distance, *i.e.*, the sum of absolute differences in corresponding individual pixel values of the two maps. The model optimization is through stochastic gradient descent, using mini-batches over different city, time and patches. We provide a more detailed description of the model architecture and training in a companion technical report [25].

C. Latent Variable Modeling with Time Input via FiLM Layer

How the latent variable (noise vector) z is integrated is an important design choice for the conditional generator. If we simply concatenate it (after reshaping) to the conditions $\boldsymbol{c}$, the generator tends to ignore the noise vector during training and leads to deterministic outputs given $\boldsymbol{c}$ [22], [23]. Proposals to instead use dropout layers as a source of stochasticity were found not effective either [26]. In CartaGenie, we address the problem by using the feature-wise linear modulation (FiLM) conditioning layer [21]. When adapted to our case, the FiLM layer takes two inputs $\boldsymbol{h} \in \Re^{N_h \times W_h \times H_h}$ and $z \in \Re^{N_z \times 1 \times 1}$, where $\boldsymbol{h}$ is an intermediate hidden representation of the spatial context $\boldsymbol{v}$ that is encoded with our model. z is transformed using a layer of neural network to $\boldsymbol{\mu} \in \Re^{N_h \times 1 \times 1}$ (without activation) and $\boldsymbol{\sigma} \in \Re^{N_h \times 1 \times 1}$ (with the softplus $f(x) = \log(1 + \exp(x))$ activation to ensure $\boldsymbol{\sigma}$ is positive). Then, an affine transformation is performed to $\boldsymbol{h}$ as $\boldsymbol{h} \leftarrow \boldsymbol{\sigma}(\boldsymbol{h} - \boldsymbol{\mu})$. Notice that all individual pixels for each channel (corresponding to each spatial context attribute) are transformed using the same shift and scale, which means our z effectively models global variations.

Another issue concerns handling of time w, which is non-spatial and is thus the same for all pixels of the traffic and spatial context. We address this issue by using w to modulate the latent variable z into a W-modal distribution, where W denotes the number of values that w can take. For the particular case when w is a binary variable (*e.g.*, week day or weekend day), we can use the transformation $z \leftarrow 0.5 + z - w$; when z is a *unimodal* distribution, this effectively makes the input to the generator a *bi-modal* distribution so that the model can learn different behavior depending on which of the two values w takes (*e.g.*, weekday traffic pattern vs. weekend traffic pattern). In general, though, w can take any number of discrete values, allowing our model to generate traffic snapshots at any desired time granularity.

D. Artifact-Free Patch-Level Traffic Snapshot Generation

In CartaGenie, we perform model training and traffic generation at smaller patch level. For training, to create the samples, we divide the $H_x^n \times W_x^n$ pixels of the traffic data $\boldsymbol{x}_t^n$ of each city n in the training data into smaller *traffic patches* $\boldsymbol{x}_t^{n,l}$, $l \in \{1, \dots, L\}$. Figure 1c illustrates one sample patch in one of the cities from our dataset. This approach has several advantages. Firstly, different cities have diverse geographical span, hence their traffic maps have varied spatial dimensions. Using fixed smaller sized traffic patches as we do allows using the same generator model architecture regardless of the city dimensions considered for training or generation. Secondly, it allows using diverse traffic patches from different snapshots together to enable a more efficient training via stochastic gradient optimization. Moreover, different local sub-regions of a same city can have similar relationship between traffic and spatial conditions: training at the level of patches can be then seen as a form of *weight-sharing* – a type of *regularization* technique – to enforce the model to learn the actual casual relationship between $\boldsymbol{c}$ and $\boldsymbol{x}$ instead of memorizing the mapping. In our experiments, we set $\boldsymbol{x}_t^{n,l} \in \Re^{N_{\boldsymbol{x}} \times N_{\boldsymbol{x}}}$ with $N_{\boldsymbol{x}}$ = 10, *i.e.*, a 0.25×0.25 km^2 area covered by each patch. Consistently with the patch level real training data, CartaGenie outputs synthetic data for each traffic patch in the target city map, *i.e.*, $\boldsymbol{s}_t^{n,l} \in \Re^{10 \times 10}$.

While patch-level synthetic traffic data can be accurate locally, it has a potential downside when used to recreate city-level traffic maps, as artifacts may appear at the boundary of patches that are separately generated. To address this concern, we take a *conditionally independent pixel-level generation* approach. This may seem counter-intuitive given that we expect pixels (especially the nearby ones) in a generated traffic map to be correlated. Our approach, however, is in fact consistent with this expectation and is rooted in the property of Markov blanket [24], which in our setting translates to: if the context patch corresponding to a traffic patch is large enough to cover the Markov blanket of any pair of nearby traffic pixels, they would be *conditionally independent* even when the marginal distribution of these traffic pixels may be correlated. In short, we assume that only a local region of condition pixels have effect on traffic pixels, a reasonable assumption for our task.

In other words, when considering a traffic patch $\boldsymbol{x}_t^{n,l}$, only the portion of the city-wide spatial context $\boldsymbol{v}^n$ that is in the geographical proximity of the patch stays relevant to the learning process. We thus associate to each $\boldsymbol{x}_t^{n,l}$ a trimmed spatial context (which we call context patch) $\boldsymbol{v}^{n,l}$ (spanning all spatial context attributes) that includes a margin around the traffic patch, as exemplified in Figure 1d. As outlined above, this margin is required to ensure that the context surrounding pixels at the border of $\boldsymbol{x}_t^{n,l}$ is properly considered during the learning process. Clearly, the number of context patches is the same as that of traffic patches, *i.e.*, L, and we denote by $\boldsymbol{c}_t^{n,l} = (\boldsymbol{v}^{n,l}, w_t)$ the complete spatiotemporal context corresponding to $\boldsymbol{x}_t^{n,l}$. In our experiments, we set $\boldsymbol{v}^{n,l} \in \Re^{N_{\boldsymbol{c}} \times N_{\boldsymbol{c}}}$ with $N_{\boldsymbol{c}}$=20.

To implement our above described conditionally independent pixel generation idea, we need to ensure that: (1) for each traffic patch, a corresponding but larger context patch is used; and (2) the neural network generates each pixel independently given the context patch. Therefore, in our architecture, we have the first convolution layer to map the context into the same shape of traffic patches by using a kernel size of $N_{\boldsymbol{c}} - N_{\boldsymbol{x}} + 1 = 11$. After the following FiLM layer, all convolutions layers have a kernel size of 1, which independently process the intermediate "hidden images" and output the synthetic traffic map.

III. Evaluation Methodology

Here we outline our evaluation methodology, elaborating on the mobile traffic & context data, metrics and baseline methods.

A. Data

1) Mobile Traffic Dataset: Our real-world mobile traffic data was provided by a major mobile network operator. The dataset covers 4 large cities in one country (indicated as Cities A, B, C, and D), and consists in measurements of the mobile traffic load generated by the whole subscriber base of the operator (which has a 30% market ratio in the country), for 47 days. Each city is divided into grid cells (pixels), each of size 250×250 m^2. Different cities have different surface areas, resulting in 80×80 cells for both City A and City B, 128×96 cells for City C and 96×80 cells for City D. For each cell, the traffic generated by all local users is aggregated over fixed time intervals so as to generate snapshots such as those in Figures 1a and 1b. By default, we use 24-hour intervals for data aggregation, which allows capturing the average daily load in the pixel; however, our approach is agnostic to the aggregation interval duration, and we also study 'peak hour' traffic snapshot generation to demonstrate that our model can support different time granularities.

2) Datasets for Conditional Attributes: We condition the synthetic traffic generation on contextual attributes. We specifically select attributes that are commonly available from public sources, so as to make the approach as reusable as possible.

Population. This attribute represents the number of residents in each grid cell, as reported in national census. By comparing the heatmap of this attribute in Figure 4 with Figure 1a, we note that population correlates with traffic in non-obvious ways.

Land use. These attributes represent the incidence of specific types of land use in each grid cell. Land uses capture the utilization of the territory, and highlight zones of the target region with different purposes. We extract information on the land use of four cities from the Copernicus Urban Atlas 2012 repository [27], and transform each land use type to an attribute map where the value associated to a grid cell represents the fraction of the cell covered by that land use

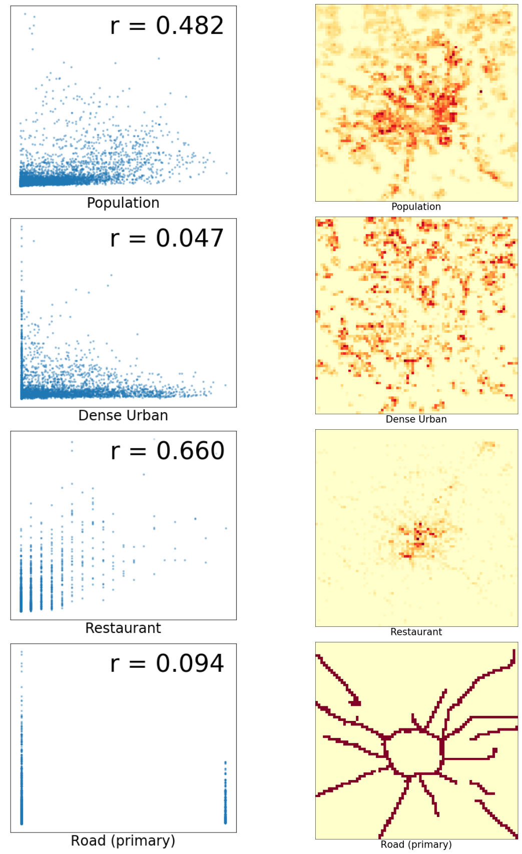


Figure 4: Scatterplots (left column): mobile traffic (y-axis) versus attribute values (x-axis) for four conditional attributes in all City A grid cells, with the resulting correlation coefficient. Heatmaps (right column): geographical distribution of values of conditional attributes, providing a visual comparison against the mobile traffic observed in City A (Figure 1a).

type. Copernicus specifies a large number of land use types, many of which yield no relevance for mobile traffic. In order to avoid feeding CartaGenie with uninformative input, we filter out land uses with near-zero Pearson's correlation coefficient with respect to the mobile traffic, while aggregating others that are semantically similar. Finally, we retain 11 land use types: (i) Continuous Urban Fabric, (ii) Discontinuous Dense Urban Fabric, (iii) Discontinuous Medium Density Urban Fabric, (iv) Discontinuous Low Density Urban Fabric, (v) Discontinuous Very Low Density Urban Fabric, (vi) Isolated Structures, (vii) Green Urban Areas, ($viii$) Industrial and Commercial Areas, (ix) Air and Sea Ports (x) Leisure Facilities, and (xi) Non-urban Barren Lands. Figure 4 shows attribute maps for a subset of these land use types for City A.

Points of Interest (PoIs). PoIs indicate the presence of specific categories of landmarks, and are complementary to population and land use. We obtain PoI data from the OpenStreetMap (OSM) initiative using the Overpass API [28], and transform each PoI category into an attribute map by counting the number of PoIs in the target category that are located in every grid cell. Similarly to what we did for land uses, we filter out the tens of insignificant PoI categories in the original data using a correlation analysis with traffic. This results in retaining 14 PoI categories: (i) Tourism, (ii) Cafe, (iii) Parking, (iv) Restaurant, (v) Post Office and Police Station, (vi) Traffic signals, (vii) Office, $(viii)$ Public transport, (ix) Shop, (x) Secondary roads, (xi) Primary roads, (xii) Motorways, and $(xiii)$ Railway stations, and (xiv) Tram stops. Figure 4 illustrates attribute maps for two selected PoIs for City A.

Time. Mobile data traffic is inherently time-varying, and different geographical distributions of the demand for mobile services can be observed at different times. This suggests conditioning the generation of synthetic data on a time attribute. When considering daily traffic aggregates, we use a binary time condition, which differentiates working days (Monday to Friday) from weekend days (Saturday and Sunday), as outlined in §II-C. We additionally present results that focus on synthesizing peak hour traffic snapshots.

As described above, CartaGenie uses a total of 26 different spatial conditions and 1 condition for time. An important remark is that none of the attributes is correlated in a straightforward way with the mobile traffic, as exemplified by Figure 4 and extensively presented in [25]. This suggests that no naive statistical model can be easily derived from any of these individual attributes nor even an elaborate regression model combining these attributes to produce a reliable traffic estimate from the attributes, which justifies the sophisticated and tailored approach we take in designing CartaGenie.

B. Evaluation Criteria

1) Quantitative fidelity metrics: Since we represent traffic snapshots as images, image quality measures are an obvious choice for evaluating the fidelity of our synthetic traffic data. Specifically, we use two well established image quality assessment measures: Peak Signal-To-Noise Ratio (PSNR) and Structural Similarity Index Metric (SSIM). The metrics $PSNR(x, s)$ and $SSIM(x, s)$ measure the similarity between a pair of real and synthetic traffic images x and s.

In fact, in our setup, we have *sets* of traffic snapshots for both real (*e.g.*, from multiple days) and synthetic (*e.g.*, from multiple runs of CartaGenie) data. Hence, x and s are samples of two distributions $p(x)$ and $p(s)$. Quantifying the extent to which these two distributions are similar is key to assessing the fidelity of CartaGenie. To this end, we forward the comparison problem to the optimal transport (OT) regime [29] by using PSNR and SSIM as cost functions, and use the resulting optimally-transported OT-SSIM and OT-PSNR metrics to measure the fidelity of synthetic traffic with respect to the real data. Intuitively, these OT variants find a binary pairing matrix between two sets for which the sum of PSNR or SSIM is maximized. OT is widely used to characterize the distance between two distributions given finite samples [29].

2) Qualitative criteria: We provide visualizations and statistics as a more intuitive way to assess the fidelity of the synthetic data. Besides synthetic traffic maps, these qualitative metrics include: (a) time series of daily total traffic; (b) traffic histograms at pixel level; and (c) histograms of total traffic at day level.

C. Benchmarks

We consider a range of benchmarks for comparative assessment of CartaGenie, as outlined below and justified in §VI.

Pix2Pix. As we represent $\boldsymbol{c}$ and $\boldsymbol{x}$ as images with different number of channels, we use the powerful *Pix2Pix* image-to-image translation framework [22] as a representative state-of-the-art approach. It has been successfully used in computer vision tasks that are similar to our traffic generation problem. These models are both conditional and stochastic, and provide a deep network architecture tailored for image-to-image translation. We use the U-Net architecture [26] without one hidden layer in the encoder and decoder to adapt it to 10×10 images.

MLP and CNN Regression. A naive approach to solve the traffic snapshot generation task is to train a model that takes conditions c as input and predicts x. Common choices for such a regression task are multi-layer perception (*MLP*) neural networks and convolution neural networks (*CNN*). These methods can be viewed as sophisticated variants of state-of-the-art mobile traffic generation approaches [13]. However, neither of them can model stochasticity, thus we expect them not to be able to characterize important features of the data. Comparing MLP and CNN regression also serves to show that convolutions provide better modeling of spatial correlations.

Generative adversarial networks (GANs). We also evaluate GANs on the traffic map generation task. In comparison to the regression baseline, GANs model stochasticity by randomly sampling latent vectors [18]. However, they aim at a free-form generation, hence their outputs cannot be conditioned on a context $\boldsymbol{c}$. GANs let us show that conditions are indeed required to synthesize high-fidelity traffic maps. We use the standard DCGAN architecture [30] for an image size of 10×10.

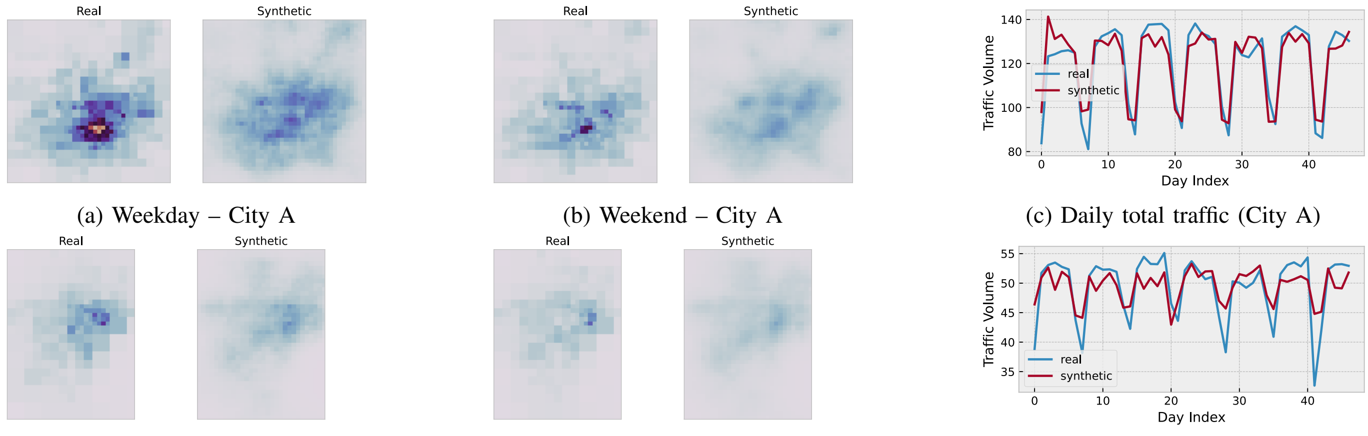
Conditional GAN and Conditional VAE. We also consider the conditional variant of GANs [31] as well as conditional VAEs [32] among our benchmarks. As these two methods can be easily added to CartaGenie by adapting the training process and modifying the loss function, we consider them as part of the ablation study of CartaGenie and report the results summary (due to page limit).

IV. Results

This section presents results assessing the fidelity and generalizability of CartaGenie relative to benchmark methods (§III-C). To this end, we use a 'leave-one-city-out' evaluation: we use 3 cities for training and 1 for testing, and repeat this for 4 times with a different test city each time, giving 4 results.

A. Generalization to unseen cities

Here we highlight the ability of CartaGenie to generalize to unseen cities by training on three cities in the dataset and generating traffic snapshots for the remaining city. We start with qualitative results with CartaGenie in Figure 5 for two arbitrarily picked test cities – City A and City B– due to space limitation. In this figure, there are three sub-figures for each city. The first two sub-figures, respectively, show synthesized traffic maps with CartaGenie averaged over weekdays and weekends. These maps show that CartaGenie successfully captures the



(a) Weekday – City A (b) Weekend – City A (c) Daily total traffic (City A)

(d) Weekday – City B (e) Weekend – City B (f) Daily total traffic (City B)

Figure 5: Generalization results using CartaGenie with City A and City B as test cities.

Table I: Quantitative fidelity results using CartaGenie with different test cities.

	Condition	OT-PSNR↑	OT-SSIM↑
City A	Weekday	24.73	0.651
	Weekend	28.28	0.721
City B	Weekday	32.00	0.797
	Weekend	32.52	0.819
City C	Weekday	28.49	0.783
	Weekend	29.31	0.801
City D	Weekday	29.68	0.806
	Weekend	30.41	0.809

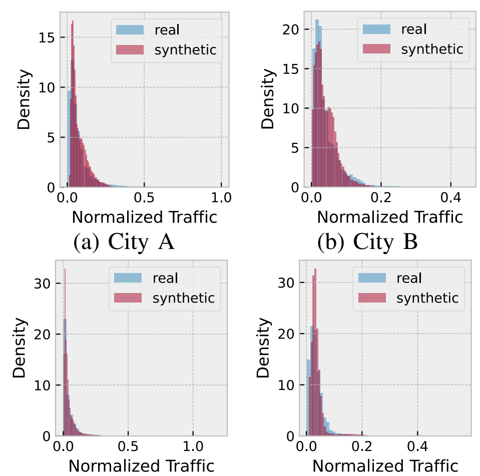
Table II: Quantitative comparisons between CartaGenie and benchmarks. Numbers are average of 4 runs using 3 cities for training and 1 for testing.

Metric	Condition	Type	GAN	MLP	CNN	Pix2Pix	CartaGenie
OT-PSNR↑	Weekday	Train	20.99	27.61	**32.88**	31.04	32.41
		Test	21.14	25.63	28.61	26.02	**28.72**
	Weekend	Train	22.61	28.09	**33.13**	31.21	32.38
		Test	21.81	27.30	**30.05**	27.43	**30.05**
OT-SSIM↑	Weekday	Train	0.43	0.72	0.85	0.82	**0.86**
		Test	0.44	0.62	**0.76**	0.61	**0.76**
	Weekend	Train	0.45	0.71	**0.85**	0.81	0.84
		Test	0.39	0.62	0.78	0.62	**0.79**

overall spatial pattern as well as differences in traffic patterns over time (weekdays versus weekend days). We also provide the total traffic per day over the entire city as a time series plot for these two test cities (last sub-figure); these results further show that CartaGenie successfuly captures traffic variations across weekdays and weekends. We obtained similar results with City C and City D as test cities.

We summarize the quantitative fidelity results in terms of OT-PSNR and OT-SSIM for each test city in Table I. We see that in most of the cases results are over the desired levels – 25 in terms of OT-PSNR and around 0.8 for OT-SSIM [33]. Results are in the borderline for City A, which has markedly different traffic patterns as reflected in Figure 5a. Specifically, the high traffic level in the hotspot area in City A is much larger than others. For this reason, the model fails to capture or generate it as such pattern has not been seen in other cities during training; this issue can be resolved if there is access to more diverse training data. Also note that the performance for weekends is consistently better than that of weekdays, likely because there are less variations during weekends.

We now take a closer look at (mis-)matches in the distribution between real and CartaGenie-generated synthetic traffic. Traffic histograms at pixel level with CartaGenie show close alignment with real data for all test cities, as shown in Figure 6, indicating that CartaGenie can accurately capture the statistics of the mobile traffic distribution at city scale. Note that for easier readability, traffic here is shown in normalized scale (with respect to global pixel level traffic volume). Figure 7 shows histograms of total traffic at city level. We observe that



(a) City A (b) City B

(c) City C (d) City D

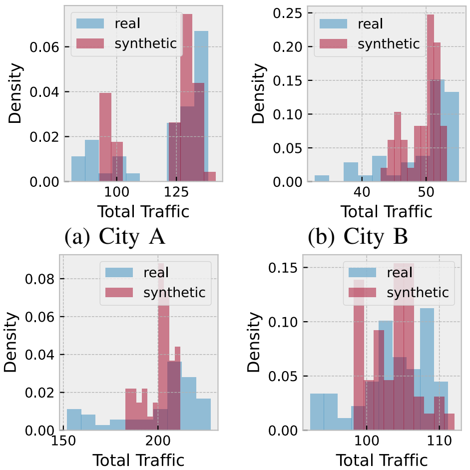
Figure 6: Traffic histogram at pixel level with CartaGenie.

CartaGenie successfully captures that there are two modes in total traffic corresponding to weekday and weekend patterns. The variation of each mode in the synthetic data is also close to that of real data, except in one case (City C).

B. *Comparison with benchmark methods*

We now compare the fidelity and generalization performance of CartaGenie against the four benchmarks, namely MLP and CNN based regression, (unconditional) GAN, and Pix2Pix.

1) Qualitative comparisons: Figures 8, 9, 10 show the similar set of qualitative results to the previous section when using



(a) City A (b) City B (c) City C (d) City D

Figure 7: Histogram of total traffic with CartaGenie.

City A as the held-out (*i.e.*, test) city*. As it can be seen from Figure 8, the generated city maps from GAN, MLP regression and Pix2Pix are visually degraded compared to that with CartaGenie (see Figures 5a and 5b). For GAN, as the model is an unconditional generator, we do not expect it to be able to generate the desired city map. For MLP regression, the use of MLP fails to capture spatial correlation well. The poor performance of Pix2Pix might be surprising as it is the state-of-the-art image translation model. However, as previously discussed, our tasks differs from image-to-image translation as the conditions serve a different role in the data generation process. Especially for pixels around the edges, for which the context is actually missing in image-to-image translation framework. Lastly, the relatively good visual performance of CNN regression is not surprising as the CNN regression based architecture differs from CartaGenie in two main ways: (i) it lacks the latent variable z to model stochasticity; (ii) it does not obey the requirement for conditionally independent pixel level generation (as discussed in §II-D). For (i), we will see next that this approach fails to altogether capture the variation in the data completely. For (ii), we can see that there are visible artifacts of fine vertical and horizontal mosaic in the city map (clearer when zoomed in).

We now look at Figure 9 and Figure 10, which show how different benchmark methods capture the distribution aspects of the data. Despite the poor visual results in Figure 8, GAN performs reasonably well in capturing the data distribution, especially in Figure 9. MLP regression and CNN regression also capture pixel-level distribution to some extent – especially for CNN regression, the performance is close to ours. However, as shown by Figure 10, they fail to capture variation in time, as by definition these approaches do not consider modeling inherent stochasticity. Finally, although Pix2Pix takes the conditional GAN approach which ideally should also model the distribution, the use of dropout as a source of stochasticity leads to degenerated performance in terms of variation in time, which has also been observed in previous studies [22], [23], [26]. Although this is not a major issue in the context of image translation, correctly modelling stochasticity is important in our setting. Crucially, correctly modeling variation caused by time paves the way for CartaGenie to be used as a component in a spatiotemporal model to control the dynamics of $\boldsymbol{x}$ via the dynamics of z. Pix2Pix results additionally suffer from severe edge effects (Figure 8).

*We omit results for other cities as they yield similar observations in terms of qualitative differences with and among benchmark approaches.

Table III: Quantitative fidelity results using CartaGenie for *peak hour* traffic snapshot generation with different test cities.

	Condition	OT-PSNR↑	OT-SSIM↑
City A	Weekday	23.85	0.660
	Weekend	29.75	0.786
City B	Weekday	32.70	0.818
	Weekend	35.75	0.872
City C	Weekday	28.16	0.806
	Weekend	31.53	0.840
City D	Weekday	30.43	0.824
	Weekend	31.78	0.820

2) Quantitative comparisons: Here we present results for the quantitative fidelity metrics. We report the average of 4 results, each corresponding to a different test city. We report both the training and testing performance, which, respectively, show how well each model fits the data, and how well each model generalizes to unseen cities. Table II summarizes these performance results in terms of OT-PSNR and OT-SSIM; best performing method in each case is highlighted in bold. Overall, CartaGenie generally outperforms the benchmarks by a clear margin, especially during test. The only exception is CNN regression, which performs similarly to CartaGenie but fails to model data stochasticity in data as per Figure 10c.

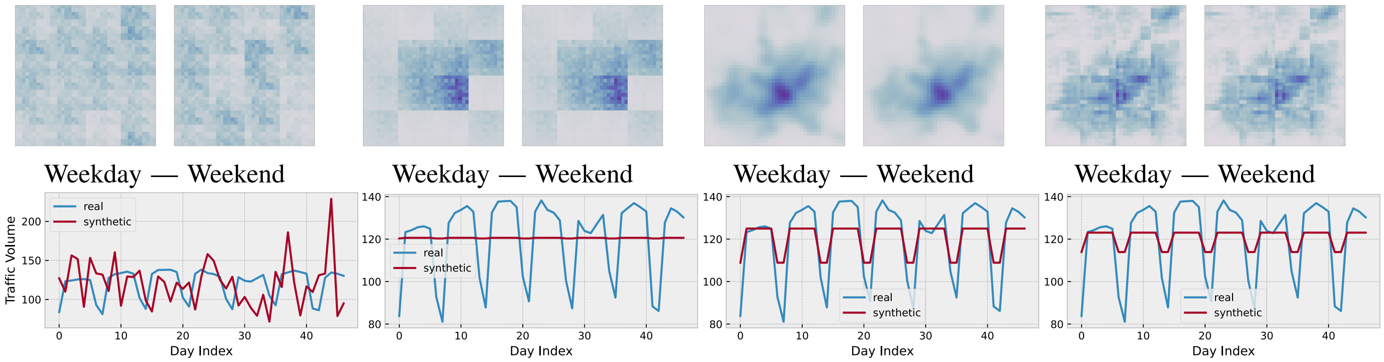
C. Additional Results

1) Peak hour snapshots: We demonstrate how CartaGenie is agnostic to time granularity by generating *peak hour* traffic snapshots. By analyzing the traffic dynamics at an hourly level for each of the four cities in our dataset, we find that traffic typically hits a peak at 12-1pm. We train CartaGenie to generate traffic snapshots for that peak hour. Table III shows quantitative fidelity results using CartaGenie to synthesize such peak hour snapshots with different test cities. The results are in general better than daily traffic snapshot synthesis in Table I. We hypothesize this is because traffic patterns across a day show more variations than in a peak hour.

2) Comparison with CGAN and CVAE: Two popular methods for (weakly) conditioned generation are CGAN and CVAE. Compared to CartaGenie, CGAN has an extra adversarial loss while CVAE uses the variational formulation for the input noise. We have evaluated the benefit with these variants of CartaGenie by: (1) adding an adversarial loss (to mimic CGAN); (2) using an encoder to provide noise (CVAE); or (3) adding both. We have observed little to marginal performance improvement with these more complex training procedures (results omitted due to lack of space) for the data we are using, which validates our choice to simply train CartaGenie using L1 loss.

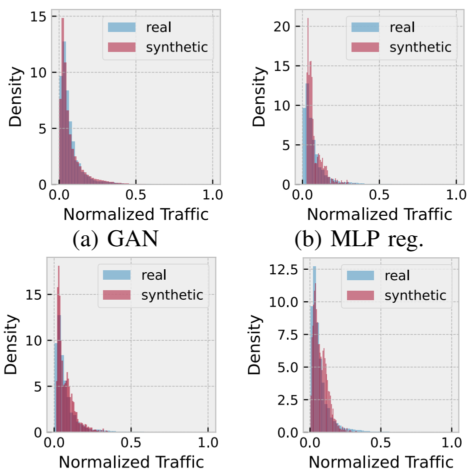
V. Use cases

To assess the utility of the synthetic datasets generated by CartaGenie to support data-driven research, we employ them



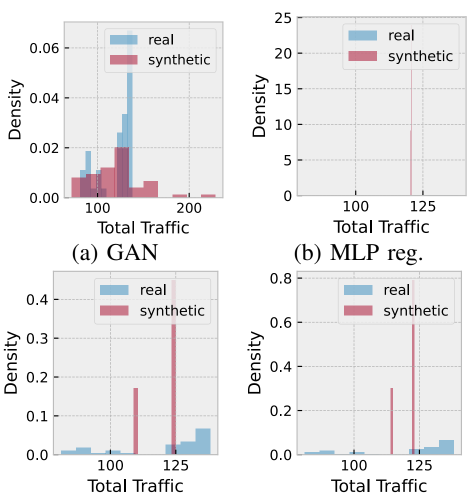
(a) GAN (b) MLP regression (c) CNN regression (d) Pix2Pix

Figure 8: Generalization results using benchmark methods with City A as test city. For each sub-figure, from top to bottom are the averaged synthetic city map for weekday (left) and weekend (right) and the time series plot of daily total traffic (bottom). Note that the traffic scale in Figure 8a is different from the rest as the synthetic data in this case spans a wider range (see y-axis ticks).



(a) GAN (b) MLP reg. (c) CNN reg. (d) Pix2Pix

Figure 9: Traffic histogram at pixel level with benchmarks.



(a) GAN (b) MLP reg. (c) CNN reg. (d) Pix2Pix

Figure 10: Histogram of total traffic with benchmarks.

to inform models for (i) the planning of edge datacenters in a mobile network, and (ii) the estimation of dynamic presence in urban areas. We use CartaGenie-generated traffic for previously unseen cities as input to these models; we then compare the results against those obtained by feeding the same models with the real traffic recorded by the operator. This lets us study if the synthetic traffic data produced by CartaGenie allows for a dependable evaluation of research solutions.

A. *Network edge datacenter planning*

Recent trends in mobile networking are fostering the softwarization and virtualization of the infrastructure. At the radio access, this has led to the emergence of functional split paradigms, where communication tasks traditionally performed in hardware at base stations are moved to edge datacenters [34]. The planning of the edge infrastructure must align to the spatial distribution of traffic demands. A recent model for the placement of datacenters can minimize the transmission latency while balancing the load across facilities [35]. Given the desired number of datacenters, the model identifies their ideal locations and served spatial areas (*e.g.*, base station coverage zones, antenna sectors, or, as in our case, tiles of a geographical tessellation corresponding to pixels).

Specifically, the model in [35] represents the system as a graph where vertices map onto pixels $p_{i,j} \in \mathcal{P}$, each with an expected daily traffic $\boldsymbol{x}_{i,j}$; edges $e_{(i,j)\to(i',j')} \in \mathcal{E}$ link pixels $p_{i,j}$ and $p_{i',j'}$ only if they are adjacent in space. Then, it partitions the graph so that the total traffic generated by the cells in each partition $d \in \mathcal{D}$ is balanced. This results in a set of geographical regions (each a union of the cells in a same partition) that can be associated to one edge datacenter: intuitively, the baricenters of the cells in each cluster can be mapped to the ideal positions where to deploy the datacenters. More formally, the model defines graph partitioning as an Integer Linear Programming (ILP) problem:

$$\min \sum\nolimits_{e_{(i,j)\to(i',j')}\in\mathcal{E}} \phi(e_{(i,j)\to(i',j')})$$
$$\text{s.t. } (1-\epsilon) \le \frac{\sum_{p_{i,j}\in\mathcal{P}} \psi(p_{i,j},d)\cdot \boldsymbol{x}_{i,j}}{\frac{1}{|\mathcal{D}|}\sum_{p_{i,j}\in\mathcal{P}} \boldsymbol{x}_{i,j}} \le (1+\epsilon), \quad \forall d \in \mathcal{D}$$
$$\sum\nolimits_{d\in\mathcal{D}} \psi(p_{i,j},d) = 1, \quad \forall p_{i,j} \in \mathcal{P}. \tag{1}$$

Here, $\phi(e_{(i,j)\to(i',j')})$ and $\psi(p_{i,j},d)$ are decision variables set to one if edge $e_{(i,j)\to(i',j')}$ is cut by the partition, and if pixel

$p_{i,j}$ is associated with datacenter d, respectively; they are zero otherwise. This means that $\forall e_{(i,j)\to(i',j')} \in \mathcal{E}$, $\forall d \in \mathcal{D}$, $\phi(e_{(i,j)\to(i',j')})$ takes a value one only if $e_{(i,j)\to(i',j')}$ links cells in partitions that are associated to different datacenters.

We assess the quality of the synthetic traffic generated by CartaGenie to solve the problem of network edge datacenter planning in (1). We first feed the synthetic traffic data to the ILP problem, and solve it: this returns cell graph partitions (hence datacenter locations) based on CartaGenie output. Then, we enforce the CartaGenie-driven partitions *on the real-world traffic*, and measure how balanced demands are across datacenters, using Jain's fairness index [36]. Finally, we compare the result with the fairness index obtained when the planning problem in (1) is solved using the actual traffic data.

Figure 11 shows results for two cities[*], with 4 to 10 edge datacenters.

The deployments driven by synthetic and real-world traffic only differ by 1-3% in terms of fairness, proving that CartaGenie can consistently enable dependable research based on optimization models like that in (1).

B. Dynamic presence estimation

The estimation of human presence in metropolitan areas is a challenging open problem in geo-informatics, which aims at capturing fluctuations in the distribution of people occurring at timescales of minutes, hours or days[*]. In this context, models have been proposed that leverage mobile traffic as a proxy for user presence, and map spatiotemporal variations in the network activity onto population density dynamics.

We consider a recent multivariate regression model that links mobile network traffic and activity levels to estimate dynamic presence [9]. Formally, the model is defined as $\boldsymbol{p} = \exp(k_1\lambda + k_2) \cdot \boldsymbol{x}^{k_3\lambda+k_4}$, where $\boldsymbol{p}$ is the population computed as a power law of the mobile traffic volume $\boldsymbol{x}$, λ is the activity level computed as the mean number of network events (*e.g.*, established calls) per subscriber, while k_1, k_2, k_3, and k_4 are constant model parameters. In our case, $\boldsymbol{x}$ is the total traffic observed in a single day, which allows inferring the average daily distribution of people in the target urban region. We set λ (the average activity level observed during a 24-hour period) to 0.5, and all constants to the typical values indicated in [9]. Note that the returned $\boldsymbol{p}$ is inherently different from inhabitant distributions captured, *e.g.*, by census: indeed, $\boldsymbol{p}$ accounts for the locations and activities of people during the whole day, which results in a very diverse presence.

In order to assess the quality of CartaGenie for this use case, we generate daily population estimates separately based on (i) the synthetic traffic by CartaGenie and (ii) the actual traffic recorded by the operator. We then compare the resulting population densities in terms of OT-SSIM and OT-PSNR. Performance figures are summarized in Table IV, when the mobile traffic in each reference city (in the columns) is generated with CartaGenie trained on the other urban regions. The results are very good overall, with OT-PSNR well above 26 in all cases, and OT-SSIM typically around 0.8.

[*]Results are equivalent in City C and D, and are omitted due to space limits.

[*]It is worth noting that dynamic presence is a very different concept from the static population density information that is available from census data, and which we use as a contextual attribute input to CartaGenie. The latter only capture home locations, whereas the former tries to model time-varying distributions induced by individual mobility. Therefore, inference of dynamic presence from population density is not possible.

VI. Discussion

a) On modeling data with different time granularity: The results in §IV demonstrate that CartaGenie can support traffic snapshot generation at different time granularity (hourly in §IV-C1 and daily in the rest of the section). To this end, only hyper-parameters tuning (*e.g.*, adapting the learning rate or the size of the hidden layers) is needed, without any modification to the CartaGenie architecture.

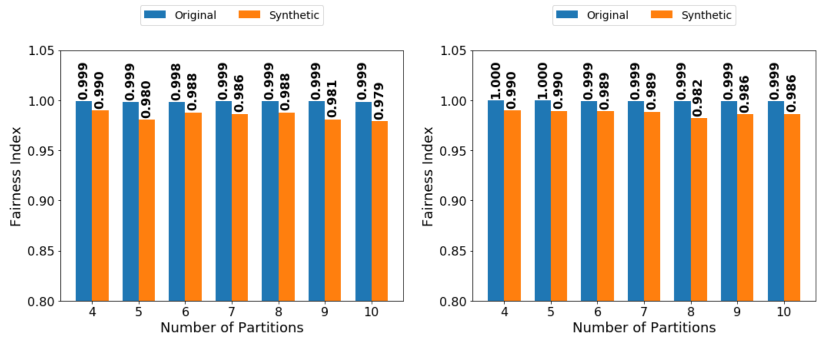
b) On temporal modeling of mobile traffic data: Note that time is provided as an explicit condition in CartaGenie, since the focus of this paper is about spatial modeling rather than temporal modeling. In fact, reproducing temporal patterns of mobile traffic data is a non-trivial task that requires dedicated investigations; for analysis and a generation tool targeting the temporal dimension of mobile traffic data, see, *e.g.*, SpectraGAN [37].

c) On modeling extreme traffic variations: To model and generate unforeseen or anomalous traffic patterns, *e.g.*, during special events or capturing pandemic effect, it would be necessary to consider additional contextual attributes such that the model can learn the appropriate correlation. Due to lack of training data with exceptional traffic variations, we do not study this issue in the paper but it is an interesting one to explore in future work.

d) On the choice of benchmark models: In §III, we mainly consider well-established deep generative methods for sample quality rather than more recent proposed deep generators that aim to improve other aspects of deep generative modeling such as tractability. In fact, as shown in a large-scale study of adversarial generative models [38], legacy GANs perform very well when compared to other more recently proposed methods in terms of sheer image quality; therefore, traditional GANs or methods based on GANs such as Pix2Pix serve as a strong baseline for comparison in our context, where the fidelity of the generated traffic is the key metric.

VII. Related Work

Network traffic generation is a very different problem depending on the dimension and scale it is tackled at. In its most traditional form, traffic generation aims at creating packet-level workloads for which several tools exist and are commonly used in research and engineering (e.g., iperf [39], Ostinato [40], D-ITG [41]) and also embedded in popular network simulators, such as ns-3 [42] or OMNeT++ [43]. This form of traffic generation does not have a spatial dimension and as such accounting for spatial correlations does not arise. Temporal correlations are also mostly overlooked. In contrast, we tackle a different and more complex challenge of generating the mobile traffic volumes at city scale (across multiple users and flows) by faithfully modeling underlying spatial and temporal fluctuations.



City A

City B

Figure 11: Jain's Fairness Index of load across a varying number of edge datacenters under real and CartaGenie-generated traffic.

Table IV: Difference between population estimates computed using original versus synthetic traffic.

	Condition	OT-PSNR↑	OT-SSIM↑
City A	Weekday	26.19	0.664
	Weekend	27.72	0.720
City B	Weekday	31.80	0.777
	Weekend	32.00	0.804
City C	Weekday	27.80	0.781
	Weekend	30.53	0.804
City D	Weekday	29.62	0.801
	Weekend	30.23	0.803

Di Francesco et al. [13], in what is the only existing work related to mobile network traffic generation to the best of our knowledge, assemble a cellular dataset for a given region based on estimated data demand per subscriber; for the latter, [13] simply models the per-subscriber demand for the busy-hour period across a given region from a real operator traffic dataset as a probability distribution (specifically, log-normal) and then "independently" samples from it when creating the dataset to estimate the per-subscriber data demand at any given location. In doing so, it implicitly makes an assumption that the peak traffic generated by a mobile subscriber corresponds to their residential location, which is seemingly incorrect especially in urban regions. In sharp contrast, CartaGenie is a neural network based data-driven method, that does not make any such assumptions, preserves correlations and finer variations in traffic over space and also models changes in traffic patterns over time. Crucially, unlike [13], we also validate the fidelity of CartaGenie against ground-truth information and demonstrate it to be capable of generating realistic synthetic traffic over previously unseen city-scale regions purely based on publicly available contextual input.

DoppelGANger [44] is a recent work that shares the same high-level goal as ours, that is to reduce the barrier to data-driven research in networking and beyond via synthetic data generation. DoppelGANger is however only broadly related in that it exclusively focuses on time-series data (e.g., web article views over time, network monitoring data over time). On the other hand, the spatial dimension features prominently in our work as we aim at generating city-scale mobile network data traffic for a given period of time, which requires modeling inherent and complex context-dependent correlations in mobile traffic across space and time.

Beyond the networking domain, our mobile traffic map generation problem is broadly related to data generation problems in two other domains – image synthesis in computer vision and road traffic generation in the transportation domain. Pix2Pix [22] is a representative example in the former case which takes a conditional GAN approach for image generation based on U-Net architecture [26]. As we show in our evaluation results, Pix2Pix approach has two key limitations when applied to the mobile traffic map generation problem compared to our well tailored CartaGenie approach: (i) from using dropout for stochasticity, it fails to model variation in the data; (ii) it introduces artifacts when sewing a city traffic map from generated traffic patches. For the case of road traffic generation, TrafficGAN [45] is a representative example. Like Pix2Pix, TrafficGAN also takes a conditional GAN approach but assumes knowledge of traffic correlations among roads in the target region for which (road) traffic needs to be generated. Such an assumption is strong and unrealistic assumption in our setting. In contrast, our CartaGenie method does not make any assumption about the nature or knowledge of (mobile) traffic in the target region and solely relies on easily available contextual input.

VIII. Conclusions

We have presented CartaGenie, a novel method rooted in deep generative modeling, for synthesizing high fidelity city-scale mobile network traffic snapshots. At its heart, CartaGenie is a conditional generator with tailored convolutional neural network model. CartaGenie requires only publicly available contextual information of a region (*e.g.*, city) such as population density in order to generate realistic mobile traffic map for that region. Our extensive evaluations based on real-world mobile traffic measurement data show that CartaGenie is not only able to generate city-scale mobile traffic snapshot data that is superior in fidelity to a range of benchmark solutions but also generalizes well in synthesizing mobile traffic maps for unseen cities. The above strengths in turn allow synthetic data generated with CartaGenie to be used for data-driven applications in mobile networking and beyond in a way that is indistinguishable from using actual data, as we demonstrate through two use cases.

Acknowledgments

We thank our shepherd Chiara Boldrini and the anonymous reviewers for their helpful suggestions to improve the paper.

References

[1] F. Calabrese, L. Ferrari, and V. D. Blondel, "Urban sensing using mobile phone network data: A survey of research," *ACM Computing Surveys*, vol. 47, no. 2, 2014.

[2] D. Naboulsi, M. Fiore, S. Ribot, and R. Stanica, "Large-scale mobile traffic analysis: A survey," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 124–161, 2016.

[3] R. Becker *et al.*, "Human mobility characterization from cellular network data," *Communications of the ACM*, vol. 56, no. 1, p. 74–82, Jan. 2013.

[4] S. Isaacman, V. Frias-Martinez, and E. Frias-Martinez, "Modeling human migration patterns during drought conditions in La Guajira, Colombia," in *Proc. of the 1st ACM SIGCAS Conference on Computing and Sustainable Societies (COMPASS)*, 2018.

[5] K. S. Kung, K. Greco, S. Sobolevsky, and C. Ratti, "Exploring universal patterns in human home-work commuting from mobile phone data," *PLOS ONE*, vol. 9, pp. 1–15, 06 2014.

[6] Y. Dong *et al.*, "Inferring unusual crowd events from mobile phone call detail records," in *Machine Learning and Knowledge Discovery in Databases*, 2015, pp. 474–492.

[7] S. Chang, E. Pierson, P. Koh *et al.*, "Mobility network models of COVID-19 explain inequities and inform reopening," *Nature*, 2020.

[8] P. Deville, C. Linard, S. Martin, M. Gilbert, F. R. Stevens, A. E. Gaughan, V. D. Blondel, and A. J. Tatem, "Dynamic population mapping using mobile phone data," *Proceedings of the National Academy of Sciences*, vol. 111, no. 45, pp. 15 888–15 893, 2014.

[9] G. Khodabandelou, V. Gauthier, M. Fiore, and M. A. El-Yacoubi, "Estimation of static and dynamic urban populations with mobile network metadata," *IEEE Transactions on Mobile Computing*, vol. 18, no. 9, pp. 2034–2047, 2019.

[10] S. Grauwin *et al.*, "Towards a comparative science of cities: Using mobile traffic records in New York, London, and Hong Kong," in *Computational Approaches for Urban Environments*, M. Helbich, J. Jokar Arsanjani, and L. M., Eds. Springer, 2015, vol. 13.

[11] A. Janecek, D. Valerio, K. A. Hummel, F. Ricciato, and H. Hlavacs, "The cellular network as a sensor: From mobile phone data to real-time road traffic monitoring," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 5, pp. 2551–2572, 2015.

[12] F. Pinelli, R. Nair, F. Calabrese, M. Berlingerio, G. Di Lorenzo, and M. L. Sbodio, "Data-driven transit network design from mobile phone trajectories," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 6, pp. 1724–1733, 2016.

[13] P. Di Francesco, F. Malandrino, and L. A. DaSilva, "Assembling and using a cellular dataset for mobile network analysis and planning," *IEEE Transactions on Big Data*, vol. 4, no. 4, pp. 614–620, 2018.

[14] G. Vallero, D. Renga, M. Meo, and M. A. Marsan, "Greener RAN operation through machine learning," *IEEE Transactions on Network and Service Management*, vol. 16, no. 3, pp. 896–908, 2019.

[15] D. Bega, M. Gramaglia, M. Fiore, A. Banchs, and X. Costa-Pérez, "Deepcog: Optimizing resource provisioning in network slicing with AI-based capacity forecasting," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 2, pp. 361–376, 2020.

[16] M. Z. Shafiq, L. Ji, A. X. Liu, J. Pang, and J. Wang, "Characterizing geospatial dynamics of application usage in a 3G cellular data network," in *Proceedings of IEEE INFOCOM*, 2012, pp. 1341–1349.

[17] U. Paul, A. P. Subramanian, M. M. Buddhikot, and S. R. Das, "Understanding traffic dynamics in cellular data networks," in *Proceedings IEEE INFOCOM*, 2011, pp. 882–890.

[18] I. Goodfellow *et al.*, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[19] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[20] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," *arXiv preprint arXiv:1401.4082*, 2014.

[21] E. Perez *et al.*, "Film: Visual reasoning with a general conditioning layer," *arXiv:1709.07871 [cs, stat]*, Dec. 2017.

[22] P. Isola, J. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[23] M. Mathieu, C. Couprie, and Y. LeCun, "Deep multi-scale video prediction beyond mean square error," *arXiv preprint arXiv:1511.05440*, 2015.

[24] D. Barber, *Bayesian reasoning and machine learning*. Cambridge University Press, 2012.

[25] K. Xu, R. Singh, H. Bilen, M. Fiore, M. K. Marina, and Y. Wang, "CartaGenie: Context-Driven Synthesis of City-Scale Mobile Network Traffic Snapshots," The University of Edinburgh, School of Informatics, Tech. Rep., Jan 2022. [Online]. Available: https://netsys-edinburgh.github.io/CartaGenie/tr-jan22.pdf

[26] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[27] Copernicus urban atlas. https://land.copernicus.eu/local/urban-atlas/urban-atlas-2012.

[28] OpenStreetMap Overpass API. http://overpass-turbo.eu/.

[29] G. Peyré and M. Cuturi, "Computational optimal transport," *arXiv preprint arXiv:1803.00567*, 2020.

[30] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[31] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

[32] D. P. Kingma *et al.*, "Semi-supervised learning with deep generative models," *Advances in neural information processing systems*, vol. 27, pp. 3581–3589, 2014.

[33] N. Thomos, N. Boulgouris, and M. Strintzis, "Optimized transmission of jpeg2000 streams over wireless channels," *IEEE Transactions on Image Processing*, vol. 15, no. 1, pp. 54–67, 2006.

[34] A. Garcia-Saavedra, G. Iosifidis, X. Costa-Perez, and D. J. Leith, "Joint optimization of edge computing architectures and radio access networks," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 11, pp. 2433–2443, 2018.

[35] C. Marquez *et al.*, "How Should I Slice My Network? A Multi-Service Empirical Evaluation of Resource Sharing Efficiency," in *Proc. ACM MobiCom*, 2018, p. 191–206.

[36] R. Jain, A. Durresi, and G. Babic, "Throughput fairness index: An explanation," *ATM Forum Document Number: ATM Forum/990045*, Feb. 1999.

[37] K. Xu, R. Singh, M. Fiore, M. K. Marina, H. Bilen, M. Usama, H. Benn, and C. Ziemlicki, "Spectragan: Spectrum based generation of city scale spatiotemporal mobile network traffic data," in *Proceedings of the 17th International Conference on Emerging Networking EXperiments and Technologies*, ser. CoNEXT '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 243–258. [Online]. Available: https://doi.org/10.1145/3485983.3494844

[38] M. Lucic, K. Kurach, M. Michalski, S. Gelly, and O. Bousquet, "Are GANs Created Equal? A Large-Scale Study," *arXiv preprint arXiv:1711.10337*, 2017.

[39] iPerf3: A TCP, UDP, and SCTP network bandwidth measurement tool. https://iperf.fr/.

[40] Ostinato packet generator. https://ostinato.org/.

[41] A. Botta, A. Dainotti, and A. Pescapè, "A tool for the generation of realistic network workload for emerging networking scenarios," *Computer Networks*, vol. 56, no. 15, pp. 3531–3547, 2012.

[42] Network Simulator (NS-3). https://www.nsnam.org/.

[43] OMNeT++: Discrete Event Simulator. https://omnetpp.org/.

[44] Z. Lin, A. Jain, C. Wang, G. Fanti, and V. Sekar, "Using GANs for sharing networked time series data: Challenges, initial promise, and open questions," in *Proc. ACM Internet Measurement Conference (IMC)*, October 2020, p. 464–483.

[45] Y. Zhang, Y. Li, X. Zhou, X. Kong, and J. Luo, "TrafficGAN: Off-Deployment Traffic Estimation with Traffic Generative Adversarial Networks," in *Proc. IEEE International Conference on Data Mining (ICDM)*, Nov. 2019, pp. 1474–1479.