DOI: 10.1111/mice.12697

ORIGINAL ARTICLE

COMPUTER-AIDED CIVIL AND INFRASTRUCTURE ENGINEERING WILEY

A Gaussian sampling heuristic estimation model for developing synthetic trip sets

S. F. A. Batista[1] | **Guido Cantelmo**[2] | **Mónica Menéndez**[1] | **Constantinos Antoniou**[2]

[1] Division of Engineering, New York University Abu Dhabi, Saadiyat Marina District, Abu Dhabi, United Arab Emirates

[2] Department of Civil, Geo and Environmental Engineering, Technical University of Munich, Munich, Germany

Correspondence
Constantinos Antoniou, Technical University of Munich, Arcisstrasse 21, 80333 Munich, Germany.
Email: c.antoniou@tum.de

Funding information
NYUAD Center for Interacting Urban Networks (CITIES), funded by Tamkeen under the NYUAD Research Institute, Award No. CG001. Swiss Re Institute under the Quantum Cities™ initiative. European Union Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement numbers 754462 and MOMENTUM N.815069.

Abstract

In this paper, we develop a heuristic model based on Gaussian processes to determine synthetic sets of trips in urban networks, considering only supply-related information. This is an alternative to the benchmark method used in the literature, which consists of repeating several trials of Monte Carlo simulations and therefore requiring a complex calibration task and large computational resources. The developed heuristic model explicitly leverages the probabilistic nature of Gaussian processes and exploits their properties to iteratively select origin–destination (od) pairs of nodes in the city network. The model then determines the shortest trip in distance for the selected od pairs and appends it to the synthetic set. We discuss the implementation and performance of both the benchmark method and the developed heuristic model on two city networks. We show that the presented model is more robust and computationally efficient than the benchmark method. This is evidenced by its ability to determine synthetic sets with much smaller sizes, naturally reducing the computational burden, when compared to the benchmark method. We also discuss how the choice of the kernel function and calibration of the hyperparameters influence the performance of the presented heuristic model.

1 | INTRODUCTION

Aggregated traffic models based on the macroscopic fundamental diagram (MFD) (Daganzo, 2007; Geroliminis & Daganzo, 2008; Godfrey, 1969; Vickrey, 2020) represent a promising tool in the design of traffic management schemes to mitigate congestion in metropolitan areas worldwide. The application of this kind of traffic model typically requires the partitioning of the city network into regions (Ji & Geroliminis, 2012; Lopez et al., 2017; Saeedmanesh & Geroliminis, 2017), that is, definition of a regional network, as depicted in Figure 1. Traffic is modeled as exchanged flows between adjacent regions. In each region, vehicles circulate at approximately the same average speed, and the traffic states are described by an MFD, that reflects the relationship between the number of vehicles (or accumulation) within the region and the average circulating flow (Daganzo, 2007; Geroliminis & Daganzo, 2008; Vickrey, 2020). The aggregated traffic models based on the MFD (Jin, 2020; Mariotte et al., 2020) have been used in a wide range of applications, including perimeter control strategies (Haddad & Zheng, 2018; He et al., 2019; Ren et al., 2020; Sirmatel & Geroliminis, 2019), route guidance (Batista & Leclercq, 2019; Yildirimoglu &

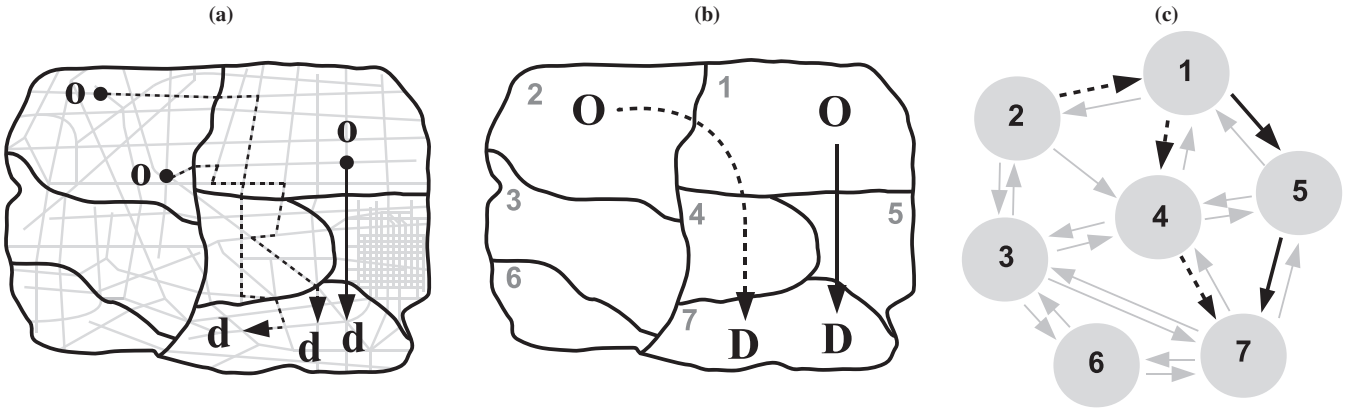

FIGURE 1 (a) Example of trips in the city network; (b) sequence of traveled regions by the trips; (c) example of paths on a regional network

Geroliminis, 2014), pricing schemes (Gu et al., 2019; Yang et al., 2019), urban parking (Cao et al., 2019), and environmental control schemes (Ingole et al., 2020).

The proper partitioning of a city network for MFD-based applications is still a question of research in the literature. From an MFD perspective (Lopez et al., 2017; Saeedmanesh & Geroliminis, 2017), the partitioning of the city network should lead to regions that are compact, well defined, fully connected, and homogeneous in terms of traffic conditions, that is, links inside each region should be in similar traffic states at any given time. Therefore, the partitioning is based on features of the supply (i.e., characteristics of the city network) and observable traffic conditions. Another alternative is to create traffic analysis Zones (TAZs) based on sociodemographic data and land-use information (Sinha et al., 1980). This approach leads to a partitioning of the city network that is consistent with the mobility demand. However, it does not guarantee that the resulting partitioning is appropriate for MFD-based applications, the same way that a partitioning based on supply features and observable traffic conditions might not be suitable for demand analysis. This research gap should be further explored, but is out of the scope of this study. In this paper, we focus on supply and traffic conditions information, inspired by the MFD-based applications, to partition the city network.

The partition of the city network permits determining the regional network, where routing options are defined (Batista & Leclercq, 2019; Batista et al., 2019). For any given pair of origin (O) and destination (D) regions, different paths are possible, each characterized by a different travel distance. Figure 1a depicts the example of three trips in the city network, each defined by an ordered sequence of traveled links from the origin node (o) to the destination node (d). The trips travel a different sequence of regions (see Figure 1b), thus representing different paths on the regional network. A path on the regional network—regional path—is represented by an ordered sequence of traveled regions from O to D, as shown in Figure 1c). The trips represented by the dashed line in Figure 1b are associated to the same regional path. One can observe in Figure 1a that each of these trips has a different traveled distance inside each of the regions. As a result, each path is characterized by a distribution of travel distance on each region traveled (Batista et al., 2019). The information about trips in the city network is not only useful for determining the paths on the regional networks but also useful to directly characterize their explicit distributions of travel distances.

The question is then how to determine a set of trips that is representative of the travelers' daily trip patterns in the city network. One alternative is to utilize real traffic data, such as vehicle trajectories gathered from license plate recognition data (Dixon & Rilett, 2002; Mo et al., 2020) or traffic counts (Cantelmo et al., 2018; Wen et al., 2018). However, the full information about trip patterns is typically unknown, and only a partial set of trips is available at best. Moreover, the representativeness of such a partial set is not guaranteed by any means. Alternatively, in the absence of any real data, one could develop a methodology to sample virtual trips in the city network (Batista & Leclercq, 2019; Batista et al., 2019). One solution would be to consider all possible shortest trips in distance or time that connect all origin–destination (od) pairs in the network. However, this would require a large number of computational resources, especially for large metropolitan areas, as the computational power required rapidly increases with the number of nodes of the city network (Kuehnel et al., 2020). Instead, Batista et al. (2019) used Monte Carlo simulations to randomly choose a smaller sample of od pairs in the city network, for which to determine the shortest trips in distance or time. Herein, we refer to such an approach as the *benchmark method*. It requires a complex calibration process to

ensure good network coverage (Batista et al., 2019), which makes it unfeasible for large networks. This task becomes even more complex when shortest trips in time are to be considered, as those are dynamic and need to be updated often.

In this paper, we develop an alternative and more robust methodological framework based on *Gaussian processes*, to address these limitations. *Gaussian processes* (Rasmussen & Williams, 2006) are a stochastic nonparametric Bayesian approach used for solving regression and classification problems. They have been used in many applications in traffic-related problems, such as the prediction of public transport flows in special events (Rodrigues et al., 2017), traffic volumes (Xie et al., 2010), mobility on demand (Chen et al., 2015), or real-time traffic (Min & Wynter, 2011), to name a few. Here, we focus on the application of *Gaussian processes* for solving regression problems, that is, kernel regression. We present a heuristic model that explicitly leverages the probabilistic nature of the *Gaussian process* and exploits its properties when sampling trips in the city network. However, we formulate the application of *Gaussian processes* differently than in classical kernel regression problems. We define the city network as a large *Gaussian process* and focus only on the characteristics of the supply to determine a set from all possible od pairs connecting all origin and destination regions. The developed heuristic model iteratively utilizes *Gaussian processes* to model and reduce the uncertainty associated with a specific data set. Additionally, the heuristic leverages the concept of *lazy Gaussian processes* (Ram et al., 2019) to ensure a fast computation even for large networks. The main contributions of this research are fourfold. First, we develop a heuristic model based on the application of *Gaussian processes* for determining synthetic sets of trips using only supply-related information, showing how variability in data can be converted into quantified uncertainty in estimates of travel distances. Second, we analyze the computational efficiency and robustness of the developed heuristic model in comparison to the *benchmark method* to determine synthetic sets of trips. Third, we show how the choice of the kernel function influences the performance of the developed heuristic model. Fourth, we investigate the scalability of the developed heuristic model to account for approximated methods to estimate the set of hyperparameters.

The remainder of this paper is organized as follows. In Section 2, we introduce the formulation of the *benchmark method* used in the literature. In Section 3, we discuss the developed heuristic model based on *Gaussian processes*, as well as a solution algorithm. In Section 4, we evaluate the performance of both methods using two city networks. In Section 5, we summarize the main conclusions of this paper and discuss several lines for future research as well as the potential and applicability of the described methodological framework beyond what is presented in this paper. In the Appendix, we introduce a table of nomenclature used in this paper.

2 | BENCHMARK METHOD

This paper focuses on the calculation of a set of virtual trips χ^{OD} in the city network, that is representative of the set η^{OD} encompassing all possible combinations of od pairs connecting all origin and destination regions. We describe η^{OD} mathematically as

$$\eta^{OD} = \left\{ \bigcup_{\substack{i \in \eta^O \\ j \in \eta^D}} (i,j) \right\}, \quad \forall (O,D) \in W \tag{1}$$

where η^O and η^D represent the sets of all nodes in the origin or destination regions, respectively; and W is the set of all regional OD pairs.

Batista et al. (2019) determined the representative sets χ^{OD} that approximate η^{OD}, by randomly sampling origin and destination pairs of nodes in the city network, without accounting for the definition of the partitioning. Instead, in this paper, we perform Monte Carlo sampling of origin and destination pairs of nodes lying inside the respective origin and destination regions and then determine the shortest trips in distance or time between each pair.

The *benchmark method* assumes that all nodes lying inside the origin and destination regions have a similar likelihood of being selected. The probability, $p((i,j) \in \eta^{OD})$, of selecting one generic (i,j) pair of nodes from η^{OD} is then

$$p((i,j) \in \eta^{OD}) = \frac{1}{|\eta^{OD}|}, \quad \forall (O,D) \in W \tag{2}$$

where $|.|$ represents the size of η^{OD}.

The main challenge when implementing the *benchmark method* is determining whether the set χ^{OD} is representative of the full set η^{OD}, for each OD pair. For notation purposes, we define $\mathbf{N}$ as a vector containing the optimal number of od pairs, N^{OD}, listed on each set $\chi^{OD}, \forall (O,D) \in W$:

$$\mathbf{N} = \left\{ \bigcup_{(O,D) \in W} N^{OD} \right\} \tag{3}$$

The optimal size of η^{OD} varies across the different OD pairs. For simplicity and to generalize this procedure across all OD pairs, we define β^{OD} as the ratio between N^{OD} and the size of η^{OD}. Mathematically, the parameter

β^{OD} is defined as

$$\beta^{OD} = \frac{N^{OD}}{|\eta^{OD}|}, \quad \forall (O,D) \in W \tag{4}$$

where $\beta^{OD} \in [1/|\eta^{OD}|, 1]$, since $N^{OD} \geq 1$. The goal is then to determine the value of β^{OD} such that χ^{OD} is representative of η^{OD} for each OD pair. In this paper, since we focus only on the supply side, we consider the distributions of travel distances determined from the virtual trips to infer the representativeness of χ^{OD}.

Definition 1. The set χ^{OD} is representative if the estimated distribution of travel distances $\hat{L}^{OD} = \{\hat{l}_{od}\}, \forall (o,d) \in \eta^{OD} \wedge \forall (O,D) \in W$, determined from its listed virtual trips, represents a good approximation of the target distribution $L^{OD} = \{l_{od}\}, \forall (o,d) \in \eta^{OD} \wedge \forall (O,D) \in W$, determined from the full set of virtual trips listed in η^{OD}.

The procedure to determining the optimal size of χ^{OD} requires the computation of this set for different values of $\beta^{OD}, \forall (O,D) \in W$. Ideally, and to avoid the enumeration of all possible virtual trips, $|\chi^{OD}| << |\eta^{OD}|, \forall (O,D) \in W$. However, small values of β^{OD} cannot ensure sufficient coverage of the city network, so the Monte Carlo sampling of the od pairs (see Equation (2)) could influence the distributions of travel distances $\hat{L}^{OD}$. This means that different trials even with the same β^{OD} could potentially yield significantly different distributions of $\hat{L}^{OD}$. This bias induced by the random sampling decreases as $\beta \to 1$, that is, as we get close to the full enumeration. Therefore, we also have to consider different trials to determine the set χ^{OD} for each of the input β^{OD} values. Let N_{trials} be the number of times that we repeat the experiment (or trial) for each value of β^{OD}. With this procedure, we can also determine how the bias introduced by the random sampling of the od pair influences the distribution of travel distances $\hat{L}^{OD}$. For this analysis, we calculate the interquartile range (IQR) of all distributions of travel distances $\hat{L}^{OD}$ determined for each of the N_{trials}, and each value of β^{OD}. The optimal size N^{OD} corresponds to the β^{OD} value for which the IQR is less than the threshold Φ (m). This threshold is set such that the variability of the IQR of $\hat{L}^{OD}$ is small for the different N_{trials} experiments.

3 | METHODOLOGICAL FRAMEWORK: GAUSSIAN SAMPLING APPROACH

In this section, we start by formulating the network as a machine learning problem and then discuss why we chose *Gaussian processes* as the basis for the developed heuristic model. We also provide a brief introduction to *Gaussian processes*, before presenting the heuristic for obtaining representative sets χ^{OD} based on Gaussian sampling. The last subsection introduces the solution algorithm for the developed heuristic model.

3.1 | Problem formulation

Let $\mathrm{Y} = \{\mathbf{X}, \mathbf{Y}\} = \{x_i, y_i\}, \forall i = 1, \ldots, S$ be the training set of a machine learning problem. The variable X represents the array of S measurements locations, and Y is a vector of measurements made at locations X, for each OD pair. In this paper, we are focusing only on the characteristics of the supply. Therefore, for each OD pair, the array of measurement locations X corresponds to the Cartesian coordinates of the od pairs, and the vector Y to the measured travel distances associated with these od pairs. For each OD pair, the training set Y contains all od pairs for which one has measured the travel distances. Note that, this might not correspond to the full set η^{OD}.

The goal is to predict the travel distances $\mathbf{Y}^* = \{y_i^*\}, \forall i = 1, \ldots, S^*$, associated to another set $\mathbf{X}^* = \{x_i^*\}, \forall i = 1, \ldots, S^*$ that contains the S^* od pairs for which we do not have the measured travel distances. We also define the test set as $\Lambda = \{\mathbf{X}^*, \mathbf{Y}^*\}$. This process is performed in two steps, that is, the *training* and *testing* phases of the model (see Figure 2a).

First, in the *training* phase and given Y, we can define a model $\mathcal{M}(\cdot)$ that describes the relationship between the location of the od pairs and their associated travel distances:

$$\mathbf{Y} = \mathcal{M}(\mathbf{X}, \boldsymbol{\Theta}) \tag{5}$$

where $\boldsymbol{\Theta}$ is a set of model-specific hyperparameters. The model $\mathcal{M}(\cdot)$ can be any supervised learning model, such as neural networks (e.g., Alam et al., 2020; Pereira et al., 2020). Given the training set Y, we can use optimization techniques to train the model $\mathcal{M}(\cdot)$ and estimate the values of $\boldsymbol{\Theta}$ that better reproduce the data. After $\boldsymbol{\Theta}$ has been calibrated, the model $\mathcal{M}(\cdot)$ becomes a generalization of the transport network that can be used to predict the travel distances $\mathbf{Y}^*$ associated to the set $\mathbf{X}^*$ of od pairs:

$$\mathbf{Y}^* = \mathcal{M}(\mathbf{X}^*, \boldsymbol{\Theta}) \tag{6}$$

The accuracy of the prediction model defined in Equation (6) depends entirely on the values of the hyperparameters $\boldsymbol{\Theta}$, which in turn depend on the training set Y. As the set of coordinates $\mathbf{X}$ defines the domain of the function $\mathcal{M}(\cdot)$, Equation (6) will return good estimations if and

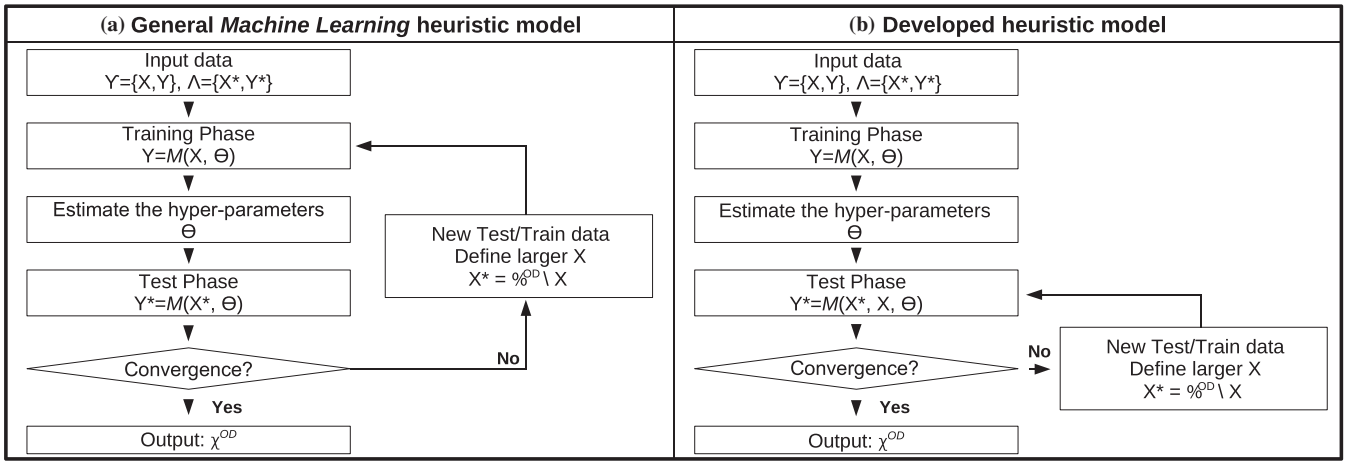

FIGURE 2 (a) General machine learning heuristic model; (b) developed heuristic model

only if $\mathbf{X}^*$ falls into this domain. Fortunately, the domain is limited to the full set η^{OD} of od pairs. Moreover, by definition, the set χ^{OD} is representative of η^{OD}. Hence, if χ^{OD} is used to train the model in Equation (5), then the model defined in Equation (6) should be able to predict the travel distances for all od pairs in η^{OD}.

In this paper, we develop a heuristic model that does not rely on a specific training set but instead learns how to create a representative sample χ^{OD}. This is based on a specific set of rules that we formulate later in this section. The developed heuristic relies on an iterative process. As the domain of η^{OD} is known, the heuristic model starts from a nonrepresentative training set $\mathrm{Y} = \{\Omega^{OD}, \hat{L}^{OD}\}$, where Ω^{OD} contains the selected od pairs. The developed heuristic model then iteratively adds new points that include the Cartesian coordinates of each od pair and corresponding travel distance. Based on this, we can estimate the distribution of travel distances $\hat{L}^{OD}$ at each iteration. Once the convergence is achieved, the representative set is given by $\chi^{OD} = \Omega^{OD}$. Figure 2 depicts a schematic flowchart of the developed heuristic model.

The problem with this representation is that many machine learning algorithms only use the training set Y to estimate the set of hyperparameters $\mathbf{\Theta}$. Once $\mathbf{\Theta}$ has been estimated, Y does not have an active role in the prediction phase, as shown in Equation (6). This leads to the need for training a new model at each iteration as shown in Figure 2a.

To avoid this problem, the heuristic developed here focuses on *Gaussian processes* (Rasmussen & Williams, 2006), instead of other machine learning algorithms. Given a training set Y and the set of hyperparameters $\mathbf{\Theta}$, instead of performing an explicit generalization of the problem, *Gaussian processes* use a kernel function $\mathcal{K}(\mathbf{x}_i, \mathbf{x}_i^*)$ to measure the similarity between a new unseen instance $\mathbf{x}_i^*$ and another instance $\mathbf{x}_i$ already available in the training set. We can then write Equation (6) as

$$\mathbf{Y}^* = \mathcal{M}(\mathbf{X}^*, \mathbf{X}, \mathbf{\Theta}) \tag{7}$$

This process leads to a more efficient framework as depicted in Figure 2b. Moreover, the kernel function $\mathcal{K}(\mathbf{x}_i, \mathbf{x}_i^*)$ provides a useful indicator to determine whether a certain sample Ω^{OD} is representative of the entire domain η^{OD}. To formalize the developed heuristic, we also need to define (i) an objective function (OF) for the convergence of the iterative process and (ii) a set of criteria to select which od nodes should be appended to the set Ω^{OD} at each iteration of the model. While any model based on kernel functions could be used with the developed heuristic algorithm, *Gaussian processes* are a fully probabilistic approach that, given a set of dependent and independent variables, permit to explicitly model the uncertainty σ_{od}^2 associated with each prediction. This is the key advantage of *Gaussian processes* compared to other models based on kernels, and the reason to adopt it in this study. We leverage this unique property of *Gaussian processes* to define the representativeness of a sample (i.e., as the OF of the problem) as well as to select od pairs to be included in the set Ω^{OD} at each iteration. Once the convergence is achieved, the representative set is given by $\chi^{OD} = \Omega^{OD}$. Below, we provide a brief introduction to *Gaussian processes* and discuss the different steps of the developed methodology.

3.2 | Gaussian sampling

Below, we introduce the mathematical formalism of *Gaussian processes*. It assumes that the data points in the training set Y are jointly Gaussian distributed. The predictive function of a *Gaussian process* is defined as the conditional probability of observing some data $\mathbf{Y}^*$ given $\mathbf{X}^*$ and the

training set Y:

$$\mathcal{P}(\mathbf{Y}^*\backslash\mathbf{X}^*, \mathrm{Y}) \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \tag{8}$$

where $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ represent the mean and the covariance of the Normal distributions, respectively. Note that the outcomes $\mathbf{Y}^*$ and $\mathbf{Y}$ are also jointly Gaussian distributed.

The *Gaussian processes* require the definition of a covariance matrix. This matrix is positive semidefined, and the kernel functions respect this property. This covariance matrix serves as a distance function for the developed heuristic model, and it is exploited by the model to determine when a sample is a representative (Heilmann et al., 2011). Therefore, we set $\boldsymbol{\Sigma}$ to be represented by a kernel function $\mathcal{K}_{\mathrm{X},\mathrm{X}^*}$ that defines how the training points $\mathbf{X}$ and the test points $\mathbf{X}^*$ are correlated. Generically, the kernel function $\mathcal{K}_{\mathrm{X},\mathrm{X}^*}$ is a block matrix defined as

$$\mathcal{K}_{\mathrm{X},\mathrm{X}^*} = \begin{bmatrix} \mathcal{K}(\mathbf{X},\mathbf{X}) & \mathcal{K}(\mathbf{X},\mathbf{X}^*) \\ \mathcal{K}(\mathbf{X}^*,\mathbf{X}) & \mathcal{K}(\mathbf{X}^*,\mathbf{X}^*) \end{bmatrix} \tag{9}$$

where the block $\mathcal{K}(\mathbf{X},\mathbf{X})$ is the variance matrix of the training data; the blocks $\mathcal{K}(X,X^*)$ and $\mathcal{K}(X^*,X)$ are the covariance matrices between the training and the test data sets; and the block $\mathcal{K}(X^*,X^*)$ is the variance matrix of the test data set. The kernel function determines the confidence level of the outcome estimation $\mathbf{Y}^*$. This means that if the test and training data are highly correlated, the prediction $\mathbf{Y}^*$ is highly reliable.

The kernel functions are characterized by a set of hyperparameters $\boldsymbol{\Theta}$, that must be calibrated during the *training phase* (see Figure 2b). In this paper, we focus on the conventional radial-based kernel function (RBF) and the Matérn kernel function (MKF) (Rasmussen & Williams, 2006). We then determine the representative set χ^{OD} from the set η^{OD}, during the *test phase*. Following Figure 2b, we apply the *Gaussian processes* in two steps:

1. *Training phase*: Definition of the initial training set Y and estimation of the set of hyperparameters $\boldsymbol{\Theta}$ for each regional OD pair.
2. *Test phase*: Given the hyperparameters and the training set Y, the *test phase* assesses whether the developed heuristic model is a good generalization of the set η^{OD}, that is, able to properly predict the distribution of travel distances L^{OD}.

During the first iterations, the developed heuristic model is not a good representation of the problem. At each iteration, the heuristic creates a new model by changing the training set Y, while keeping the same hyperparameters $\boldsymbol{\Theta}$. The uncertainty σ^2_{od}, that is, the domain knowledge provided by the kernel, is exploited within the optimization process to create a new, more representative training set. This only works because of the way the problem has been formulated. More specifically, (i) the domain η^{OD} is known and (ii) we assume that the initial set of hyperparameters $\boldsymbol{\Theta}$ can approximate the relationship that maps η^{OD} to L^{OD}. For any application when these two conditions do not hold, the developed heuristic model will not be able to provide good results and might instead overfit the data. In the next sections, we describe the two phases of the developed heuristic model in more detail.

3.3 | Training phase: Estimation of $\boldsymbol{\Theta}$

The *training phase* consists of determining the optimal set of hyperparameters $\boldsymbol{\Theta}$ that characterizes the kernel function, given the training data set Y. One possibility is to estimate the posterior distribution over the hyperparameters by utilizing Bayesian inference (Flaxman et al., 2015). However, this might not always be feasible from a pragmatic perspective. Another solution consists in determining the hyperparameters by maximizing the log-likelihood (Rasmussen & Williams, 2006). This requires solving a Gaussian model involving the computation of a kernel function that has the same size as the training data set Y. A practical computational problem arises when the size of Y becomes too large, which means that handling and storing the kernel matrix become quickly unfeasible. The computational requirements of the exact implementation of a *Gaussian process* scale with $\mathcal{O}(S^3)$ for the computational time, and with $\mathcal{O}(S^2)$ for the computational memory (Bauer et al., 2016; Titsias, 2009). This problem depends only on the number of od pairs listed in the training data set Y. In the case of Equation (12), this means storing a covariance matrix $S \times S$ and determining its inverse (see Quiñonero-Candela & Rasmussen, 2005).

Below, we first discuss an approach based on the log-likelihood maximization that is utilized when the size of η^{OD} is manageable. We then discuss approximation methods of *Gaussian processes* that can be utilized within the developed heuristic for determining the hyperparameters $\boldsymbol{\Theta}$ during the *training phase*, for when the size of η^{OD} is too large.

3.3.1 | Log-likelihood maximization

From Equation (8), given a training set Y, the conditional probability of estimating a specific set of hyperparameters $\boldsymbol{\Theta}$ is given by

$$\mathcal{P}(\boldsymbol{\Theta}\backslash\mathrm{Y}) \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \tag{10}$$

The main challenge is to define an initial set Y for estimating the hyperparameters. We thus need to artificially create a training set Y that emulates the characteristics of the problem. We do so with a two-step approach. First, we determine the Cartesian coordinates of the centroid $\bar{\mathbf{C}}_\mathbf{r}$ for each origin and destination pair of regions. If $\bar{\mathbf{C}}_\mathbf{r}$ does not match any of the Cartesian coordinates of the od pairs listed in X, we assign the centroid to the closest od pair based on the following metrics:

$$\min f(\mathbf{X} - \bar{\mathbf{C}}_\mathbf{r}) \tag{11}$$

where $f(\cdot)$ is a performance function measuring the Euclidean distance between each node and the center of the region. One option is to use performance indicators such as the root mean squared error (RMSE) to compute $f(\cdot)$. Alternatively, the kernel function can also be used to compute this distance.

In the second step, we define the training set Y to be used in Equation (10) as

$$\mathrm{Y} = \{\eta^{OD}, \tilde{L}^{OD}\} \tag{12}$$

where $\tilde{L}^{OD} = \{\tilde{l}_{od}\}, \forall (o,d) \in \eta^{OD} \wedge \forall (O,D) \in W$ represents an artificial distribution of travel distances. This distribution is created by assuming that the travel distances $\tilde{l}_{od}$, between all od pairs listed in η^{OD}, are equal to the distance traveled between the centroids of the origin and destination regions. We determine $\tilde{l}_{od}$ based on the calculation of the shortest trip in distance between the centroids of the regions.

Equation (12) returns suboptimal values for the hyperparameters. However, they do represent one of the infinite numbers of possible solutions to the problem, as they can map η^{OD} to $\tilde{L}^{OD}$ and satisfy the conditions imposed by the Gaussian process. Additionally, the artificial training set allows tuning the hyperparameters to the specific network model, as the hyperparameters need to consider whether the information is expressed in meters or kilometers.

3.3.2 | Sparse Gaussian models

Several approximation methods have been described to improve the scalability of the *Gaussian processes*, including *inducing points* (Titsias, 2009), stochastic variational optimization (Hensman et al., 2015), and kernel manipulation (Wang et al., 2019). At the current stage, several approximation methods have been successfully applied to large data sets with several millions of training points (Krauth et al., 2017; Wang et al., 2019). While these models can potentially be used to estimate the initial value of the hyperparameters, approximation methods do not always preserve the properties of the exact formulation of a *Gaussian process*. We then need to investigate whether approximated models can capture the dynamics discussed in the previous section and to estimate the set of hyperparameters $\boldsymbol{\Theta}$, within the context of the developed heuristic model. As finding the best performing model is out of the scope of this paper, we use *sparse Gaussian processes* to handle large data sets. This family of models is the most widely adopted in the literature.

Sparse models use a small set of training points, called *inducing points*, M as support (or inducing) variables, leading to a computational time that scales with $\mathcal{O}(M \times S^2), M < S$ (Bauer et al., 2016). In Sparse models, inducing variables are treated exactly, while all the remaining variables are approximated. There are several sparse approximations discussed in the literature. In this paper, we focus on two of the most common models, namely, the *subset of data* (*SoD*) and the *fully independent training conditional* (*FITC*). In both these models, we replace the set η^{OD} in Equation (12) by a smaller set of observations Γ^{OD}, such that $|\Gamma^{OD}| \leq |\eta^{OD}|$.

In the *SoD* (Bauer et al., 2016), one can manually (or randomly) choose M od pairs of nodes from η^{OD}, and append them to the set Γ^{OD}. However, the reliability of the hyperparameters determined using the *SoD* strongly depends on how well the set Γ^{OD} approximates η^{OD}.

In the *FITC* (Quiñonero-Candela & Rasmussen, 2005), the M *inducing points* are randomly extracted (e.g., from the training set) and then optimized over the data (see Equation (10)). As different *inducing points* lead to different predictions, the *FITC* treats them as hyperparameters that are jointly estimated with $\boldsymbol{\Theta}$. These *inducing points* will define the set Γ^{OD}. We then rewrite Equation (10) as

$$\mathcal{P}(\boldsymbol{\Theta}, \Gamma^{OD} \backslash \eta^{OD}, \tilde{L}^{OD}) \sim \mathcal{N}(\mu, \Sigma) \tag{13}$$

Besides treating the *inducing points* as parameters, *FITC* also considers the training points as a function of the *inducing points* (Quiñonero-Candela & Rasmussen, 2005)—hence the term *inducing points*, as they induce the dependencies between training and test points. By using the *inducing points* to quantify the correlation between training and test points, the model uses a simplified marginal likelihood to learn the hyperparameters and the best set Γ^{OD} from the data through a gradient-based optimization.

3.4 | Test phase: Estimation of χ^{OD}

After having calibrated the set of hyperparameters $\boldsymbol{\Theta}$ for each regional OD pair, we need to determine a representa-

tive training set $\Upsilon = \{\Omega^{OD}, \hat{L}^{OD}\}$, given the full set of data points η^{OD}. This is done during the *testing phase* depicted in Figure 2b. During this step, the Gaussian regression will give an estimation of the distribution of travel distances $\hat{L}^{OD}$ associated with the set Ω^{OD} that is iteratively populated, given the test points η^{OD}. Mathematically, we describe this process as the conditional probability of estimating $\hat{L}^{OD}$ given the test points η^{OD}, the training set Y, and the set of hyperparameters $\boldsymbol{\Theta}$:

$$\mathcal{P}(\hat{L}^{OD}\eta^{OD}, \Upsilon, \boldsymbol{\Theta}) \sim \mathcal{N}(\mu, \mathcal{K}(\eta^{OD}, \Omega^{OD}, \boldsymbol{\Theta})) \quad (14)$$

The model defined in Equation (14) provides a good estimation of $\hat{L}^{OD}$, that is, $L^{OD} \approx \hat{L}^{OD}$, if and only if the set Ω^{OD} is representative and the set of hyperparameters $\boldsymbol{\Theta}$ is properly calibrated. The goal is to determine the smallest set Ω^{OD}, for each OD pair, such that the condition from Definition 1 is satisfied. The developed heuristic model starts with a small nonrepresentative training set Y. Then, at each iteration, the heuristic model appends the od pair with the largest uncertainty (or standard deviation) σ^2_{od}, to the set Ω^{OD}. The standard deviation σ^2_{od} is determined using the Kernel function (see Equation (9)). If two or more od pairs have similar uncertainties, the heuristic model randomly chooses one candidate to append to the set Ω^{OD}. The travel distance for the selected od pair is determined based on the calculation of the shortest trip in distance. This travel distance is then appended to $\hat{L}^{OD}$. By selecting the od pair with the largest uncertainty σ^2_{od}, the heuristic model ensures that the overall uncertainty of the system is minimized. This is similar to using a sampling technique that provides the best coverage of the input space (Ge & Menendez, 2017). The question now is to define the stopping criterion of this iterative process, that is, when the training set Y becomes representative. For this, we use the average uncertainty $\overline{\sigma}^2$ of Y at each iteration k, to determine the convergence of the iterative process. This idea is based on traditional techniques for ensuring convergence in optimization (Spall, 2005). We define an OF that we try to minimize, as follows:

$$OF = \frac{\overline{\sigma}^2}{k^{\gamma}} \quad (15)$$

where $\gamma \in [0, 1]$ is a free parameter that must be calibrated. Note that (i) for $\gamma = 0$, the denominator in Equation (15) goes to 1; and (ii) the closer γ is to 1, the faster the convergence is achieved.

The iterative process stops when the OF (Equation (15)) is inferior to a predefined threshold Δ and when for two consecutive iterations k and $k-1$, the estimated distribution of travel distances does not change significantly, that is, $\hat{L}^{OD}_k \approx \hat{L}^{OD}_{k-1}$. For each OD pair, the final representative set χ^{OD} corresponds to Ω^{OD}. The size of Ω^{OD} gives the value of β^{OD}.

Algorithm 1 Implementation pseudo-code of the developed heuristic model

```
Input the city network topology.
Define the convergence threshold Δ.
for each (O, D) ∈ W do
    Determine the od pair of nodes that are the closest to the
      center of gravity of the OD pair of regions (see Equation (11)).
    Determine the shortest-trip between the centroid nodes,
      and construct an artificial distribution of travel distances
      L̃^OD. Determine the set of hyper-parameters Θ,
      following the procedure described in Section 3.3.
    Set k = 1.
    while OF ≥ Δ do
        Create a Gaussian Model as defined in Equation (10).
        Calculate the uncertainty σ²_od, ∀(o, d) ∈ η^OD,
          associated to each od pair listed in η^OD.
        Select the od pair that has the largest uncertainty and
          append to Ω^OD. If there is more than one od pair
          with similar uncertainty, randomly choose one
          candidate to append to Ω^OD.
        Determine the shortest-trip in distance between this
          od pair, and append its travel distance to the
          training set Υ = {Ω^OD, L̂^OD}.
        Evaluate the objective function (see Equation (15)).
        Set k = k + 1
    end
    Set χ^OD = Ω^OD.
end
```

3.5 | Solution algorithm

Algorithm 1 summarizes the implementation procedure of the developed heuristic based on *Gaussian processes* to determine a representative set χ^{OD} of od pairs for each regional OD pair. The heuristic model takes as an input the training set Y as well as the test set Λ, for each OD pair. Then, it determines the od pairs of nodes that are the closest to the centroids of the origins and destinations regions. The heuristic model creates an artificial distribution of travel distances $\tilde{L}^{OD} = \{\tilde{l}_{od}\}, \forall (o, d) \in \eta^{OD} \wedge \forall (O, D) \in W$ by assuming that $\tilde{l}_{od}$ is similar for all od pairs and equal to the travel distance between the centroid nodes of the corresponding origin and destination regions. This travel distance is calculated based on a shortest trip in distance. We then determine the set of hyperparameters $\boldsymbol{\Theta}$ following the discussion of Section 3.3, and depending on the size of η^{OD}. The heuristic model iteratively populates a new training set $\Upsilon = \{\Omega^{OD}, \hat{L}^{OD}\}$, following the procedure described in Section 3.4. This iterative process stops when $OF \leq \Delta$ (see Equation (15)). The final representative set χ^{OD} corresponds to Ω^{OD}. We implement this procedure in Python and utilize the libraries related to *Gaussian processes* (GPy, 2012) called GP_pro and GPy.

The developed heuristic model can be regarded as a *lazy Gaussian process*, following Ram et al. (2019). In a conventional *Gaussian process*, the set of hyperparameters $\boldsymbol{\Theta}$ is constantly updated. This procedure has two major drawbacks. First, the iterative optimization of $\boldsymbol{\Theta}$ is

computationally demanding. Second, updating $\boldsymbol{\Theta}$ in every iteration entails changing the space of solutions, which is not desirable. The *Gaussian process* is called *lazy* when the set of hyperparameters is optimized only once because the model assumes that $\boldsymbol{\Theta}$ are always correct, remaining unchanged throughout all iterations. An advantage of using a *lazy* model is that when $|\eta^{OD}|$ is large, one can use sparse approximated models to calibrate the hyperparameters $\boldsymbol{\Theta}$ while leveraging the exact model to determine a representative set χ^{OD}. However, we should note that when a representative χ^{OD} has been determined, one should calibrate a new set $\boldsymbol{\Theta}$ and then a new set χ^{OD}.

4 | CASE STUDIES AND RESULTS

In this section, we test and discuss the implementation of the *benchmark method* and the *Gaussian processes* methodology on two city networks, for determining the sets $\chi^{OD} \in \eta^{OD}, \forall (O,D) \in W$. First, we introduce the two test networks. Second, we discuss the implementation of the *benchmark method*, putting in evidence its main limitations. Third, we discuss the implementation of the *Gaussian processes*, shedding light on the appropriate choice of the kernel function and discussing the convergence process. Moreover, we test the robustness of the convergence criterion and highlight its advantages compared to the *benchmark method*. Fourth, we discuss the use of the approximation methods *SoD* and *FITC* to handle larger data sets η^{OD} when estimating the set of hyperparameters $\boldsymbol{\Theta}$. We then analyze the performance of both the *benchmark method* and the *Gaussian processes* to estimate the optimal sets $\chi^{OD} \in \eta^{OD}, \forall (O,D) \in W$.

4.1 | Networks settings

We test and discuss the application of the *benchmark method* and the *Gaussian processes* on two city networks, depicted in Figure 3. These networks were gathered from OpenStreetMaps, using the routine OSMnx (Boeing, 2017) developed for Python. They were then processed to correct the network for bugs (e.g., missing links, unconnected links). The sixth district of Lyon (France), depicted in Figure 3a has 757 links and 431 nodes. It is partitioned into eight regions, leading to a total of 64 OD pairs. We use this network to investigate and discuss the implementation of the developed heuristic model to handle smaller sets η^{OD}. The smallest set η^{OD} contains 1299 possible od pairs and corresponds to the OD pair 22, with the first and last digits referring to the Origin and Destination regions, respectively. The largest set has 6014 possible trips and corresponds to the OD pair 55. The metropolitan area of Innsbruck (Austria), depicted in Figure 3b, has 1992 nodes and 4448 links. The network is partitioned into four regions, leading to a total of 16 OD pairs. The smallest set η^{OD} contains 140,249 possible od pairs and corresponds to the OD pair 22, while the largest set η^{OD} has 362,153 possible od pairs and corresponds to the OD pair 33. We use this network to investigate and discuss the implementation of the developed heuristic model to handle larger data sets η^{OD}, for the estimation of the hyperparameters set $\boldsymbol{\Theta}$.

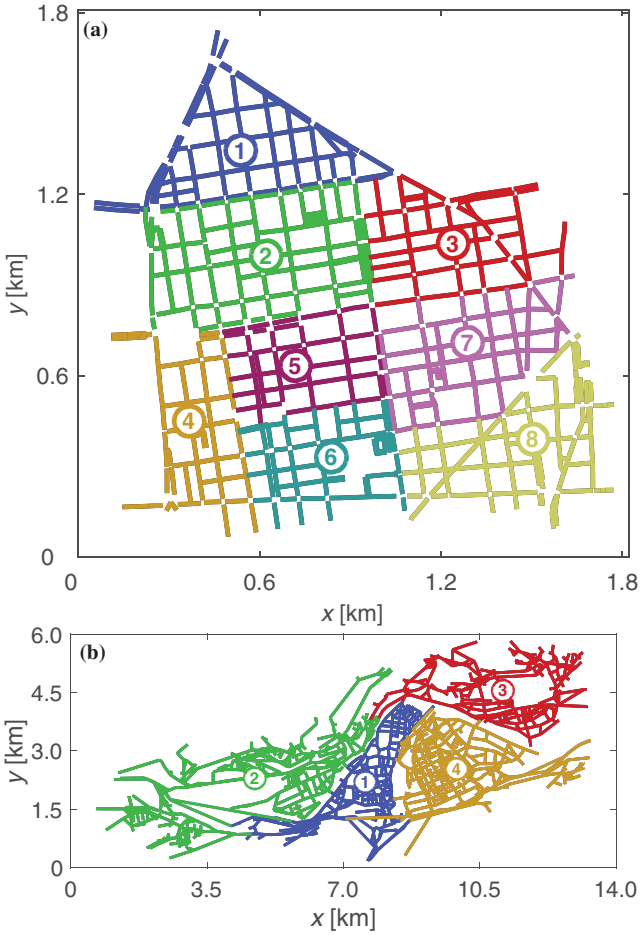

FIGURE 3 (a) Sixth district of Lyon (France) network, partitioned into eight regions; (b) city of Innsbruck (Austria) partitioned into four regions

As a reference for the discussion in this section, we have determined the full set η^{OD} and corresponding distributions of travel distances L^{OD} for all the OD pairs of both networks, based on the calculation of shortest trips in distance. This full enumeration of trips is utilized as a reference, and its computation is only feasible because the city networks are small compared to larger metropolitan areas with more than 20,000 nodes. We discuss in Section 4.5 the computational costs of the full enumeration and the *benchmark method* compared to the developed heuristic model based on *Gaussian processes*.

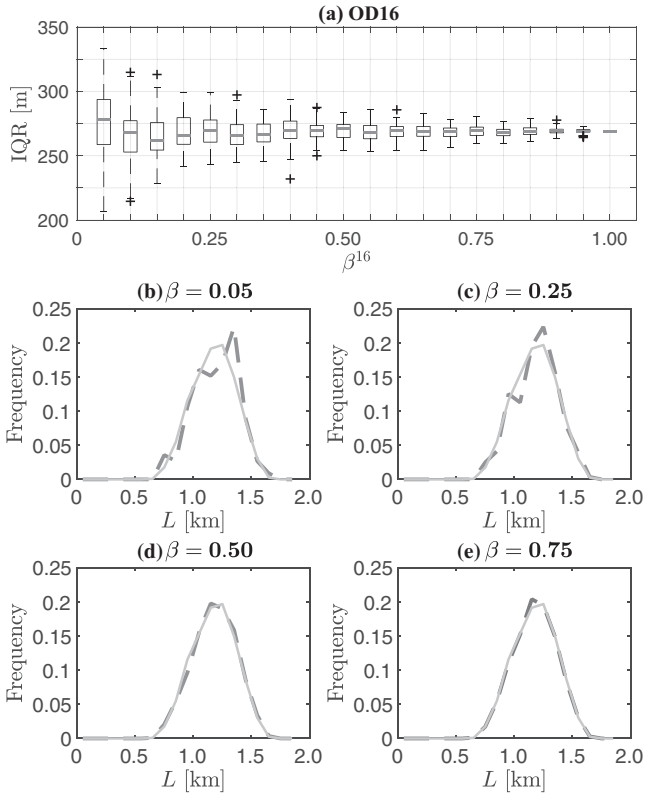

FIGURE 4 (a) Evolution of the IQR of the $\hat{L}^{OD}$ as a function of β^{16}. The estimated $\hat{L}^{OD}$ (dashed line) and target L^{OD} (solid line) density distributions are also depicted for β^{16}: (b) 0.05; (c) 0.25; (d) 0.50; and (e) 1.00

4.2 | Benchmark method

Here, we discuss the application of the *benchmark method* for determining the optimal set χ^{OD}. We focus on the OD pair 16 of the Lyon sixth district network depicted in Figure 3a, but must stress that all other 63 OD pairs show similar trends. We then look at values of β^{OD} starting at 0.05 and increasing until 1 with a step size of 0.05. For each of these β^{OD} values, we run 20 trials (i.e., $N_{trials} = 20$) to determine χ^{OD}, and the IQR of each of the 20 $\hat{L}^{OD}$ distributions.

Figure 4a depicts the box-and-whisker diagrams of the IQRs of the 20 distributions of $\hat{L}^{OD}$, for each value of β^{OD}. We observe that as β^{OD} increases, the variability of the box-and-whisker diagrams decreases and converges to the median of the distribution. Figure 4b–e depicts the estimated distribution $\hat{L}^{OD}$ (dashed line) compared to the target one L^{OD} (solid line), for $\beta^{16} = 0.05, 0.25, 0.50, 0.75$. Note that we have randomly selected one out of the 20 trials to plot these density distributions. One can observe that as β^{16} increases, the distribution $\hat{L}^{16}$ approaches the target L^{16} distribution. In other words, the bias introduced by the random sampling of od pairs in the estimated distribution decreases as the network coverage increases, that is, β^{16} increases.

The question now is how to determine the optimal χ^{16}. We do so by determining the first β^{16} value for which the IQR is less than the predefined threshold Φ. We set $\Phi = 10$, then $\beta^{16} = 0.45$. From Figure 4a, we observe that for $\beta^{16} > 0.45$, the variability of the box-and-whisker diagram is relatively small and close to the median value, and this median remains more or less constant. The optimal size N^{16} is then determined following Equation (4). This procedure is valid for all OD pairs.

This process shows that the *benchmark method* is computationally inefficient for determining χ^{OD}, as it requires the computation of several trials for each of the β^{OD} values considered. This is unfeasible from a pragmatic perspective for large city networks, with thousands of shortest trips in distance or time. In this paper, we set β^{OD} ranging from really small values to the full enumeration, but in practice, we should avoid doing so. We advise the user to start the analysis with $\beta^{OD} = 0.5$ and then consider a smaller and a larger value to evaluate the difference between the IQRs and the threshold Φ. This can be a cumbersome task, showing another major practical limitation of the *benchmark method*. The goal of this paper is to reduce the computational costs by avoiding large β^{OD} values that are close to the full enumeration.

4.3 | Gaussian processes

Below, we discuss the application of the *Gaussian processes* for determining the optimal set χ^{OD} for the sixth district of the Lyon network. We start by investigating the influence of the kernel function on the performance of the *Gaussian processes* methodology. We then test the convergence scheme described in Section 3.4. We focus the analysis on two OD pairs, however, these observations hold for all OD pairs. In this analysis, we calibrate the set of hyperparameters $\boldsymbol{\Theta}$, considering the methodology described in Section 3.3.1 where η^{OD} is used as the training set.

4.3.1 | Influence of the kernel function

Here, we investigate the influence of the Kernel function on the application of the *Gaussian processes* methodology. For this purpose, we focus on the conventional RBF and MKF. These are two of the most commonly used kernel functions to analyze spatial correlations between variables. They also have very similar properties. Intuitively, the RBF is expected to perform better as its radial nature better approximates isotropic networks such as

Manhattan's grid ones. Nevertheless, neither of them will exactly reproduce the complex topological relationships that can occur in real transport networks. Developing ad hoc kernel functions based on topological properties is an option. However, it would require the calculation of adjacency matrices between nodes, demanding a complex and computationally expensive process, which is what we want to avoid in the first place. So, there are three points that we aim to discuss in this subsection. (i) Is the developed approach capable of providing a good approximation for the target distribution L^{OD}? (ii) Which kernel function better approximates the problem? (iii) What are the consequences of adopting a suboptimal kernel function?

To investigate these three questions, we tune the implementation of the *Gaussian processes* for both Kernel functions, using (i) the information retrieved from the full set η^{OD} to define the training set Y and (ii) the methodology described in Algorithm 1 where only 10% of the nodes listed in the full set η^{OD} are considered to estimate $\hat{L}^{OD}$ ($\beta^{OD} = 0.1$). We focus on the regional OD 12. The results are depicted in Figure 5. The dashed lines represent the estimated distribution $\hat{L}^{12}$ by the *Gaussian processes*, while the solid lines represent the target distribution L^{12} determined for the full set η^{OD}. This figure also shows the scatter plots between the predicted $\hat{l}_{od}$ and real l_{od} travel distances, where the black dashed lines represent the case of a perfect estimation. Ideally, all the points would be positioned on this line, which would mean that we have a good estimation. The results are depicted for both the MKF (a and c panels) and RBF (b and d panels) kernels. The results depicted in this figure clearly show that when the full set η^{12} is used as the training set Y, the RBF performs better than the MKF (a and b panels in Figure 5). The estimated $\hat{L}^{12}$ distribution almost perfectly matches the L^{12} one for the RBF kernel. We can also observe that fitting the relationship between the predicted $\hat{l}_{od}$ and real l_{od} trip lengths, for the RBF (see panel b.ii in Figure 5), leads to a correlation coefficient of 0.993; that is, the estimated $\hat{L}^{12}$ distribution is almost a perfect representation of the real L^{12} one. However, as expected, the differences between these two distributions increase when only 10% of the nodes listed in η^{12} are considered as a training set (c and d panels). In any case, RBF seems to provide overall a better representation, as it yields a smoother distribution and seems less prone to systematic errors. Therefore, we focus on the RBF kernel function for the rest of this paper.

We must here remind the reader that the objective of the developed heuristic model is not to estimate the travel distance of a single OD pair but rather the distribution of travel distances for a given pair of regions. This result is achieved often even for small values of the correlation coefficient (e.g., ≈ 0.75, see Figure 5).

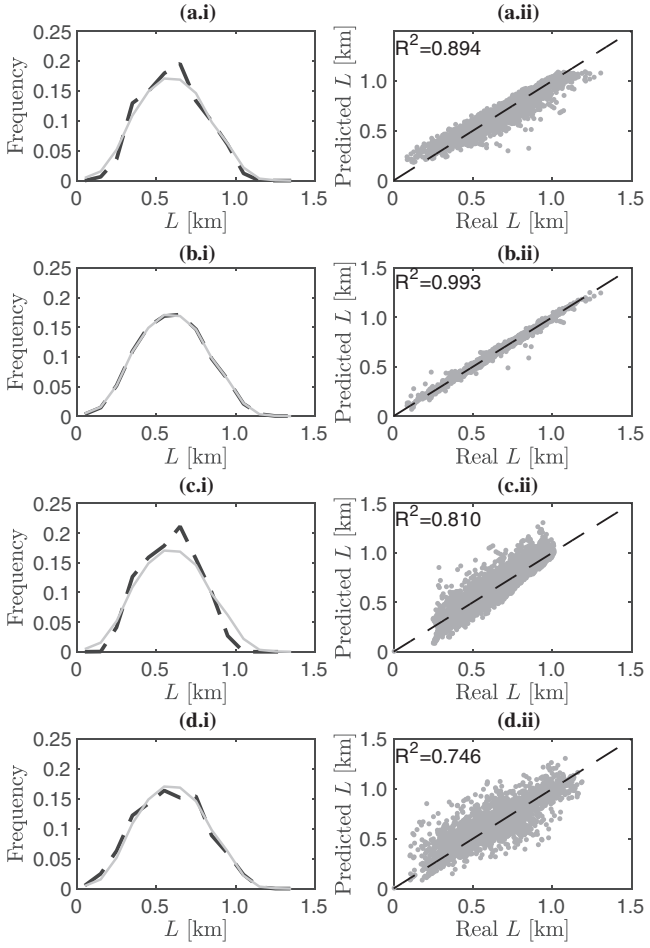

FIGURE 5 Left: Estimated $\hat{L}^{OD}$ (dashed line) and target L^{OD} (solid line) density distributions. Right: Scatter plots that show the relationship between the predicted $\hat{l}_{od}$ and target l_{od} trips lengths. The results are shown for the OD pair 12, and for the following cases: when η^{12} is used as the training set, for both MKF (panel a) and RBF (panel b); and when only 10% of the nodes listed in η^{12} are considered for the training set, for both the MKF (panel c) and RBF (panel d)

4.3.2 | Investigating the convergence

In this section, we discuss how the described convergence criterion for the Gaussian process performs for identifying a representative set χ^{OD}. The discussed convergence criterion (see Equation (15)) accounts for the evolution of the average uncertainty $\bar{\sigma}^2$ of χ^{OD} as a function of the sample size or iteration k. To set up the convergence, we need to calibrate the free parameter γ as well as the convergence threshold or tolerance Δ. In the analysis of this section, we consider four values for calibrating the free parameter $\gamma = 0, 0.2, 0.5, 1.0$. We run the Gaussian process until the full enumeration is achieved for the OD pair 16. For this OD pair, we then investigate the evolution of the convergence

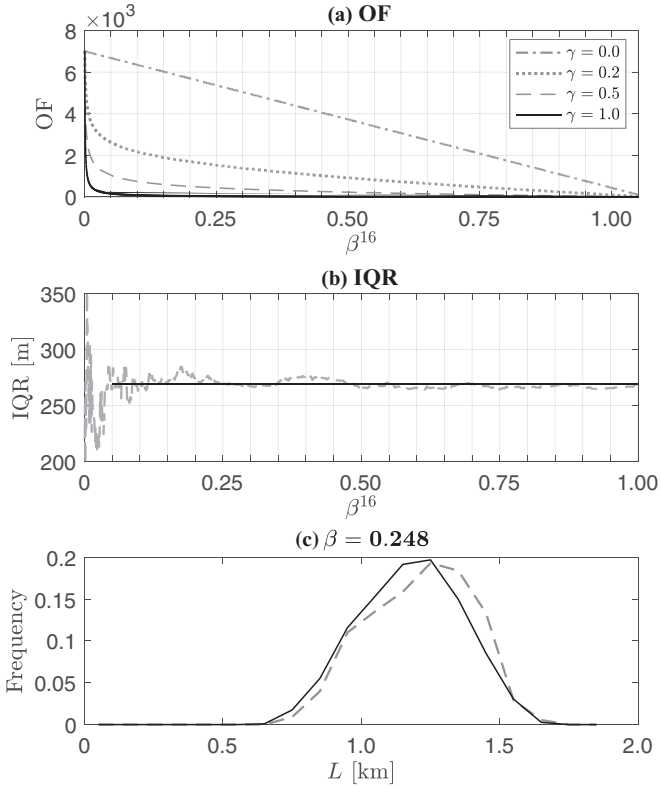

FIGURE 6 (a) Evolution of the OF as a function of β^{16}, for $\gamma = 0, 0.2, 0.5, 1.0$. (b) Evolution of the IQR (m) for the estimated $\hat{L}^{OD}$ for the developed heuristic model (dashed line) as function of β^{16}. The solid line represents the reference IQR of L^{OD} for $\beta^{16} = 1$. (c) Estimated $\hat{L}^{OD}$ (dashed line) and target L^{OD} (solid line) density distributions for $\beta^{16} = 0.248$

criterion for the four settings of γ, as well as the evolution of the IQR of the estimated distributions of travel distances. With this analysis, we can set up the convergence threshold Δ. Note that even though we discuss the case of one OD pair, the analysis is extremely consistent across all 64 OD pairs.

Figure 6a shows the evolution of the OF as a function of β^{OD} for the OD pair 16. Figure 6b shows the evolution of the IQR of the estimated distribution of travel distances $\hat{L}^{OD}$ (dashed line) as a function of β^{OD}, for the same OD pair. The horizontal solid line represents the IQR of the target distribution L^{OD}. We can observe that the developed methodology minimizes the total uncertainty of the system. For $\gamma = 0$, the convergence shows the expected decreasing linear trend as a function of β^{OD}. In this case, the Gaussian model would only be able to identify a representative set χ^{OD} for values of β^{OD} very close to 1, that is, the full enumeration case, which we want to avoid. For larger values of $\gamma = 0.5, 1.0$, the model converges too fast. In this case, it might lead to sets χ^{OD}, which yield an estimated distribution of travel distances $\hat{L}^{OD}$ that is not a good approximation of the target one, that is, L^{OD}. We can observe in Figure 6 that for low β^{OD} values, the IQR still shows a large variability. As β^{OD} increases, the variability of the IQR of the distribution $\hat{L}^{OD}$ decreases, converging to a small constant value as $\beta^{OD} \to 1$, that is, the IQR converges to the median of the distribution. This confirms the previous observation. The setting of the free parameter to $\gamma = 0.2$ leads to a balance trend of the convergence, which occurs neither too fast nor too slow. As such, we set $\gamma = 0.2$.

There are also two other important aspects to analyze regarding the evolution of the IQR. First, the results depicted in Figure 6b show that for low β^{OD} values, the variability of the IQR of the estimated distribution $\hat{L}^{OD}$ by the developed heuristic model for the OD pair 16 decreases faster than the one of the distribution estimated by the *benchmark method*. This gives the first hint that the developed heuristic model can identify a representative set χ^{OD} as well as to provide an accurate estimation of the target distribution L^{OD} for smaller β^{OD} values compared to the *benchmark method*. Second, we can observe that the IQR of the estimated distribution $\hat{L}^{OD}$ by both the developed heuristic model and the *benchmark method* converges to the same value as $\beta^{OD} \to 1$. This shows that when $\beta^{OD} = 1$, the developed heuristic model can perfectly estimate the target distribution L^{OD}. One can also observe that for $\beta^{OD} \geq 0.25$, the estimated distribution $\hat{L}^{OD}$ by the developed heuristic model has an IQR very close to the one of the target distribution L^{OD}, as evidenced in Figure 6b. However, for values of β^{OD} close to 0.25, the box-and-whisker diagrams of the IQRs of the estimated distribution $\hat{L}^{OD}$ for the *benchmark method* still show a large variability around the median value. This variability only decreases for larger β^{OD} values. We remind the reader that the optimal $\beta^{OD} = 0.45$ for the *benchmark method* and the OD pair 16.

We also need to infer the convergence threshold Δ to determine the optimal sets for all the other 63 OD pairs. For this purpose and based on Figure 6b, we observe that the IQR stabilizes around $\beta^{OD} \sim 0.25$ for the OD pair 16. We then check the OF value (see the solid curve in Figure 6) that corresponds to $\beta^{OD} = 0.25$. This leads to a threshold of $\Delta = 1500$, that we fix to determine the optimal sets χ^{OD} for the other OD pairs. We run again the model considering $\Delta = 1500$, verifying that it converges for a very similar $\beta^{OD} = 0.248$. Figure 6c shows that the estimated distribution $\hat{L}^{OD}$ (dashed line) is a good approximation of the target one L^{OD} (solid line), which also shows that the model is robust compared to its inputs.

Figure 6 also shows that the IQR is a good indicator of convergence, which raises the question of whether we should use it. The IQR enables us to infer when the set χ^{OD} is representative, that is, when $\hat{L}^{OD} \approx L^{OD}$.

However, it does not guarantee that the developed heuristic model based on *Gaussian processes* has safely taken into account all the nonlinearities between the data points, that is, two nodes of the network that are close but not connected might be correlated. Additionally, the IQR value also shows a large variability for small β^{OD} values. This could lead to premature convergence. We then use the criterion OF as defined in Equation (15) for the convergence of the developed heuristic model, and the IQR as a quality indicator to determine if the final set $\chi^{OD}, \forall (O,D) \in W$ is representative.

4.4 | Sparse Gaussian models

In this section, we discuss the implementation of the *SoD* and *FITC* models to determine the set of hyperparameters $\boldsymbol{\Theta}$, within the developed heuristic, during the *training phase*. We then use the developed heuristic during the *testing phase* as discussed in Section 3.4 to determine the representative sets χ^{OD}. This is feasible because the size of χ^{OD} appears to be small as discussed in the previous section, meaning that we can still handle the computation of the covariance matrices. The tests are conducted on the Innsbruck (Austria) network, where the size of η^{OD} is larger than 100,000 for all OD pairs. We consider the RBF kernel function, with the set of hyperparameters $\boldsymbol{\Theta}$ properly calibrated, following the discussion of the previous section. Based on a similar analysis of the previous section, we set the free parameter of the convergence function to $\gamma = 0.2$ and the convergence threshold $\Delta = 1500$. To better showcase these results, we define a criterion Ψ as the ratio between the IQR of the distribution of travel distances determined from χ^{OD}, and the one determined from the distribution of travel distances of η^{OD}. This criterion Ψ is mathematically defined as

$$\Psi_{\chi^{OD},\eta^{OD}} = \frac{IQR(\chi^{OD})}{IQR(\eta^{OD})}, \quad \forall (O,D) \in W \tag{16}$$

Ideally, $\Psi_{\chi^{OD},\eta^{OD}}$ would be equal to 1, meaning that the distribution of travel distances determined from the set χ^{OD} is representative of the target distribution determined from η^{OD}. Note that we calculated the IQR of the target distribution of travel distances for all OD pairs, based on the full enumeration of virtual trips, that is, $\beta^{OD} = 1$. Figure 7 depicts the density distributions of $\Psi_{\chi^{OD},\eta^{OD}}$ for all 16 OD pairs as well as for both *SoD* and *FITC* models, used during the *training phase*. As one can observe, both models perform very well, as shown by the distributions of $\Psi_{\chi^{OD},\eta^{OD}}$ that are centered around 1 and have a maximum deviation of 5% among all ODs, as depicted by the histograms. This shows that the distribution of travel distances determined from χ^{OD} approximates well with the one estimated from the full set η^{OD}, and therefore $\hat{L}^{OD} \approx L^{OD}$.

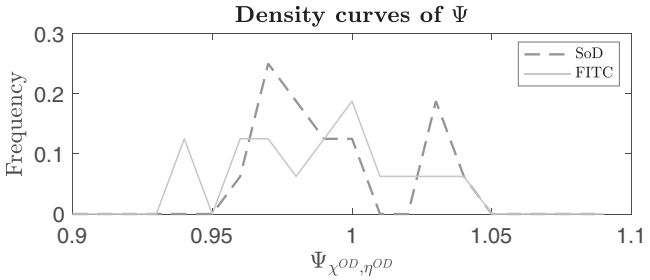

FIGURE 7 Density distributions of Ψ for the *SoD* and *FITC*

These results validate the use of sparse Gaussian models, in particular, the *SoD* and *FITC*, to calibrate the set of hyperparameters $\boldsymbol{\Theta}$ during the *training phase*, for when η^{OD} is large. This shows once again the robustness of the developed heuristic model regarding its inputs.

4.5 | Investigating the performance of both methods

In this section, we discuss the performance of both the *benchmark method* and the developed heuristic model based on *Gaussian processes* for estimating the set χ^{OD} for both city networks depicted in Figure 3. For this, we focus on the distribution of the optimal β^{OD} values determined through both methods. In the case of the network representing the sixth district of Lyon, the methodology used to determine the optimal β^{OD} for all 64 OD pairs follows the discussion of Section 4.2. For the case of the Innsbruck network, we follow the methodology discussed in Section 4.4, where both the *SoD* and *FITC* models are used in the *training phase* to determine the set of hyperparameters $\boldsymbol{\Theta}$, for all 16 OD pairs. For this network, we also apply the *benchmark method*, considering different values of β^{OD} starting at 0.05 and increasing until 1 with a step size of 0.05. For each of these β^{OD} values, we run 20 trials ($N_{trials} = 20$) to determine optimal set χ^{OD}.

Figure 8 depicts the box-and-whisker diagrams of the distribution of the optimal β^{OD} values for both the Lyon sixth and the Innsbruck networks and both the *benchmark method* as well as the developed heuristic model based on *Gaussian processes*. One can observe that the median of the optimal β^{OD} distribution is much smaller for the case of the developed heuristic model based on *Gaussian processes* than for the *benchmark method*. This is true for both city networks and shows that the developed heuristic can identify representative χ^{OD} sets with a smaller β^{OD}. Converging on smaller β^{OD} values provides two advantages. One clear advantage is the reduced computational

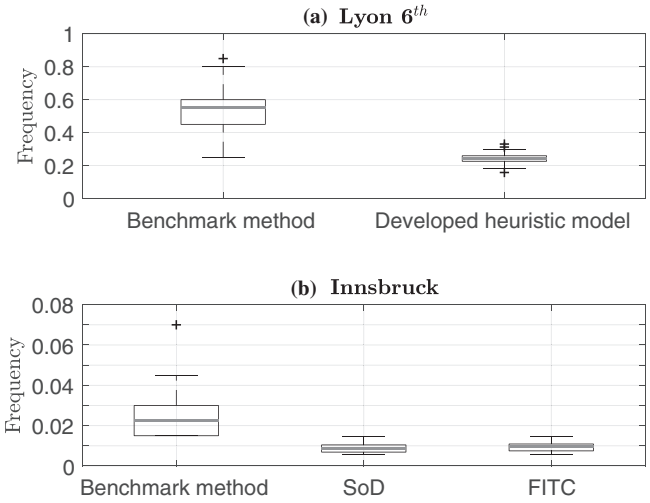

FIGURE 8 Box-and-whisker diagrams of the distribution of the optimal β^{OD} for all possible OD pairs of the (a) Lyon sixth and (b) Innsbruck networks, for both the *benchmark method* and the developed heuristic model

requirements for identifying representative sets χ^{OD}. The second one is the possibility to still use the exact model on a larger network. Moreover, the developed heuristic based on *Gaussian processes* only converges when it has properly accounted for all nonlinearities between the data points, which is another strength of the model.

We also analyze the computational time required for determining the optimal χ^{OD} for all OD pairs of both city networks, using the *benchmark method* and the developed heuristic model. In the case of the sixth district Lyon network, determining the optimal χ^{OD} for all the 64 OD pairs, using the *benchmark method* took $\sim$ 1.5 days, while using the developed heuristic model took less than 1 hr. In the case of the Innsbruck network, the *benchmark method* took $\sim$4 weeks for the 16 OD pairs, while the developed heuristic model took $\sim$1 week. Note that the computational time for the developed heuristic model can be further reduced if one exploits the advantages of parallel computing. We discuss this in more detail in the next section. These times mentioned above were obtained on a MacBook Pro, with a 2.9 GHz Intel Core i9 processor and a memory of 32 GB DDR4 with a frequency of 2400 MHz.

5 | CONCLUSIONS AND DISCUSSION

In this paper, we develop a heuristic model based on the application of *Gaussian processes* to determine synthetic sets of trips, by recognizing that data can be converted into quantified uncertainty. This heuristic model considers only supply-related information, that is, topological features of the city network. We analyze the performance of the developed heuristic model against the *benchmark method*. This permits to extensively test and apply the developed heuristic model without the need to formulate any behavioral assumption regarding the users. The tests are conducted on two networks representing the sixth district of Lyon (France) and the city of Innsbruck (Austria). We show that the presented approach provides a more robust sampling of od pairs than the *benchmark method*, reducing the number of trips that we have to compute and naturally the computational complexity as well. This is evidenced by its ability to efficiently determine representative sets χ^{OD} with smaller β^{OD} than the ones calculated using the *benchmark method*. Moreover, the efficiency of the developed heuristic model is also shown by its ability to provide sets χ^{OD} from which the estimated distribution of travel distances $\hat{L}^{OD}$ is a good approximation of the L^{OD}. We also show that an appropriate choice of the kernel function, able to emulate the characteristics of the city network, is required for the good performance of the developed heuristic model. The developed model can calibrate the set of hyperparameters $\boldsymbol{\Theta}$ for different sizes of the training set Y. In particular, we validate the use of sparse Gaussian models, notably the *SoD* and the *FITC*, to calibrate $\boldsymbol{\Theta}$ during the *training phase* for when the size of Y is large.

This paper represents the first building block for several promising lines for future research. The next natural step is to include features of the demand in the developed heuristic model. The distribution of the demand over the network plays an important role in the calculation of synthetic sets of trips. This is because all od pairs have different prevailing levels related to how the demand is distributed over the network. This induces different demand weights that should be incorporated into the framework. Another discipline where machine learning techniques are widely adopted is traffic state estimation (Laña et al., 2019), where there is an urgent need for efficient algorithms (Dharia & Adeli, 2003) as well as representative samples of data (Benkraouda et al., 2020).

The developed heuristic model is based on an iterative process that utilizes *Gaussian processes* and tries to minimize the average uncertainty of the system associated with the data points. This model is designed to extract transport-related information from a network, and create small yet representative sets. For instance, the proposed iterative procedure permits the determination of representative prevailing mobility patterns, which can be used as a reference to gather or buy crowd-sensed information (such as Google data) or GPS trajectories. However, this implies an additional cost in terms of computation, as the iterative nature of the developed model does not permit to properly exploit the widespread availability of high-performance computing (HPC). We also would like to emphasize that the developed heuristic model can also be utilized to identify

mobility patterns and travel times on other types of data, such as bike-sharing trip data (Cantelmo et al., 2020).

In this paper, we focused on the calculation of shortest trips in distance. However, the developed heuristic model is fully flexible to consider other metrics such as observed travel times instead of measured travel distances or other models that calculate trips in urban networks. For this, it is just necessary to replace the shortest-trip calculation with any other approach discussed in the literature (see, e.g., Fischer, 2020; Flötteröd & Bierlaire, 2013), time-dependent trips (Fakhrmoosavi et al., 2019), or focusing on the most reliable trips (Hadjidimitriou et al., 2015; Zockaie et al., 2016). On the other hand, the implementation of the developed heuristic model assumes independence between the regional OD pairs. This is a fair assumption since we are focusing on the generation of a static synthetic set of trips. However, if one wants to account for the dynamics of the system (i.e., traffic states), this assumption might be too strong. Hence, we propose to formulate this problem at the network level as a future research line, where one can focus on time-dependent synthetic sets of trips and observed travel times as the reference metrics.

In this paper, we have also discussed the calibration of the hyperparameters $\boldsymbol{\Theta}$ and the choice between two kernel functions. Within this topic, there are two natural lines of future research. The first is to investigate the performance of the developed heuristic model for different kernel functions. One can also consider designing new kernel functions for the problem at hand. However, there is still a need to understand how supply and demand–related characteristics influence the optimality of the system. Additional research is needed to include these aspects (temporal and spatial correlations) within the kernel and to consider temporal dependencies within the process, an element that will increase problem sizes and complexity. The second is to improve the robustness of the calibration of the hyperparameters $\boldsymbol{\Theta}$. It would also be interesting to investigate alternative methods to model the hyperparameters for intraregional OD pairs. In this paper, we leverage sparse models, specifically *SoD* and *FITC* approaches, to calibrate the hyperparameters considering large data sets and reduce the complexity to a more manageable $\mathcal{O}(M \times S^2), M < S$. Several approximation methods have been successfully applied to large data sets with millions of training points, reducing the computational time from cubic to quadratic—for example, $\mathcal{O}(S^2)$ (Krauth et al., 2017; Ram et al., 2019; Wang et al., 2019). These new approximations of the sparse Gaussian should be tested in future work.

Overall, the developed heuristic model described in this paper for determining synthetic sets of mobility patterns represents the first building block for several promising lines of research and applications, not only for the Traffic Flow Theory community but also for the Network Modeling community and researchers focusing on the study of urban mobility data.

ACKNOWLEDGMENTS

We thank all the reviewers for their comments and suggestions that have improved the quality of this paper. S. F. A. Batista and Mónica Menéndez acknowledge support by the NYUAD Center for Interacting Urban Networks (CITIES), funded by Tamkeen under the NYUAD Research Institute Award CG001 and by the Swiss Re Institute under the Quantum Cities™ initiative. Guido Cantelmo and Constantinos Antoniou acknowledge funding from the European Union Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement numbers 754462 and MOMENTUM N.815069. We thank Gabriel Tilg for help on gathering and correcting the network of Innsbruck from OpenStreetMaps.

Open access funding enabled and organized by Projekt DEAL.

REFERENCES

Alam, K. M. R., Siddique, N., & Adeli, H. (2020). A dynamic ensemble learning algorithm for neural networks. *Neural Computing with Applications*, *32*, 8675–8690.

Batista, S. F. A., & Leclercq, L. (2019). Regional dynamic traffic assignment framework for MFD multi-regions models. *Transportation Science*, *53*, 1563–1590.

Batista, S. F. A., Leclercq, L., & Geroliminis, N. (2019). Estimation of regional trip length distributions for the calibration of the aggregated network traffic models. *Transportation Research Part B: Methodological*, *122*, 192–217.

Bauer, M., van der Wilk, M., & Rasmussen, C. E. (2016). Understanding probabilistic sparse Gaussian process approximations. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett (Eds.), *Advances in neural information processing systems* (pp. 1533–1541). Curran Associates, Inc.

Benkraouda, O., Thodi, B. T., Yeo, H., Menendez, M., & Jabari, S. E. (2020). Traffic data imputation using deep convolutional neural networks. *IEEE Access*, *8*, 104740–104752.

Boeing, G. (2017). Osmnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks. *Computers, Environment and Urban Systems*, *65*, 126–139.

Cantelmo, G., Kucharski, R., & Antoniou, C. (2020). Low-dimensional model for bike-sharing demand forecasting that explicitly accounts for weather data. *Transportation Research Record*, *2674*(8), 132–144.

Cantelmo, G., Viti, F., Cipriani, E., & Nigro, M. (2018). A utility-based dynamic demand estimation model that explicitly accounts for activity scheduling and duration. *Transportation Research Part A: Policy and Practice*, *114*, 303–320.

Cao, J., Menendez, M., & Waraich, R. (2019). Impacts of the urban parking system on cruising traffic and policy development: The case of Zurich downtown area, Switzerland. *Transportation*, *46*, 883–908.

Chen, J., Low, K. H., Yao, Y., & Jaillet, P. (2015). Gaussian process decentralized data fusion and active sensing for spatiotemporal

traffic modeling and prediction in mobility-on-demand systems. *IEEE Transactions on Automation Science and Engineering*, *12*(3), 901–921.

Daganzo, C. (2007). Urban gridlock: Macroscopic modeling and mitigation approaches. *Transportation Research Part B: Methodological*, *41*, 49–62.

Dharia, A., & Adeli, H. (2003). Neural network model for rapid forecasting of freeway link travel time. *Engineering Applications of Artificial Intelligence*, *16*(7–8), 607–613.

Dixon, M. P., & Rilett, L. R. (2002). Real-time od estimation using automatic vehicle identification and traffic count data. *Computer-Aided Civil and Infrastructure Engineering*, *17*(1), 7–21.

Fakhrmoosavi, F., Zockaie, A., Abdelghany, K., & Hashemi, H. (2019). An iterative learning approach for network contraction: Path finding problem in stochastic time-varying networks. *Computer-Aided Civil and Infrastructure Engineering*, *34*(10), 859–876.

Fischer, S. M. (2020). Locally optimal routes for route choice sets. *Transportation Research Part B: Methodological*, *141*, 240–266.

Flaxman, S., Wilson, A. G., Neill, D. B., Nickisch, H., & Smola, A. J. (2015). Fast Kronecker inference in Gaussian processes with non-Gaussian likelihoods. *Proceedings of the 32nd International Conference on Machine Learning* (Vol. 37). Lille, France. JMLR: W&CP.

Flötteröd, G., & Bierlaire, M. (2013). Metropolis-hastings sampling of paths. *Transportation Research Part B*, *48*, 53–66.

Ge, Q., & Menendez, M. (2017). Extending Morris method for qualitative global sensitivity analysis of models with dependent inputs. *Reliability Engineering & System Safety*, *162*, 28–39.

Geroliminis, N., & Daganzo, C. (2008). Existence of urban-scale macroscopic fundamental diagrams: Some experimental findings. *Transportation Research Part B: Methodological*, *42*, 759–770.

Godfrey, J. W. (1969). The mechanism of a road network. *Traffic Engineering and Control*, *11*, 323–327.

GPy. (Since 2012). *GPy: A Gaussian process framework in Python*. Retrieved from http://github.com/SheffieldML/GPy

Gu, Z., Waller, S. T., & Saberi, M. (2019). Surrogate-based toll optimization in a large-scale heterogeneously congested network. *Computer-Aided Civil and Infrastructure Engineering*, *34*(8), 638–653.

Haddad, J., & Zheng, Z. (2018). Adaptive perimeter control for multi-region accumulation-based models with state delays. *Transportation Research Part B: Methodological*, *137*, 133–153..

Hadjidimitriou, S. N., Dell'Amico, M., Cantelmo, G., & Viti, F. (2015). Assessing the consistency between observed and modelled route choices through GPS data. In *2015 International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)* (pp. 216–222), IEEE.

He, H., Yang, K., Liang, H., Menendez, M., & Guler, S. I. (2019). Providing public transport priority in the perimeter of urban networks: A bimodal strategy. *Transportation Research Part C: Emerging Technologies*, *107*, 171–192.

Heilmann, B., El Faouzi, N.-E., de Mouzon, O., Hainitz, N., Koller, H., Bauer, D., & Antoniou, C. (2011). Predicting motorway traffic performance by data fusion of local sensor data and electronic toll collection data. *Computer-Aided Civil and Infrastructure Engineering*, *26*(6), 451–463.

Hensman, J., Matthews, A., & Ghahramani, Z. (2015, 09–12 May). Scalable variational Gaussian process classification. In G. Lebanon & S. V. N. Vishwanathan (Eds.) *Proceedings of Machine Learning Research* (Vol. 38, pp. 351–360). San Diego, California, USA.

Ingole, D., Mariotte, G., & Leclercq, L. (2020). Perimeter gating control and citywide dynamic user equilibrium: A macroscopic modeling framework. *Transportation Research Part C: Emerging Technologies*, *111*, 22–49.

Ji, Y., & Geroliminis, N. (2012). On the spatial partitioning of urban transportation networks. *Transportation Research Part B: Methodological*, *46*(10), 1639–1656.

Jin, W.-L. (2020). Generalized bathtub model of network trip flows. *Transportation Research Part B: Methodological*, *136*, 138–157.

Krauth, K., Bonilla, E. V., Cutajar, K., & Filippone, M. (2017, 08). Autogp: Exploring the capabilities and limitations of gaussian process models. In *UAI 2017, Conference on Uncertainty in Artificial Intelligence*, August 11–15, 2017, Sydney, Australia.

Kuehnel, N., Ziemke, D., Moeckel, R., & Nagel, K. (2020). The end of travel time matrices: Individual travel times in integrated land use/transport models. *Journal of Transport Geography*, *88*, 102862.

Laña, I., Lobo, J. L., Capecci, E., Del Ser, J., & Kasabov, N. (2019). Adaptive long-term traffic state estimation with evolving spiking neural networks. *Transportation Research Part C: Emerging Technologies*, *101*, 126–144.

Lopez, C., Leclercq, L., Krishnakumari, P., Chiabaut, N., & van Lint, H. (2017). Revealing the day-to-day regularity of urban congestion patterns with 3D speed maps. *Scientific Reports*, *7*, 1–11.

Mariotte, G., Leclercq, L., Batista, S., Krug, J., & Paipuri, M. (2020). Calibration and validation of multi-reservoir MFD models: A case study in lyon. *Transportation Research Part B: Methodological*, *136*, 62–86.

Min, W., & Wynter, L. (2011). Real-time road traffic prediction with spatio-temporal correlations. *Transportation Research Part C: Emerging Technologies*, *19*(4), 606–616.

Mo, B., Li, R., & Dai, J. (2020). Estimating dynamic origin–destination demand: A hybrid framework using license plate recognition data. *Computer-Aided Civil and Infrastructure Engineering*, *35*(7), 734–752.

Pereira, D. R., Piteri, M. A., Souza, A. N., Papa, J., & Adeli, H. (2020). FEMA: A finite element machine for fast learning. *Neural Computing and Applications*, *32*, 6393–6404.

Quiñonero-Candela, J., & Rasmussen, C. E. (2005). A unifying view of sparse approximate Gaussian process regression. *Journal of Machine Learning Research*, *6*(Dec.), 1939–1959.

Ram, R., Müller, S., Pfreundt, F.-J., Gauger, N. R., & Keuper, J. (2019). Scalable hyperparameter optimization with lazy Gaussian processes. In *2019 IEEE/ACM Workshop on Machine Learning in High Performance Computing Environments (MLHPC)*, (pp. 56–65), Denver, CO, USA.

Rasmussen, C. E., & Williams, C. K. I. (2006). *Gaussian processes for machine learning*. The MIT Press.

Ren, Y., Hou, Z., Sirmatel, I. I., & Geroliminis, N. (2020). Data driven model free adaptive iterative learning perimeter control for large-scale urban road networks. *Transportation Research Part C: Emerging Technologies*, *115*, 102618.

Rodrigues, F., Borysov, S. S., Ribeiro, B., & Pereira, F. C. (2017). A Bayesian additive model for understanding public transport usage in special events. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *39*(11), 2113–2126.

Saeedmanesh, M., & Geroliminis, N. (2017). Dynamic clustering and propagation of congestion in heterogeneously congested urban traffic networks. *Transportation Research Part B: Methodological*, *105*, 193–211.

Sinha, K. C., Mahmassani, H. S., Mekemson, J. R., & Hanscom, E. W. (1980). *Use of synthetic demand modelling techniques in transportation planning for small urban areas in Indiana* (Report Nr. FHWA/IN/JHRP-80/9). Joint Highway Research Project West Lafayette, IN.

Sirmatel, I. I., & Geroliminis, N. (2019). Nonlinear moving horizon estimation for large-scale urban road networks. *IEEE Transactions on Intelligent Transportation Systems*, *21*(12), 4983–4994.

Spall, J. C. (2005). *Introduction to stochastic search and optimization: Estimation, simulation, and control* (Vol. 65). John Wiley & Sons.

Titsias, M. (2009). Variational learning of inducing variables in sparse Gaussian processes. In *Twelfth International Conference on Artificial Intelligence and Statistics*, (AISTATS), JMLR: W&CP 5, (pp. 567–574), PMLR, Florida USA.

Vickrey, W. (2020). Congestion in midtown Manhattan in relation to marginal cost pricing. *Economics of Transportation*, *21*, 100152.

Wang, K., Pleiss, G., Gardner, J., Tyree, S., Weinberger, K. Q., & Wilson, A. G. (2019). Exact Gaussian processes on a million data points. In H. Wallach, H. Larochelle, A. Beygelzimer, F. Alché-Buc, E. Fox & R. Garnett (Ed.) *Advances in Neural Information Processing Systems* (pp. 14648–14659). Curran Associates, Inc.

Wen, T., Gardner, L., Dixit, V., Waller, S. T., Cai, C., & Chen, F. (2018). Two methods to calibrate the total travel demand and variability for a regional traffic network. *Computer-Aided Civil and Infrastructure Engineering*, *33*(4), 282–299.

Xie, Y., Zhao, K., Sun, Y., & Chen, D. (2010). Gaussian processes for short-term traffic volume forecasting. *Transportation Research Record*, *2165*(1), 69–78.

Yang, K., Menendez, M., & Zheng, N. (2019). Heterogeneity aware urban traffic control in a connected vehicle environment: A joint framework for congestion pricing and perimeter control. *Transportation Research Part C: Emerging Technologies*, *105*, 439–455.

Yildirimoglu, M., & Geroliminis, N. (2014). Approximating dynamic equilibrium conditions with macroscopic fundamental diagrams. *Transportation Research Part B: Methodological*, *70*, 186–200.

Zockaie, A., Mahmassani, H. S., & Nie, Y. (2016). Path finding in stochastic time varying networks with spatial and temporal correlations for heterogeneous travelers. *Transportation Research Record*, *2567*(1), 105–113.

How to cite this article: Batista SA, Cantelmo G, Menéndez M, Antoniou C. A Gaussian sampling heuristic estimation model for developing synthetic trip sets. *Comput Aided Civ Inf*. 2021;1–17. https://doi.org/10.1111/mice.12697

APPENDIX

Table A1 summarizes the notations used in this paper.

TABLE A1 Nomenclature used in this paper

o	Origin node in the city network
d	Destination node in the city network
O	Origin region
D	Destination region
W	Set of all OD pairs of the regional network
χ^{OD}	Optimal set of od pairs
η^{OD}	Full set of od pairs
η^{O}	Set of all nodes in the origin region
η^{D}	Set of all nodes in the destination region
N^{OD}	Optimal number of od's for the OD pair
$\mathbf{N}$	Vector containing all optimal N^{OD} values
β^{OD}	Ratio between N^{OD} and the size of η^{OD}
$\hat{L}^{OD}$	Estimated distribution of travel distances from χ^{OD}
$\hat{l}_{od}$	Estimated travel distances for the od pair
L^{OD}	Target distribution of travel distances from η^{OD}
l_{od}	Measured travel distance for the od pair
N_{trials}	Number of trials for the *benchmark method*
Φ	IQR threshold for the *benchmark method*
$\mathrm{Y} = \{\mathbf{X}, \mathbf{Y}\}$	Training set
S	Total number of training points
X	Measurement locations of the data points
Y	Measurements made at locations X
$\Lambda = \{\mathbf{X}^*, \mathbf{Y}^*\}$	Test set
S^*	Total number of test points
$\mathbf{X}^*$	Measurement locations of the test points
$\mathbf{Y}^*$	Measurements made at locations $\mathbf{X}^*$
$\boldsymbol{\Theta}$	Set of hyperparameters.
Ω^{OD}	Set of selected od pairs, that is iteratively populated by the developed model
σ^2_{od}	Uncertainty of each prediction
$\bar{\sigma}^2$	Average uncertainty of all points
$\boldsymbol{\mu}$	Mean vector of the Normal distributions
$\boldsymbol{\Sigma}$	Covariance of the Normal distributions
$\bar{\mathbf{C}}_{\mathbf{r}}$	Centroid of a generic region r
$\tilde{L}^{OD}$	Artificial distribution of travel distances
$\tilde{l}_{od}$	Travel distance between the centroids of the O and D regions
M	Number of inducing points
Γ^{OD}	Set of inducing points
γ	Free parameter of the model
Δ	Convergence threshold of the model
Ψ	Ratio between the IQRs of the predicted and target distributions of travel distances