TrafficGAN: Off-Deployment Traffic Estimation with Traffic Generative Adversarial Networks

Yingxue Zhang*, Yanhua Li*, Xun Zhou†, Xiangnan Kong*, Jun Luo‡
*Worcester Polytechnic Institute
†University of Iowa
‡Lenovo Group Limited
yzhang31@wpi.edu; yli15@wpi.edu; xun-zhou@uiowa.edu; xkong@wpi.edu; jluo1@lenovo.com

***Abstract*—The rapid progress of urbanization has expedited the process of urban planning, *e.g.*, new residential, commercial areas, which in turn boosts the local travel demand. We propose a novel "off-deployment traffic estimation problem", namely, to foresee the traffic condition changes of a region prior to the deployment of a construction plan. This problem is important to city planners to evaluate and develop urban deployment plans. However, this task is challenging. Traditional traffic estimation approaches lack the ability to solve this problem, since no data about the impact can be collected before the deployment and old data fails to capture the traffic pattern changes. In this paper, we define the off-deployment traffic estimation problem as a traffic generation problem, and develop a novel deep generative model TrafficGAN that captures the shared patterns across spatial regions of how traffic conditions evolve according to travel demand changes and underlying road network structures. In particular, TrafficGAN captures the road network structures through a dynamic filter in the dynamic convolutional layer. We evaluate our TrafficGAN using a large-scale traffic data collected from Shenzhen, China. Results show that TrafficGAN can more accurately estimate the traffic conditions compared with all baselines.**

***Index Terms*—traffic estimation, TrafficGAN, generative model.**

I. Introduction

Over the past a few decades, we have witnessed drastic urbanization at the global scale. It is reported that the world's urban population ratio has reached 54% in 2014, and it is projected that by 2050, two-thirds of the world population will live in urban areas [3].

With the rapid progress of urbanization, urban planning is becoming a vital problem concerning with resources allocation, urban transportation efficiency and living environment. The fast development of new residential and commercial areas always comes with population growth, which in turn increases the travel demands and the risk of worsening traffic conditions due to the overload of the transportation infrastructures. For example, the Olympic Village was built in the northern area of Beijing for the 2008 Olympic Games with many new residential and commercial areas constructed in its nearby areas as illustrated in Fig. 1. The population in that region increased drastically after 2008, which significantly worsened the local traffic conditions [11]. This could have been avoided if more thorough and accurate traffic evaluation had been done before the constructions. Therefore, it is crucial to foresee both positive and negative impacts on traffic conditions before an urban construction plan is deployed. In our work, we refer to such a problem as "*off-deployment traffic estimation*" problem. Solving this problem is technically challenging, since no new data can be collected before deployment in an area, while old data collected before deployment fails to capture the traffic pattern changes.

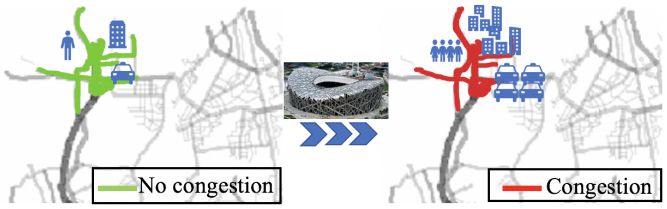

Fig. 1: Traffic condition changes around Olympic Village in Beijing, China

The traffic estimation problem has been extensively studied in the literature [2], [10], [17]. These works use the historical traffic data of regions to build machine learning models that capture the correlations among the past traffic, environmental features and the future traffic. However, when predicting the traffic impact of a newly developed construction plan, these models will fail because they cannot capture the future traffic pattern changes caused by the new deployment plan due to the lack of training samples. Traditionally in civil engineering, agent-based simulation models [12] or physical models [18] are used to estimate the projected traffic volume after constructions. However, these models rely heavily on model choice and parameter settings, which are not transferable across urban regions.

In this paper, we propose a novel traffic generative adversarial network (TrafficGAN) to tackle the off-deployment traffic estimation problem. The proposed TrafficGAN captures the traffic correlations along the underlying road networks, and can estimate traffic conditions prior to deployment of a construction plan. Our **main contributions** are summarized as follows:

• We model the off-deployment traffic estimation problem as a traffic data generation problem, and propose a novel deep generative model – TrafficGAN, which captures the shared patterns across spatial regions of how traffic conditions

evolve according to travel demand changes and underlying road network structures. (See Sec IV.)

• We evaluate TrafficGAN using a large scale traffic data collected during 7/2016-12/2016 from Shenzhen, China. The unique dataset represents a wide range of regions with diverse travel demands and traffic conditions in both rural and urban areas. Our results demonstrate that our proposed TrafficGAN can accurately estimate the traffic conditions compared with all baselines. (See Sec VI.)

II. Overview

In this section, we define the off-deployment traffic estimation problem, outline our solution framework and describe the datasets we use.

A. Problem Definition and Solution Framework

Urban planning, especially, governmental zoning[1], is a process of planning land use and development in a target region. In this work, we focus on urban deployment and zoning plans when developing certain new residential or commercial areas in a target region. Denote a city under planning as R_0.

Definition 1 (Grid cell s). The planning city R_0 is partitioned into N_0 grid cells with equal side-length in latitude and longitude, denoted as $S = \{s_i\}$, where $1 \quad i \quad N_0; i \in \mathbb{N}$.

Definition 2 (Target region R). A target region R is a square geographic region in R_0, formed by $\grave{} \quad \grave{}$ grid cells. Formally, $R = \langle s; \grave{} \rangle$ is uniquely defined by an anchor grid cell s on its top-left corner and a number $\grave{}$ of grid cells on the side[2].

Definition 3 (Travel demand of a grid cell and a target region). The travel demand of a geographic area captures the total number of departures in a period of time, *e.g.*, one hour interval. Thus, we denote the travel demand of a grid cell s as $d_s \in \mathbb{N}$. Given a target region R, D_R is an $\grave{} \quad \grave{}$ matrix representing the travel demand distribution of all grid cells in R. Moreover, we denote the *total travel demand* of a target region R as $d_R \in \mathbb{N}$, which is the sum of travel demands in all grid cells within R, *i.e.*, $d_R = \sum_{s \in R} d_s = \sum_{1 \quad i,j \quad \grave{}} D_R(i,j)$.

In general, it is hard to obtain the total travel demand in a region including all transport modes. In this work, we use the demand for taxis to represent the regional travel demand, where many studies have shown that taxi demands represent the total demands quite well [6], [16].

Definition 4 (Traffic status of a grid cell and traffic distribution of a target region). Traffic status includes various measures representing the quality of traffic in a geographic region, such as average driving speed, traffic inflow/outflow, traffic volume, *etc*. Taking traffic inflow as an example, we denote m_s as the traffic inflow of grid cell s in a period of time. Similar, given a target region R with $\grave{} \quad \grave{}$ grid cells, we denote an $\grave{} \quad \grave{}$ matrix M_R as the traffic distribution in R. Each element of M_R represents the taxi inflow in a grid cell.

Definition 5 (Urban deployment plan). An urban deployment plan in a target region R is referred to a plan to construct new residential or commercial areas in the region R without changing the road structures. As a part of the plan, the expected travel demand after deployment is specified[3], denoted by $\hat{d}_R$.

Problem definition. Given a city area R_0 partitioned into grid cells S, the citywide historical travel demands and traffic distributions $D_{R_0;t}$ and $M_{R_0;t}$ are available over a time span $1 \quad t \quad T$. For a target region $R = \langle s; \grave{} \rangle$ and a deployment plan in R with the expected travel demand $\hat{d}_R$, we aim to estimate the traffic distribution $M_R(\hat{d}_R)$.

Solution framework. Our off-deployment traffic estimation framework takes taxi GPS data and road map data as inputs, processes the data in three stages to get the output: *Stage 1 (Data Preprocessing)*, *Stage 2 (TrafficGAN Training)* and *Stage 3 (Urban Plan Evaluation)* which are detailed in Sec III, IV and V, respectively.

B. Data Description

We use two real world datasets in this paper, (1) taxi GPS data; (2) road map data. For consistency, all datasets are collected from the same time interval, *i.e.*, from Jul 1st to Dec 31st, 2016 in Shenzhen, China.

Taxi GPS data contains GPS records collected from taxis in Shenzhen, China from Jul 1st to Dec 31st, 2016. There are 17,877 taxis equipped with GPS sensors, each GPS sensor generates a GPS record every 40 seconds on average. Overall, a total number of 51,485,760 GPS records are collected each day.

Road map[4]**.** In our study, we use the Google GeoCoding[5] to retrieve the bounding box of Shenzhen. The bounding box is defined between $22{:}534$ to $22{:}87$ in latitude and $113{:}77$ to $114{:}40$ in longitude.

III. Stage 1: Data Preprocessing

A. Map Gridding

For the ease of implementation in practice, we adopt the grid based method, which simply partitions the map into equal side-length grids [13]. In this paper, we divide the map of Shenzhen City into $40 \quad 50$ grid cells with a side-length $l_1 = 0{:}0084$ in latitude and $l_2 = 0{:}0126$ in longitude.

B. Training Sets Construction

Given all $40 \quad 50$ grid cells in Shenzhen, we choose target region size $\grave{} = 10$ as an example in this study, where our TrafficGAN can actually apply to any target region size. Thus, there are in total $1{,}271$ possible target regions with size $10 \quad 10$. The location of each region is described with a tuple $(i; j)$ which indicates the coordinates of the first grid cell (the upper-left one) in the region, *i.e.*, the row and column index $0 \quad i \quad 30; 0 \quad j \quad 40; i, j \in \mathbb{N}$. However, it is unnecessary and too costly to use data from all $1{,}271$ regions as training data.

[1] https://en.wikipedia.org/wiki/Zoning

[2] Note that target regions can also be defined as rectangles rather than squares. For simplicity, we use square shape of target regions in this work.

[3] The expected travel demand $\hat{d}_R$ after deploying a construction plan is assumed given in this paper, which can be done by commonly used Four-Steps demand forecasting approaches in Civil Engineering [14].

[4] http://www.openstreetmap.org/

[5] https://developers.google.com/maps/documentation/geocoding/

Instead, we select 63 regions covering entire Shenzhen city as target regions, extract their traf c distributions and travel demands over time.

Travel Demand. We use six months taxi GPS records of Shenzhen, China in 2016 to extract the travel demand of each grid cell and region in Shenzhen. In each time slot, i.e., one hour, we count the total pickup events within each grid cell and each` ` region.

Traf c Distribution. Traf c distribution re ects the traf c conditions in a region, which is quanti ed with traf c in ow in this study. Since it is hard to obtain the total traf c in ow in a grid cell including all transport modes, in this paper, we use taxi in ow to represent traf c in ow and many studies has proved its effectiveness [8], [20]. In each time slot of each day, we count all taxis which stay or arrive at each grid cell as the taxi in ow.

IV. Stage 2: TrafficGAN Training

Taking an analogy, our "off-deployment traf c estimation" problem is similar as image generation problem, where the traf c distribution of a region can be viewed as a gray-scale "image", the traf c status (e.g., in ow) of each grid can be viewed as a "pixel" value. Thus, image generation approaches, such as GANs [7], sound a promising solution. However, the unique challenges of our problem prevent the state-of-the-art GAN models from solving it:

• Traf c correlations along road networks. In a target region R, the traf c of neighboring grids along the underlying road networks has strong correlations. Capturing such correlations is non-trivial since the correlation patterns are de ned by the road network structures, which may have irregular shapes (rather than squares or rectangles).

• Conditioned Traf c Distribution Generation. The generated traf c distribution is meaningful only when conditioned on the given region R and the travel demand d_R. However, how to design a generative model that outputs the traf c distributions for a desired region and travel demand is challenging.

In this section, we introduce our Traf cGAN for off-deployment traf c estimation problem.

A. Quantifying Traf c Correlation

Traf c correlation is used to capture the inherent traf c dependence between a grid cell pair. We use the Pearson correlation coef cient between time series traf c data of a grid cell pair to quantify its traf c correlation. Pearson correlation coef cient [9] a can be calculated by Eq. 1, where X and Y are time series taxi in ow data of two grid cells in our case, $\overline{X}$ and $\overline{Y}$ are the mean of X and Y:

$$a_{XY} = \frac{\sum_{i=1}^{n} (X_i - \overline{X})(Y_i - \overline{Y})}{\sqrt{\sum_{i=1}^{n} (X_i - \overline{X})^2} \sqrt{\sum_{i=1}^{n} (Y_i - \overline{Y})^2}}; \quad (1)$$

a $2 [-1; 1]$. For an` ` region R, its corresponding traf c correlation matrix A is a symmetric` 2 ` 2 matrix, where the entry a_{ij} is the traf c correlation between grid cells s_i and s_j, $s_i; s_j \in R$.

In a road network, nearby road segments (resp. nearby grid cells) often have stronger correlations in traf c according to the First Law of Geography [19], and the effective traf c correlations are generally positive. In our work, for a speci c grid cell, we only keep its nearby grid cells which are directly connected with it by roads and thus (likely) to have positive traf c correlations.

After removing other uncorrelated grids, we perform row normalization for the traf c correlation matrix A, i.e., $a_{ij} = a_{ij} = \sum_{j=1}^{`^2} a_{ij}$, so it will not affect the scale of features when multiplied to the feature matrix in Eq. 2.

Next, we will elaborate on how to integrate the traf c correlation matrix for traf c distributions estimation and generation.

B. Traf cGAN

In this paper, to solve the challenges mentioned above, we propose a novel conditional generative model – Traf cGAN which can capture the traf c correlations of road networks, control the generation results with desired region and travel demand conditions, and generate realistic traf c distributions. Traf cGAN consists of a generator G and a discriminator D, and it applies dynamic convolutional layers in G and D.

1) Dynamic Convolutional Layer: The goal of dynamic convolutional layer is to learn a function of traf c features in a region including traf c in ow, volume, speed, etc. The input of dynamic convolutional layer includes two parts:

• A traf c feature matrix H of size $N \times F_0$ (N: number of grid cells in a region, $N = ` \times `$; F_0: initial number of traf c features).

• A non-negative and row-normalized traf c correlation matrix A of size $N \times N$.

The output is a new feature matrix after one-layer convolution. The layer-wise propagation rule is:

$$H_{i+1} = f(H_i; A) = \sigma(AH_iW_{i+1}); \quad (2)$$

where H_i is the feature matrix of a region got after ith layer and is the input of the $(i+1)$th layer, W_{i+1} is the weight matrix in $(i+1)$th layer and σ is an activation function.

Dynamic Convolutional Layer vs. Standard Convolutional Layer. By introducing the traf c correlation matrix A in dynamic convolutional layer, a dynamic " lter" is created and applied to the feature matrix H, where the size and the shape of the dynamic lter is controlled by A. The lter of a standard convolutional layer has xed size (e.g., a 3×3 square), which cannot naturally captures the traf c correlations along the road networks. In contrast, the dynamic lters created by the traf c correlation matrix can align with the road network well.

Moreover, the corresponding dynamic de-convolutional layer is the same as the dynamic convolutional layer. This is because the matrix operation of the dynamic convolutional layer and dynamic de-convolutional layer is invariant. We omit the detailed proof for brevity.

2) Traf cGAN Architecture: To tackle the challenge of conditioned traf c distribution generation, we introduce conditional generative model structure in designing Traf cGAN. The goal of the generator G is to generate traf c distributions

with respect to the region location loc and travel demand d. The input of the generator G includes three parts, i) a low-dimensional code vector z, randomly sampled from Gaussian distribution, ii) condition vector $c = [loc; d]$, defining the desired region and travel demand and iii) a traffic correlation matrix A_{loc}. The discriminator D takes three inputs, i) a traffic distribution M, ii) condition information $c = [loc; d]$ and iii) a traffic correlation matrix A_{loc}. D outputs a scalar indicating whether the traffic distribution M is real and whether the input M and c are matched. The detailed structures of generator G and discriminator D are detailed in Fig. 2a and Fig. 2b, $len(c)$ represents the number of conditions in c. In our case, $N = 100$, $F_0 = 1$ since the only traffic feature is taxi inflow.

Algorithm 1 TrafficGAN Training Process

Input: Training iteration K, a training set Z, initialized G and D.
Output: Well trained G and D.
1: In each training iteration $iter$:
2: repeat
3: Sample Z_0 from training set Z.
4: Sample N from Gaussian distribution.
5: Generate $\tilde{T}$ with G.
6: Sample $\hat{T}$ from training set Z.
7: Update D with Eq. 5 to maximize Eq. 4.
8: Update G with Eq. 7 to maximize Eq. 6.
9: until $iter > K$.

3) TrafficGAN Loss Function: In TrafficGAN, the generator G aims to generate "like-real" traffic distributions so that the discriminator D cannot distinguish the generated traffic distributions from the real traffic distributions well. As a result, the loss function of TrafficGAN is in the form of Eq. 3, modeled as a Min-Max game. (See more details in [15].)

$$\min_G \max_D V(D,G) = \mathbb{E}_{M \sim p_{data}(M)}[\log D(c; A_{loc}; M)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(c; A_{loc}; z)))]. \quad (3)$$

4) Training Process During the training process, we apply batch gradient descent. The detailed training process is shown in Algorithm 1, where the discriminator D and the generator G are updated in line 3 – 7 and line 8, respectively. Denote the training set which contains n samples as $Z = \{(c^1; A^1_{loc}; M^1); \ldots; (c^n; A^n_{loc}; M^n)\}$, with $c^i = [loc^i; d^i]$ as a condition vector. Denote $Z_0 = \{(c^1; A^1_{loc}; M^1); \ldots; (c^m; A^m_{loc}; M^m)\}$ (line 3) as a subset of Z containing m triples, where $m < n$. Denote $N = \{z^1; z^2; \ldots; z^m\}$ as a set of m codes sampled from Gaussian distribution (line 4), $\tilde{T} = \{\tilde{M}^1; \ldots; \tilde{M}^m\}$ as a set of m traffic distributions generated with G (line 5), where $\tilde{M}^i = G(c^i; A^i_{loc}; z^i)$. Denote $\hat{T} = \{\hat{M}^1; \hat{M}^2; \ldots; \hat{M}^m\}$ as a set of m traffic distributions sampled from the training set Z (line 6), each $\hat{M}^i$ is mismatched with $(c^i; A^i_{loc})$. In each training iteration, we update the parameters $_D$ of D with Eq. 4 and Eq. 5, where is learning rate.

$$\nabla_D = \frac{1}{m}\sum_{i=1}^{m} \log(1 - D(c^i; A^i_{loc}; \tilde{M}^i)) + \log D(c^i; A^i_{loc}; M^i) + \log(1 - D(c^i; A^i_{loc}; \hat{M}^i)); \quad (4)$$

$$_D = _D + _D \cdot \nabla_{_D}(_D): \quad (5)$$

Then, we update the parameters $_G$ of G with Eq.6 and Eq.7.

$$\nabla_G = \frac{1}{m}\sum_{i=1}^{m} \log D(G(c^i; A^i_{loc}; z^i)); \quad (6)$$

$$_G = _G + _G \cdot \nabla_{_G}(_G): \quad (7)$$

V. Stage 3: Urban Plan Evaluation

The generator G obtained from Stage 2 can be used by urban planners to evaluate urban construction plans at various locations, and search for more appropriate plans. To do so, given an urban deployment plan, the generator takes (i) the expected travel demand $\hat{d}_R$, (ii) the location of the target region R, (iii) traffic correlation matrix of R, and (iv) random code vector z, as inputs to generate traffic distributions for the plan to be evaluated.

Note that future traffic distributions hinge on many factors such as weather, etc. To capture the entire distribution of what the future traffic will look like over all potential (hidden) factors, we randomize a large number of random code vectors to regenerate the traffic distributions for the urban plan. All L generated traffic distributions $[\tilde{M}^1; \ldots; \tilde{M}^L]$ are used to capture the future traffic distributions. The urban planners can summarize and evaluate various statistics of their interests using the L generated traffic distributions, for example, the mean, variance, minimum, maximum of traffic distributions.

VI. Evaluations

We conduct experiments to evaluate our proposed TrafficGAN with baseline approaches using large scale real world taxi GPS data.

A. Experiment Design

We performed two sets of experiments: (i) Generate traffic distributions in a target region R that was "seen" by TrafficGAN in the training set but under other travel demands; (ii) Generate traffic distributions for an "unseen" target region R with a specific target travel demand d_R, where R was not included in the training set, therefore, TrafficGAN has never seen traffic distributions of R under any travel demand during training process. Obviously, the second task is more challenging.

In this paper, Euclidean distance is used to evaluate the quality of a generated traffic distribution against the ground truth traffic distribution of a target region R. Euclidean distance is defined as follows. For two vectors $V_s = (v_1; \ldots; v_n)$

(a) Generator of Traf cGAN (b) Discriminator of Traf cGAN

Fig. 2: Traf cGAN architecture

and $\hat{V} = (\hat{v}_1, \ldots, \hat{v}_n)$, the Euclidean distance between V and $\hat{V}$ is:

$$\|\hat{V} - V\|_2 = \sqrt{\sum_{i=1}^{n} (\hat{v}_i - v_i)^2}. \quad (8)$$

We de ne four statistics P_1, P_2, P_3, P_4 to measure and evaluate the difference between the generated traf c distribution and the ground-truth traf c distribution.

P_1: For each R and d_R pair, we calculate the average traf c distribution using real traf c distributions and refer to it as "true average distribution". We also calculate the average of generated traf c distributions and refer to it as "generated average distribution". The smaller Euclidean distance between the two average distributions (denoted with P_1) re ects that the mean of the generated data are similar to the mean of the true data.

P_2 and P_3: Under the condition of target region R and target travel demand d_R, for each grid cell $s \in R$, we calculate the Euclidean distance between s in real traf c distributions and in generated distributions so that we have $N = |R|^2$ Euclidean distances for all $s \in R$. The mean of them is denoted as P_2 and the standard deviation is denoted as P_3.

P_4 refers to the Euclidean distance between real traf c distributions and generated traf c distributions with various travel demands.

B. Baseline Models

We compare our Traf cGAN with four baseline approaches below.

Standard cGAN [15]. Without deep convolutional layers, the generator and discriminator are both composed of four fully-connected layers.

Conditional DCGAN [4]. The generator and discriminator of cDCGAN are composed of four transposed convolutional/convolutional layers.

Spatial smoothing approach with neighboring regions [5]. This method uses the traf c distributions of 9 closest regions under the same travel demand to compute a mean distribution as the resulting estimation.

Regression [1]. Ridge regression is applied to estimate the taxi in ow of each grid cell with the location of the grid cell and the travel demand as predictors.

TABLE I: Comparisons of $[P_1, \ldots, P_4]$ for an "unseen" region

	Traf cGAN	cGAN	cDCGAN	smoothing	regression
P_1	956.78	14321.60	1452.82	1178.62	55302.89
P_2	420.50	7096.76	523.37	NA	NA
P_3	314.21	1914.90	539.78	NA	NA
P_4	5249.24	73505.63	7519.95	NA	NA

TABLE II: Comparisons of $[P_1, \ldots, P_4]$ for a "seen" region

	Traf cGAN	cGAN	cDCGAN	smoothing	regression
P_1	896.95	14436.03	1473.42	1418.74	57792.57
P_2	361.74	6393.57	455.25	NA	NA
P_3	277.67	1837.80	512.83	NA	NA
P_4	4560.26	66524.57	6857.47	NA	NA

C. Experiment Settings

In the experiments, we train Traf cGAN, cGAN and cDCGAN both for 200 epochs, and randomly sample code z from a standard normal distribution. All models are trained using Adam [11] with $_1 = 0.5$ and $_2 = 0.999$, and a learning rate of $2 \quad 10^{-5}$ for the rst 10 epochs and linearly decayed to $2 \quad 10^{-6}$. In the training process, we use batch stochastic gradient descent with a batch size of 128.

D. Evaluation Results

1) Statistics comparisons with four baselines: We pick two representative regions (seen and unseen) as target regions with a speci c travel demand. The statistics of $P_1 \quad P_4$ are shown in Table. I and Table. II. For both "seen" and "unseen" regions, Traf cGAN has the lowest error in any metric among $P_1 \quad P_4$, which indicates the generated traf c distributions with Traf cGAN are much closer to the real ones. Compared with cGAN and cDCGAN, our Traf cGAN model brings down the P_1 error by up to 93.79% and 39.12% on the "seen" region and up to 93.32% and 34.14% on the "unseen" region.

2) Spatial pattern visualization: In this part, we visualize the generated/estimated traf c distributions and compare them with the real one. Here the traf c distributions are normalized to the same scale. Fig. 3 shows the visualizations of spatial patterns of 9 connected "unseen" regions, where each region has a corresponding travel demand. Fig. 3a marks the locations of selected 9 regions with red color on the whole city map. Fig. 3b shows the zoomed-in road map of the 9 regions. Fig. 3c shows the true average distribution and Fig. 3d shows the