Transportation Research Part C 161 (2024) 104523

Contents lists available at ScienceDirect

Transportation Research Part C

journal homepage: www.elsevier.com/locate/trc

Correcting missingness in passively-generated mobile data with Multi-Task Gaussian Processes

Ekin Uğurel [a], Xiangyang Guan [b], Yanchao Wang [c], Shuai Huang [d], Qi Wang [c], Cynthia Chen [a,*]

[a] THINK Lab, Civil & Environmental Engineering, University of Washington, Seattle WA 98195, United States
[b] WSP, Atlanta GA 30326, United States
[c] Civil & Environmental Engineering, Northeastern University, Boston, MA 02115, United States
[d] Industrial & Systems Engineering, University of Washington, Seattle, WA 98105, United States

ARTICLE INFO

Keywords:
Mobile data
Human mobility
Travel behavior
Gaussian processes
Multi-task learning
Missing data

ABSTRACT

The prevalence of mobile devices and the ubiquity of network connectivity have generated a massive amount of temporally- and spatially-stamped data. A key characteristic of this mobile data is the prevalence of sparsity—for every recorded user, there is a significant amount of missing data. Sparsity leads to bias in the inferred mobility patterns and thus, correcting bias by imputing the missing data potentially creates opportunities for directly using the corrected mobile trajectory data for large-scale simulations in various applications. We propose a multi-task Gaussian process regression model to correct missingness in mobile data. Gaussian processes (GPs) allow for flexible modeling of diverse (i.e., non-linear, locally periodic) data patterns and quantify prediction uncertainty in an interpretable manner. We develop a methodological framework for applying GPs to mobile data. In doing so, we consider the correlations between users' coordinates (latitudes and longitudes) through multi-task learning and adjust for individual-level differences in data characteristics through parameter initialization and optimization. We introduce and demonstrate the use of smooth (i.e., rational quadratic) and periodic kernels in modeling human mobility data. Relatedly, we analyze our model's parameters and imputation accuracy with respect to different types of trips (e.g. slower vs faster trips). We also demonstrate our model's performance in two experiments with real app-based data, in which it outperforms alternative imputation methods. Our implementation is open-source at https://github.com/ekinugurel/GPSImpute.

1. Introduction

The past two decades have brought a range of technological advancements that have made it easy to gather large sets of time- and location-stamped mobile data. Coupled with the ubiquity of network connectivity, this has led to a massive increase in individual trace sightings. Those datasets include sightings generated by Global Positioning Systems (GPS) or other position-signaling devices, geo-tagged posts from social media platforms, and call detail records (CDRs) derived from triangulation of cellular tower positions. Human mobility patterns inferred from such datasets have been used for many applications, including for example, to quantify urban

* Corresponding author.
E-mail address: qzchen@uw.edu (C. Chen).

https://doi.org/10.1016/j.trc.2024.104523
Received 19 May 2023; Received in revised form 8 February 2024; Accepted 8 February 2024
Available online 7 March 2024

vitality (Sulis et al., 2018), understand commuting patterns (Frias-Martinez et al., 2012), and predict pandemic spreading (Hao et al., 2020).

Unlike household travel surveys, which design and implement a set of pre-defined questions to answer (thus called "actively-solicited data"), mobile data is passively-generated as a byproduct of processes that have little to do with answering research questions (Chen et al., 2016). As a result, while mobile data offers sample sizes orders of magnitudes greater than traditional travel surveys, there exist many issues relating to its representativeness. Chief among these issues is sparsity—for a substantial portion of users, there is a significant amount of missing data. In other words, sparsity occurs when the observation frequency of a device is low.

The observation frequency of mobile data varies greatly across users, periods, and geographies for various reasons. Users commonly have the choice of only allowing location transmissions while apps are in use (rather than continuously in the background). Perceived benefits in network externalities (e.g., increased accuracy for Google Maps travel time estimation) have been linked to greater willingness to provide personal information, such as location data (Kim et al., 2019). Therefore, third-party apps that do not provide any perceived benefits to the data-solicited user may receive limited access to location data. Certain versions of the Android operating system (OS) have also been found to aggressively shut down apps running in the background, preventing continuous data collection (Zhou et al., 2020). In addition to stochastic events like battery drain, user- and OS-based decisions are likely to lead to long observation gaps for users, ranging anywhere from a few hours to multiple days or weeks.

Additionally, app-based datasets have large variations in their intra- and inter-day sparsity—observations tend to cluster temporally, resulting in frequency 'peaks' and 'valleys'. Ban et al. (2018) offer two insights regarding the temporal distribution of observations in app-based datasets: First, during weekdays, observations peak in the morning (7–9 AM) and in the evening (4–6 PM), while during weekends there is only one mid-day peak (12–3 PM). There are also significantly fewer observations overnight than during the day. Second, weekdays have more observations than weekends—Fridays tend to have the most observations, while Sundays tend to have the fewest, with the exception to the rule being holidays. This clustering tendency can be caused by users' mobile device usage patterns or the update frequency of the data provider. The problem is further shrouded by the opaqueness of those providers in disclosing neither the source of the data nor the reason for its generation.

Finally, missingness may also be caused by a range of geographical factors. Shorter gaps in continuity are often the result of temporary signal loss, which may occur in dense urban areas, while traveling through tunnels and other enclosed infrastructure, or due to the "cold-start problem"—when signals are dropped due to a lack of clear line of sight from GPS satellites. Additionally, the "urban canyon effect" describes the tendency of receivers in Global Navigation Satellite Systems (GNSS) to output erroneous location estimations while operating in areas with a high density of tall buildings, which block the satellite's direct line of sight to the receiving device (Ben-Moshe et al., 2011).

When a mobile dataset has missing data, derived mobility patterns are vulnerable to bias, meaning that they may misrepresent individuals' actual movements. Several outputs may be altered due to missing data, such as the duration of the inferred stays (Ban et al., 2018), the extent of one's travel (McCool et al., 2022), and users' interactions with the built environment (e.g., the types of locations visited; Merrill et al., 2020). While short and infrequent observation gaps have minimal influence on derived mobility metrics, longer and more frequent gaps will result in a downward bias (McCool et al., 2022). On the other hand, improvements in data quality, such as decreases in rates of missingness or increases in location data accuracy (i.e., the closeness of observation to the real location), have been linked to more accurate calculations of related mobility metrics, such as home census tracts and trip rates (Ban et al., 2018). Correcting missingness in mobile data is thus important to researchers and practitioners alike.

Before prescribing a method, we briefly outline some of the challenges that are present while modeling individual mobile data. In general, people tend to use various modes of transportation to get around. Different modes have different underlying physics (i.e., average velocity, acceleration behavior), meaning that they possess varying data-generating mechanisms. Additionally, people tend to change modes, either within the same trip or between different trips. These realities suggest that any method to correct missingness needs to be flexible enough to capture different data-generating mechanisms and the transition between these states.

A further complexity involves individuals' heterogeneous travel behavior. Previous works have explored the predictability of human mobility, proposing various models and outlining their properties (Barbosa et al., 2018; González et al., 2008; Song et al., 2010a). In general, however, no single model has outperformed every other model in every context—in contrast, each model has had success in a specific domain application. Thus, a key takeaway is that there may not exist universal "laws" of human mobility—different people have different tendencies to explore and exploit their environments and therefore the distributions of their data are different. Oftentimes, the data distribution depends heavily on context and time (Hills et al., 2015). For example, an individual who has recently moved to a new city may be more interested in trying out a range of restaurants, since they have had no prior experience to guide their preferences. On the other hand, a 15-year resident of a neighborhood may have only one or two restaurants they regularly visit. Clearly, the reverse can also be true.

These individual-level complexities suggest that making rigid assumptions about the shape or form of the underlying distribution of an individual's mobile data may not be a successful approach. Rather, a non-parametric data-driven approach may be a promising one—such models are less restrictive, can accommodate complex patterns, and can better capture underlying relationships in the data, which may involve both temporal and spatial correlations. One example of a family of non-parametric methods includes Gaussian processes (GP). GPs can be understood as a generalization of multivariate Gaussian distributions to infinitely many variables. Two components define a GP—the mean function specifies the mean at any point in the input space, and the covariance function (or kernel) embeds a measure of similarity between any pair of points in a multi-dimensional space. In practice, specifying a kernel that can accurately capture relationships in the data is a rather difficult task. Simultaneously, a fitting kernel can seamlessly capture complex dependencies in mobile data.

In the context of mobile phone trajectory data, which often exhibits highly non-linear and context-dependent human mobility

patterns, Gaussian Processes (GPs) offer an appropriate framework for capturing these complexities. For instance, various travel modes exhibit distinct underlying physics—cars move faster than bikes, and whereas a walker can move freely in any direction in a park, a car is limited to paved roads. GPs can manage these differences with kernel functions, facilitating context-dependent modeling of mobile data points. Moreover, GPs provide not only point estimates but also uncertainty estimates, which are invaluable for downstream analysis. The precision of the mobility patterns derived from trajectory data can be significantly influenced by these uncertainty measures. Additionally, there is an increasing necessity to integrate diverse data types, such as categorical, continuous, or even textual data, and GPs are well-equipped to handle this complexity effectively.

In this study, we propose a generalizable multi-task GP framework to correct missingness in mobile data at the individual level. Our framework imputes a user's latitudes and longitudes simultaneously based on a set of predictor variables (primarily time), hence the word "multi-task" (the modeling of latitudes and longitudes as two tasks). We train the framework's parameters on longitudinal trajectory data. We use the word "generalizable" to refer to two of our framework's properties: (1) its ability to accurately correct varying levels of missingness (i.e. short and long gaps), (2) its propensity to capture changing data-generating mechanisms (i.e. due to mode changes). Both properties are achieved through the specification of an appropriate covariance function and the ensuing parameter optimization process. In this way, the model is kept general enough to apply to any user and unique enough to be accurate.

No model is perfect and GPs are no exception. One notable trade-off of using GPs is its demand on computational resources: though GPs have become more scalable, they can still be computationally demanding for extensive datasets. Our implementation addresses this challenge by leveraging GPyTorch (Gardner et al., 2018), an efficient GP implementation that reduces the asymptotic complexity of exact GP inference from $\mathcal{O}(n^3)$ to $\mathcal{O}(n^2)$. Another trade-off involves model complexity. Setting up GPs requires meticulous attention to kernel functions and hyperparameters, contributing to their perceived complexity. Our study demonstrates that kernel choices are directly related to the underlying behavioral patterns and the estimated hyperparameters exhibit systematic patterns that can be explained by human mobility behaviors (Sections 3.3 and 6.1). Additionally, compared to parametric regression methods, GPs are often seen as "black-box" models compared to parametric models, which might limit the interpretability of the results for some users. However, GPs are considered less of a black-box in comparison to deep neural networks, as choices of basic kernels are based on the nature of the phenomenon of interest (in this context, human mobility patterns) and their hyperparameters such as lengthscale are explainable (they directly tell us the periodicity of the underlying function, which characterizes the periodicity of individual mobility patterns). Additionally, in our context which is imputing missing data in the raw mobility trajectories, interpretability is not a primary concern. This is because the outputs we seek to obtain are repaired trajectories with the missingness filled; those repaired trajectories still need further processing to obtain meaningful mobility metrics such as the number of stays.

We summarize our contributions below:

- We develop a methodological framework to correct missingness in heterogeneous mobile data using a multi-task GP model. The resulting (repaired) data can be used for many transportation and mobility applications downstream. Our model considers the correlations between users' coordinates and adjusts for individual-level differences in data characteristics;
- We analyze our model's key hyperparameters (e.g., lengthscale parameters) and imputation accuracy concerning different types of trips (e.g. slower vs faster trips). Specifically, we show the linkage between trip types and lengthscale parameters;
- We introduce and demonstrate the combined use of smooth kernels (i.e., the rational quadratic kernel) and kernels with periodicities (i.e., the periodic kernel) in modeling human mobility data; and
- We benchmark our model against other missing data imputation methods, demonstrating its effectiveness under a range of missingness scenarios (short and long gaps).

The rest of the paper is organized as follows. Section 2 summarizes related work on missing mobile data imputation. Section 3 outlines our methodology. Section 4 presents our implementation while Section 5 describes the dataset we used in our experiments. Section 6 shows our results. We discuss the implications of our work in Section 7.

2. Related work

2.1. Methods for imputing missingness in the data

In general, missing data imputation methods for mobile data fall into one of three categories: (1) using time-series smoothing and interpolation methods, (2) leveraging external data sources to facilitate co-learning or (3) employing machine learning methods such as kernels or deep learning architectures. The first approach uses existing methods for smoothing and time-series interpolation to impute missing data points. These range from simpler methods like linear interpolation to more complex ones like seasonal autoregressive integrated moving averages (SARIMAX). These models have a long history of being used to impute missing data in a range of contexts (i.e., in hydrology, finance, etc.) and therefore are more mature (Cipra et al., 1995; Huo et al., 2010; Kohn and Ansley, 2012; McCool et al., 2022). Though these methods are relatively interpretable, their interpretability depends on the validity of their assumptions, which tend to relate to the regularity of the data (i.e., seasonality or trend), linear dependence over time, and independent noises. Relatedly, the drawback of these models is that they are often not flexible enough to handle the non-linear nature of individual-level trajectory data.

The second approach leverages external data sources to make inferences about the original dataset. This is an especially prevalent approach when dealing with call detail records—CDRs tend to provide limited spatial and temporal resolution, making it challenging to capture detailed and nuanced mobility patterns. Common external datasets include real-time video feeds, transit agency-issued

smart cards, and even aggregated credit card purchase histories. For example, Zheng et al. (2012) fused vehicle video feeds with smart card and mobile data to develop an algorithm to explore human mobility patterns. Similarly, Gong et al. (2020) developed two indicators to measure the similarity of spatiotemporal trajectories across multiple data streams. The biggest drawback of such approaches is the difficulty of obtaining additional datasets—for many applications, external sources (a) may not overlap spatially or temporally, (b) may be expensive to acquire, or (c) may have privacy limitations. Data acquisition and pre-processing for multiple data sources can also be time-consuming.

Finally, kernel- and deep architecture-based learning methods have also gained traction in the context of missing data. These methods often boil down to one of two approaches. The first involves using kernels or activation functions to predict the similarity between two spatial trajectories, then using the more complete trajectory to fill in gaps in the sparser trajectory. For example, Wang et al. (2020) utilize the head-direction information of trajectories together with the displacement attributed to an attention mechanism to learn from past trajectory points with different priorities. Similarly, Liu and Onnela (2021) bidirectionally sample discrete displacements for missing segments based on similar trajectories in linear time complexity. The drawback of this approach is that even if two trips have the same ground-truth trajectory, varying data sampling rates (due to the data provider) may distort the shape of the trajectory observed in the data. This could lead one to conclude that these two trips are not similar. Unless access to consistently high-sampling rate training data is available, using trajectory shapes as a measure of similarity is liable to produce errors.

The third approach involves using recurrent neural nets (RNN), attention mechanisms, or other related methods to capture long- and short-term temporal dependencies in users' locations. This is the approach we take in this paper, but rather than using time- and resource-intensive deep neural networks, we opt for multi-task GPs. We review two examples of the deep neural network approach. Sun et al. (2021) propose a model that captures complex location transition patterns with graph neural networks and uses two attention mechanisms to capture the multi-level and shifting periodicity of human mobility. However, this model is less adept at fine-grained trajectory recovery, which involves accurately tracking and predicting very specific, detailed movements at a high temporal resolution. On the other hand, Ren et al. (2021) propose a framework that can recover and map match the fine-grained points in trajectories from coarse-grained GPS data. It uses a multi-task sequence-to-sequence learning model to capture the spatial and temporal dependencies of trajectories. However, their model requires knowledge of the underlying street network, which may not be available in a lot of scenarios (relating to the above paragraph on external data sources).

2.2. GPs in transportation and mobility applications

The use of Gaussian processes in transportation and mobility applications is not new. As early as 2009 (a few years after Rasmussen & Williams (2006) released their pivotal book on Gaussian Processes), GPs were being proposed to predict travel times for arbitrary origin–destination pairs (Idé and Kato, 2009). Since then, GPs have been leveraged in a variety of domains, including in traffic operations and forecasting (Xie et al., 2010), transportation system estimation (Liu et al., 2022), and pedestrian behavior modeling (Nasernejad et al., 2021).

Within the field of traffic operations and forecasting, Gaussian Processes (GPs) have emerged as a powerful tool for modeling and predicting traffic dynamics. Yuan et al. (2021) proposed a physics-regularized GP framework, pushing the boundaries of macroscopic traffic flow modeling. Beyond macroscopic flows, GPs have been used to predict fine-grained traffic speeds, as demonstrated by Le et al. (2017). Similar to the method of our paper, Rodrigues et al. (2019) leveraged multi-task GPs to impute missing traffic speeds while considering the spatial dependencies with nearby road segments. They found that using GPs to capture spatial correlations with nearby road segments led to substantial improvements in imputation performance over the benchmark methods.

Concurrently, GPs have demonstrated efficacy in handling uncertainties within transportation systems. Storm et al. (2022), for example, introduced an efficient method for evaluating stochastic traffic flow models using GP-based approximations. This not only enhanced computational efficiency but also refined the accuracy of stochastic traffic models. Expanding on uncertainty management, Steentoft et al. (2023) utilized GPs to provide uncertainty estimates for mobility flows derived from large-scale taxi data. Their approach enhanced the reliability of mobility flow predictions in addition to addressing the critical issue of variable selection in complex transportation datasets.

GPs have also found their use in shared mobility and city-scale travel demand modeling. In addressing the challenges of micromobility planning, Gammelli et al. (2020) focused on improving demand forecasting accuracy with censored data, which inherently contains a biased representation of the true demand due to supply constraints. They employed Gaussian Processes (GPs) to replace the traditional linear functional form in specifying the likelihood of data being censored. This approach enables the model to avoid bias due to censoring and improve the accuracy of demand predictions. On the other hand, Batista et al. (2022) employed GPs for estimating city-scale Origin-Destination (OD) matrices, focusing on supply-related characteristics of urban networks. Their methodology provides an efficient alternative to Monte Carlo simulations, offering computational efficiency through iterative OD pair selection and shortest trip determination. Significantly, their method creates smaller synthetic sets, substantially reducing computational demands.

In our study, we also leverage Gaussian Processes, extending their application to a different facet of travel demand modeling—specifically, addressing missing data in mobile trajectory datasets. There is a thematic resonance between our work and that of Batista et al., particularly in the use of GPs to handle complex urban mobility data. Both studies underscore the flexibility of GPs in urban mobility contexts, whether it be in handling synthesizing trip sets or managing missing trajectory data. Uniquely, our method introduces multi-task learning to exploit correlations in user coordinates while also incorporating a distance-based compression algorithm to reduce the size of our training data, acknowledging the bounded nature of individual mobility patterns as suggested by González et al. (2008). Furthermore, our model utilizes a unique kernel designed to align with the characteristics of our dataset as well

as findings from the human mobility literature, which suggest varying levels of regularity in individual travel behavior patterns (Kitamura and Van Der Hoorn, 1987; Song et al., 2010b; Teixeira et al., 2021). This tailored kernel selection echoes the importance of kernel choice in GP models, as also highlighted by Batista et al. in their work.

3. Methodological framework

Our imputation workflow has three modules: Data Preprocessing, Model Development, and Gap Imputation, described in detail in Fig. 1. The first module filters out erroneous and noisy data points, compresses a user's trajectory using a pairwise distance-based algorithm, and normalizes a user's coordinates. Model Development identifies appropriate kernel functions and learns related parameters based on the marginal log-likelihood (the loss function) concerning the training data. Given a set of missing time inputs, Gap Imputation predicts the most likely location for those inputs.

3.1. Spatiotemporal modeling of human mobility with Multi-Task GP learning

We adopt a multi-task GP framework to capture the correlations between two highly correlated tasks in mobile data: predicting latitudes (ϕ) and longitudes (λ). Given a set $\mathbf{X}$ of n inputs $\mathbf{x}_1, \cdots, \mathbf{x}_n$, we define $\mathbf{Y} = \begin{bmatrix} \mathbf{y}_\phi \\ \mathbf{y}_\lambda \end{bmatrix}^{\mathrm{T}} = \begin{bmatrix} y_{1\phi}, \cdots, y_{n\phi} \\ y_{1\lambda}, \cdots, y_{n\lambda} \end{bmatrix}^{\mathrm{T}}$, where y_{ij} is the response for the j^{th} task at i^{th} observation, and j refers to either latitudes ϕ or longitudes λ.

We assume the underlying data generation process for y_{ij} as follows:

$$y_{ij} = f_j(\mathbf{x}_i) + \epsilon_{ij} \tag{1}$$

where f_j is the latent function mapping inputs $\mathbf{x}_i$ to outputs y_{ij} and ϵ_{ij} is a white noise process associated with the j^{th} task and $\epsilon_{ij} \sim \mathcal{N}(0, \delta_j^2)$ are independent random variables. y_{ij} is assumed to be normally distributed, or $y_{ij} \sim \mathcal{N}(f_j(\mathbf{x}_i), \delta_j^2)$.

In multi-task Gaussian process learning, a multivariate normal distribution is used as the prior for modeling multiple outputs (in our case, there are two: latitudes and longitudes). In this setting, we can consider both the temporal correlation within a task but also the correlation between tasks. Thus, we denote the covariance matrix for all n observations and all m tasks as $\mathbf{K}$, which can be expressed as follows:

$$\mathbf{K} = \mathbf{K}^f(\mathbf{y}_\phi, \mathbf{y}_\lambda) \bigotimes \mathbf{K}^x(\mathbf{X}, \mathbf{X}) \tag{2}$$

where $\bigotimes$ is the Kronecker product, $\mathbf{K}^x$ is the covariance matrix of the training data for temporal correlations (n by n), $\mathbf{K}^{\mathrm{f}}$ is the inter-task covariance matrix (Bonilla et al., 2008). In our context where $m = 2$, the resulting covariance matrix $\mathbf{K}$ is $2n$ by $2n$. The hyperparameters associated with $\mathbf{K}^x$ and $\mathbf{K}^{\mathrm{f}}$ can be estimated by minimizing the negative marginal log-likelihood of the dataset (see Section 3.4).

In inference, for a new input $\mathbf{x}_*$, the predictive distribution of the output $\mathbf{y}_*$ is also Gaussian and can be computed as[1]

$$\mathbf{y}_* \big| \mathbf{x}_*, \mathbf{X}, \mathbf{Y} \sim \mathcal{N}(\boldsymbol{\mu}_*, \boldsymbol{\sigma}_*^2) \tag{3}$$

where

$$\boldsymbol{\mu}_* = (\mathbf{k}_j^f \bigotimes \mathbf{k}_*)\left(\mathbf{K}^f \bigotimes \mathbf{K}^x + \mathbf{D} \bigotimes \mathbf{I}\right)^{-1} \mathrm{vec}(\mathbf{Y}), \tag{4}$$

$$\boldsymbol{\sigma}_*^2 = \left(\mathbf{k}_j^f \bigotimes \mathbf{k}_{**}\right) - \left(\mathbf{k}_j^f \bigotimes \mathbf{k}_*\right)\left(\mathbf{K}^f \bigotimes \mathbf{K}^x + \mathbf{D} \bigotimes \mathbf{I}\right)^{-1}\left(\mathbf{k}_j^f \bigotimes \mathbf{k}_*\right).$$

Here, $\mathbf{k}_j^f$ selects the j^{th} column of $\mathbf{K}^f$, $\mathbf{k}_* = \mathrm{k}(\mathbf{x}_*, \mathbf{X})$ is the covariance vector between the test point and the training inputs, $\mathbf{k}_{**} = k(\mathbf{x}_*, \mathbf{x}_*)$ is the auto-covariance of the test point, $\mathbf{D}$ is a 2×2 diagonal matrix with the variances of the noise processes for latitude and longitude, and $\mathbf{I}$ is the identity matrix.

3.2. Using kernels to model human mobility

Human mobility observations at nearby time inputs tend to be spatially close to one another. This is an expanded application of Tobler's first law of geography—that near things are more related than distant things. To model this phenomenon, we use the rational quadratic kernel:

$$k_{\mathrm{RQ}}(x, x') = \Omega^2 \left(1 + \frac{(x - x')^2}{2\alpha\ell^2}\right)^{-\alpha} \tag{5}$$

[1] For more background on the derivation of these equations, refer to Section 2.2 on Rasmussen and Williams (2006).

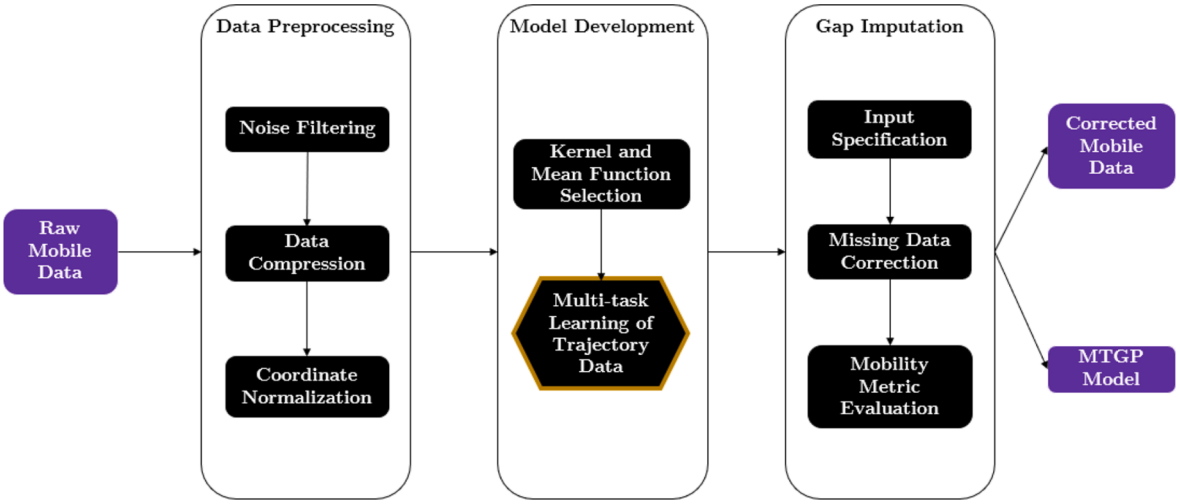

Fig. 1. Overview of imputation workflow for passively-generated mobile data with varying levels of missingness (long and short gaps). Data Preprocessing reduces the computational complexity of the imputation task while also improving the accuracy of training points. Model Development specifies the kernel and mean functions and learns the kernel parameters through marginal log-likelihood maximization. Gap Imputation "corrects" the raw data by imputing the missing locations.

where x and x' are two inputs, Ω is a scale parameter, determining the average distance of the function away from its mean, $\alpha > 0$ is the scale mixture parameter, determining the relative weighting of large-scale and small-scale variations in the data, and $\ell > 0$ is the lengthscale, specifying the smoothness of the function (i.e., the frequency of the gradient of the function to change sign). The lengthscale is commonly interpreted as the maximum distance the model can extrapolate beyond the training data without reverting to the mean. Note that as x and x' approach each other, the term with the exponent $-\alpha$ goes to 1, signifying the increasing likelihood of nearby points to covary.

This works fine for smooth trajectories with relatively low levels of sparsity. For prediction regions in which we have little-to-no observations, data from other days, weeks, and months for the same individual can be leveraged to improve model performance. This is possible because human mobility data exhibit patterns of periodicity. Previous literature suggests that people's mobility patterns exhibit history dependency, meaning that where one is at time t depends upon their location at $t-1$ (Kitamura and Kermanshah, 1983). More specifically, activity locations center around frequently- and recently-visited locations such as home and work (Barbosa et al., 2015; Song et al., 2010a; Teixeira et al., 2021). There are rhythms of activity and travel patterns—people usually go to their workplaces and return home every day, inducing correlations between the days of the week. Other behavior like grocery shopping, going out with friends, or attending family gatherings happens less frequently, and hence has longer intervals between consecutive instances. These activities tend to be correlated between the weeks of the month and sometimes even the months of the year (i.e., Christmas, Thanksgiving, etc.).

To reflect these patterns, we incorporate categorical variables like days and weeks by representing them as sets of binary variables, using a one-of-k encoding. For example, as the days of the week x_d can take one of seven values, $x_d \in \{M, Tu, W, Th, F, Sa, Su\}$, then a one-of-k encoding of x_d will correspond to seven binary inputs and one-of-k $(Tu) = [0, 1, 0, 0, 0, 0, 0]$. One approach to embed multiple inputs in a GP framework is to multiply kernels defined on each individual input. This family of kernels is called Automatic Relevance Determination (or ARD), so named due to the existence of a different lengthscale parameter for each input dimension d (Duvenaud, 2014). In this case, the kernel function $\mathbf{K}_{\mathrm{RQ}}$ in Equation (5) becomes (for notational simplicity we drop the subscript i in $\mathbf{x}_i$):

$$k_{\mathrm{RQ-ARD}}(\mathbf{x}, \mathbf{x}') = \prod_{s=1}^{S} \Omega_s^2 \left(1 + \frac{(x_s - x_s')^2}{2\alpha \ell_s^2}\right)^{-\alpha} \tag{6}$$

where ℓ_s is the lengthscale parameter for input dimension s. Thus, if the optimal lengthscale for a given categorical variable (i.e., day of the week) is small, the model has determined relatively low correlation between data in that category. On the other hand, categorical input dimensions with large lengthscales imply relatively little variation (in other words, high correlation) along those dimensions in the output variable (i.e., locations).

In addition to a one-of-k encoding strategy, we also use periodic kernels, which allow GPs to model functions that repeat themselves:

$$k_{\mathrm{PER}}(x, x') = \Omega^2 \exp\left(-\frac{2\sin^2(\pi|x - x'|/p)}{\ell^2}\right) \tag{7}$$

where the period length p determines the distance between repetitions of the function and the lengthscale ℓ works in the same way as

in the RQ kernel. The incorporation of the periodic kernel captures the various rhythmic patterns revealed by human mobility (people tend to conduct different activities and trips at varying frequencies). We note that the periodic kernel works best for continuous input dimensions that are unbounded in the input space.

3.3. Kernel and mean function selection

Before prescribing a covariance function, we give some intuition on algebraic operations with kernels. The set of positive semi-definite kernels is closed under sum and product operations.[2] Multiplying two kernels can be interpreted as an AND operation while adding two kernels can be interpreted as an OR operation (Duvenaud, 2014). Let k_1 and k_2 be kernels each of which depends on a single input vector, $\mathbf{x}$ and $\mathbf{y}$, respectively. The product of k_1 and k_2 will result in a prior over functions that vary across both $\mathbf{x}$ and $\mathbf{y}$ and hence the function value $f(x_i, y_i)$ is only expected to be similar to some other function value $f(x_g, y_g)$ if x_i is close to x_g AND y_i is close to y_g. On the other hand, the sum $k_1 + k_2$ will result in a prior over functions which are the sum of one-dimensional functions, and hence the function $f(\mathbf{x}, \mathbf{y}) = f_x(\mathbf{x}) + f_y(\mathbf{y})$.

We design a composite kernel that is the sum of two nonlinear product kernels. We multiply the RQ-ARD kernel (introduced earlier) with a periodic kernel and then add a second identical component:

$$\mathbf{K}(\mathbf{x}, \mathbf{x}') = \eta_1(\mathbf{K}_{\mathrm{RQ-ARD}} \times \mathbf{K}_{\mathrm{PER}}) + \eta_2(\mathbf{K}_{\mathrm{RQ-ARD}} \times \mathbf{K}_{\mathrm{PER}}) \tag{8}$$

where $\eta = \Omega^2$ denotes the weight of a kernel component.

As described by Duvenaud (2014), the product of a smooth kernel (like the RQ kernel) and the periodic kernel results in functions that are periodic but can slowly vary over the input space—that is, the shape of the repeating part of the function can change. This is an appropriate kernel to use as human mobility (as expressed in the way one's latitude and longitude changes over time) tends to be periodic (i.e., home-work-home routine), but with slight variations over time (Kitamura and Van Der Hoorn, 1987; Song et al., 2010b; Teixeira et al., 2021).

The two identical product terms in Equation (8) are to capture periodic patterns with different scales: the first periodic kernel is initialized with $p = 24$ hours (aiming to capture the home-to-work-to-home mobility pattern), while the second is initialized with $p = 7$ days (aiming to capture the weekly ebbs and flows of mobility). η_1 and η_2 are weights associated with the two product terms, respectively. When $\eta_1 = \eta_2$, the model places equal emphasis on the daily and weekly components. When $\eta_1 \gg \eta_2$, the model identifies that the daily pattern is much more pronounced one than the weekly one and vice versa for $\eta_1 \ll \eta_2$. We constrain the weights such that their sum is always equal to one. We initialize the two components with different period lengths. In estimation, these two weights are treated as hyperparameters, like the lengthscale of a kernel component. Therefore, they are estimated in the same optimization procedure as described in the Appendix (Initialization Strategy & Parameter Optimization).

Furthermore, we use a constant mean function, for which the baseline value is set as the median value of the training set. The median latitude and longitude tend to correspond to (or are close to) a person's home location, as one's activity pattern tends to evolve around home. Thus, that becomes our model's baseline prediction in the absence of any information captured in the covariance function. Compared to other mean functions (e.g., linear, multivariate orthogonal polynomial), the constant mean is a safe choice for data-driven models—in intra-trip inference, a majority of the prediction task is related to points temporally close to the training set and hence the covariance function explains most of the variation in the predictions. Our experiments with other mean functions showed that they tend to produce irrational predictions once the distance between the testing and training sets exceeds a certain threshold.

3.4. Model training

In training our model, the goal is to optimally determine the set of hyperparameters $\boldsymbol{\vartheta}$ that minimize error with respect to the training data. The most straightforward method is to do so by minimizing the negative marginal log-likelihood (Rasmussen and Williams, 2006).

$$-\log\, p(\mathbf{Y}|\mathbf{X}, \boldsymbol{\vartheta}) = -\frac{1}{2}\mathrm{vec}(\mathbf{Y})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\mathrm{vec}(\mathbf{Y}) - \frac{1}{2}\log|\boldsymbol{\Sigma}| - n\log(2\pi), \tag{9}$$

where $\boldsymbol{\vartheta}$ is the set of model parameters and $\boldsymbol{\Sigma} = \mathbf{K}^f \bigotimes \mathbf{K}^x + \mathbf{D} \bigotimes \mathbf{I}$ and $|\boldsymbol{\Sigma}|$ is the determinant of the covariance matrix. Equation (9) provides a target function for kernel learning. The first component here estimates the model fit, while the second and third terms act as the regularization (Rasmussen and Ghahramani, 2000). Practically, the inverse term in the first component of this equation may pose a computational barrier when the size of the training data (and hence the kernel matrix) becomes too large. Traditional exact GP inference scales in $\mathcal{O}(n^3)$, while newer implementations take advantage of batched linear conjugate gradients to reduce this to $\mathcal{O}(n^2)$ (Gardner et al., 2018).

Various approximation methods have been proposed to overcome the computational limits of GPs on large datasets. Sparse Gaussian Processes (SGPs), for example, ease the computational burden by employing a subset of the training data ("inducing points") to reduce the dimensionality of the covariance matrix (Titsias, 2009). The key to the success of these methods is the careful selection of

[2] For details on why this is the case, we refer the reader to Section 4.2.4 of Rasmussen and Williams (2006).

inducing points, which must be informative enough to explain most of the variation in the training data. Popular SGP methods use gradient-based optimization algorithms to choose optimal inducing points—these approaches run into problems when considering a large inducing point set size and/or high dimensional input spaces (Snelson and Ghahramani, 2005).

For passively-generated mobile data, most of the variation is generated from users' taking trips that differ from each other in a number of dimensions. Specifically, destination and mode choice, as well as departure time, are the three primary dimensions in which individuals' travel patterns vary. A wide range of literature has proposed parametric models to estimate these variables (Abkowitz, 1981; Daisy et al., 2018; Kitamura, 1988). In our model, we relate the variance of these variables to changes in a high-dimensional time matrix through the covariance function. However, we also do not need each available data point while training the model—many observations are close to each other in space, particularly if the sampling rate for a given period is high.

Algorithm 1: Trajectory data compression

Input: *Raw trajectory data of one or multiple users; spatial radius parameter r*
Output: *Compressed trajectory data*
LenTraj ← *Count the total number of observations in the raw trajectory*
Sorting: *Sort by 'User ID' and 'Datetime'*
Initialization: *Create lists* $[\phi]$ *and* $[\lambda]$*, initializing them with* ϕ_i *and* λ_i*. Initialize* $i = 0$ *and* $j = 1$
for $(i <$ ***LenTraj)*** ***do*** *// Read each observation in the raw trajectory*
    for $(j <$ ***LenTraj*** $+ 1)$ ***do***
        $d_{ij} \leftarrow$ *Measure Haversine distance between* (ϕ_i, λ_i) *and* (ϕ_j, λ_j)
        if $d_{ij} > r$ ***do***
            break *// End current for loop*
        Add ϕ_j *and* λ_j *to lists* $[\phi]$ *and* $[\lambda]$
        $j \mathrel{+}= 1$
    $(\boldsymbol{\phi}, \boldsymbol{\lambda}) \leftarrow$ ***np.median*** $([\phi])$, ***np.median*** $([\lambda])$ *// Replace each point in* $[\phi]$ *and* $[\lambda]$ *with the median point of each list*
    $i \leftarrow i + j$;
    $j \leftarrow i + 1$ *// Update indices so that the compressed points are skipped in next iteration*
Return Output *// The compressed trajectory data*

We reduce the size of trajectory data through a batched compression algorithm that aims to generate an approximated trajectory largely retaining the shape of the original trajectory. This algorithm is a variation of the well-known Douglas-Peucker algorithm (Douglas and Peucker, 1973), though instead of specifying an error requirement, we specify a Haversine distance—if multiple points are within a certain distance of one another, we replace them using the median of those points. We chose this variation as it was faster to implement and used a more interpretable parameter based on distance. Algorithm 1 describes the full algorithm, which achieves the same outcome as a gradient-based optimization algorithm that carefully chooses training points which capture most of the variability. This reduces the computational burden of our model.

4. Implementation

We implement our algorithm and data analysis in Python. Specifically, we leverage GPyTorch, an efficient and modular implementation of GPs, as well as scikit-mobility, a data processing framework for GPS traces in the context of human mobility (Gardner et al., 2018; Pappalardo et al., 2022). The entire library of functions and classes we develop can be found at https://github.com/ekinugurel/GPSImpute. The "requirements.txt" file in the repository lists the required packages and their versions to run our program.

4.1. Data Preprocessing: Oscillation Correction, noise Filtering, and coordinate normalization

We begin by preprocessing raw GPS traces to remove noisy data points—specifically, we filter by maximum velocity, using 200 km/h as the upper limit. Because segment velocities are calculated "as the crow flies" in our analysis (and hence expected to be smaller than the true value), a more conservative limit was appropriate (we provide more thoughts on this choice in the Appendix - Max Speed Threshold Sensitivity Analysis). This erases many of the oscillations or physically infeasible jumps that may be observed in mobile data due to the urban canyon effect. We then remove noisy points from the raw data by excluding observations more than 300 m in precision (this is the location radius for which the data provider has 95% confidence). In practice, the potential of these noisy points to provide previously unobserved location information is shadowed by the problems they cause for model calibration, disrupting the continuity of smooth trajectories. Finally, prior to training the model, we normalize each user's training coordinates such that they have a mean of 0 and a variance of 1.

4.2. Gap Imputation: Input specification

As described under Section 3.2, we represent time in multiple ways to encode the complexities of human mobility patterns (e.g., history dependency, periodicity) into the multi-task GP model. We use a combination of continuous, categorical, and binary variables. The continuous variables include a monotonically-increasing Unix time, normalized to be 0 at users' first data point, as well as the number of seconds elapsed after midnight in a given day. Furthermore, we represent the days of the week using a one-of-k encoding as previously described, and they make up our categorical variables. Finally, we use binary variables to denote whether the given day is a

public holiday, whether it is a weekend, and whether it is included in the AM or PM peaks (defined as 7-10am and 3–6 pm, respectively). Table 1 summarizes the temporal dimensions used in our experiments.

With these in mind, we briefly remark on the previously-introduced kernel function (in Equation (8)). The RQ-ARD kernels are specified to fit every input dimension (i.e., produce a unique lengthscale for each input vector), while the PER kernels fit only a monotonically-increasing Unix time variable t_u, the only continuous and unbounded input in our input space. Also note that the PER kernels are multiplied by the RQ-ARD kernels—this can be loosely thought of as an AND operation, rather than an OR operation (which corresponds to addition). Therefore, the periodic kernels are only meant to supplement the hidden structures captured through the specification of additional temporal variables, not replace them.

5. Dataset

We employ privacy-protected, passively-generated mobile data from Spectus, a U.S.-based data solution provider specializing in geospatial analytics. The dataset contains time-stamped location traces of 2,000 anonymous, opted-in individuals in the Greater Seattle Area between December 2019 and July 2020. More specifically, the data includes timestamps, unique device identifiers, latitude and longitude coordinates, and a measure of data precision (i.e., a location radius for which we have 95 % confidence). The dataset does not include any demographic or socioeconomic information. Locational data is sent to Spectus servers in encrypted form, via an HTTPS protocol through three options: an Application Protocol Interface (API), through a publisher that has licensed Spectus' software development kit, or via direct server-to-server integration (Spectus, 2022a).

Spectus enhances users' privacy through two methods. First, they remove data from locations that do not meet privacy standards determined by their Sensitive Points of Interest (SPOI) policy. Though not an exhaustive list, the following is an example of data Spectus does not use for attribution: health-related, locations with vulnerable populations, sensitive lawful businesses, military-related, locations with first responders, correctional facilities, locations with firearms, churches/religious facilities, Native American reservations, sexual orientation-related, adult-oriented entertainment, and social demonstrations (Spectus, 2022b). Second, Spectus anonymizes data in around home locations using patent-pending technology. What this means is that traces near identified home locations are replaced by points at the centroid of the address' census block group (CBG), thereby not revealing one's actual residence. CBGs are the second smallest geographical unit for which the U.S. Census Bureau publishes sample data, and they contain a nationwide average of 51 blocks.

5.1. Defining Missingness: Temporal occupancy

Because Spectus' datasets are passively-generated, the gaps between any two adjacent observations are rarely equal in length. Thus, we need a convention to mathematically denote varying levels of missingness in mobile data. We discretize a user's total available data time $\mathcal{T}$ into P intervals of length τ, which we refer to as the "temporal resolution." The choice of τ is important—it decides the sparseness of a user's observed trajectory, in which each interval is assigned an indicator variable

$$\mathrm{I}_p = \begin{cases} 1 & \text{if } p \text{ has at least one observation} \\ 0 & \text{otherwise} \end{cases}$$

We thus define temporal occupancy (or the inverse of sparsity) as

$$q_\tau = \frac{1}{P}\sum_{p=1}^{P} \mathrm{I}_p \tag{10}$$

Note that $0 \leq q_\tau \leq 1$. A period with high q has fewer gaps in data and vice versa. For example, a period with $q_{30} = 0.9$ suggests that there exists at least one observation in 90 % of the 30-minute intervals.

5.2. Descriptive Analysis: Quantifying missingness

We provide some brief statistics on variables in the Spectus data, shown on Fig. 2. Most data points are highly precise—95 percent of all traces are within a 65-meter radius of their true location (top left). This is a significant change from a 2018 study looking at a

Table 1
Temporal dimensions used in the robustness experiments.

Variable	Notation	Type	Model Inputs	Code Notation
Unix Time (normalized)	t_u	Continuous	[0, 1,…, n]	`unix_min`
Second of the Day Sine	t_{ss}	Continuous	[0, …, 1]	`sam_sin`
Second of the Day Cosine	t_{sc}	Continuous	[0, …, 1]	`sam_cos`
Day of Week	t_d	Categorical	[0, 1, 2, 3, 4, 5, 6]	`dow`
Public Holiday	t_h	Binary	[0, 1]	`holiday`
Weekend	t_{we}	Binary	[0, 1]	`weekend`
AM Peak	t_{am}	Binary	[0, 1]	`am_peak`
PM Peak	t_{pm}	Binary	[0, 1]	`pm_peak`

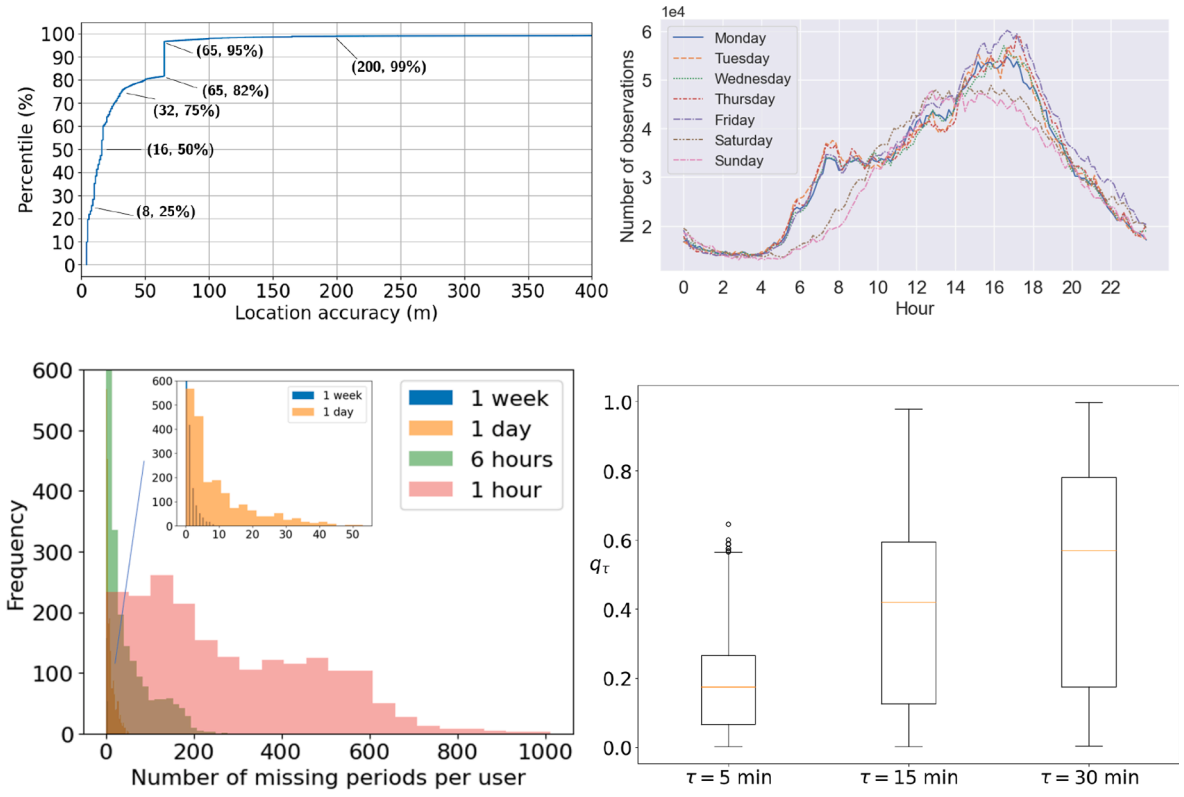

Fig. 2. Descriptive analysis of Spectus data for all 2,000 users. (Top Left) Cumulative distribution of location accuracy. (Top Right) Distribution of observations within each day of the week. (Bottom Left) Histogram of gaps categorized by size. (Bottom Right) Boxplots of temporal occupancy with three temporal resolutions.

similar dataset, in which about 7 percent of all observations had a precision less than 1,000 m (Ban et al., 2018). In terms of the distribution of observations throughout the day (top right), our findings are consistent with that of Ban et al. (2018) . We observe two peaks on weekdays, one in the morning (around 7am) and one in the evening (around 6 pm). For weekends, there is one mid-day peak. For every day, there is a positive trend on the number of observations as the hours progress.

As expected, a mobile dataset's level of "missingness" depends on the temporal resolution one chooses. The bottom right figure in Fig. 2 shows a boxplot of individual temporal occupancies of 2,000 users from the Spectus dataset for three different temporal resolutions. As the temporal interval τ increases, so does the average temporal occupancy. The bottom left figure in Fig. 2 shows that the gaps are not always short either—roughly 40 percent of all users recorded in Spectus data have at least one week that is continuously missing (i.e., there are no observations for a whole week). Meanwhile, 96 percent of all users have at least one day of missingness, and 99 percent have at least one six-hour period of missingness. Thus, we anticipate the need to prescribe a method to correct missingness in the general sense, whether the gaps are short or long.

6. Experiments

We do two sets of experiments to showcase our model. The first set (lengthscale analysis) explores the parameter space of multi-task GPs in the context of human mobility. Our goal here is to understand what model parameters work well for mobile data with certain characteristics (i.e., highway drives vs. urban walks), a benefit of which is developing a greater intuition on how to initialize GP model parameters in different contexts. The second set of experiments (robustness checks) benchmarks our model against alternative missing data imputation methods, including simple exponential smoothing, exponential smoothing (Huo et al., 2010), Holt-Winters (Cipra et al., 1995), ARIMA and SARIMAX (Kohn and Ansley, 2012), as well as a multi-task GP using the standard RBF-ARD kernel on each input dimension (Duvenaud, 2014).

For the first set of experiments (Section 6.1), we first preprocess raw GPS traces to obtain two datasets for 10 randomly chosen users (1920 trips total): (1) a set showing all trip-related data points for a user, and (2) a set containing various trip information, including average velocity, total distance, total duration, among other metrics. We fit GPs to trip-related data points to predict missing locations. Then, we relate the optimized model parameter values with respect to the compressed trip information, which allows us to infer the type of trip (i.e., highway drive or a walk). For the second set (Section 6.2), we randomly select a subset of 50 users with at least 10,000 observations and at least 3 months of data (from the first data point until the last). Within the selected 50, we only retain data from January and February 2020 to reduce the number of data points. We use this longitudinal data to train the model and make predictions

on varying gap lengths.

6.1. Lengthscale analysis

In this experiment, we determine well-fitting model parameters (i.e., ones that minimize the loss function) and relate them to a range of data characteristics, including average speed, total distance, and trip duration among other metrics. Identifying optimal kernel parameters is important for model accuracy. Fig. 3 illustrates how a suboptimal lengthscale can deteriorate model fit. The left subplot shows a trip with noisy training data and a lengthscale value that is too low. The prediction's mean reverts to the constant mean (likely near zero) only a few seconds after a training observation.

Furthermore, the optimization process deduces high levels of noise, producing uncertainty even on training inputs. Both processes result in low prediction accuracy as the prediction mean remains distant from the testing set. The right subplot, on the other hand, shows a model with low levels of noise and an appropriate lengthscale for the same trip. The prediction mean shifts smoothly between the different trip segments and the confidence interval is narrow near training points.

Using preprocessed Spectus data, we leverage k-means clustering to group together similar trips using trip characteristics (shown in the columns of Table 2). We filter a subset of mobile data based on three criteria, which help avoid anomalies like airplane-based travel and out-of-state roadtrips: average velocity does not exceed 80 mph, total travel time does not exceed 6 hours, and the number of observations (in a trip) is greater than 4. Using the elbow method, we identify three clusters from the filtered data as sufficient to explain most of the variation in trip-level data. Table 2 summarizes the average mobility metrics associated with the trips in the different clusters. The heading change rate shows the ratio of consecutive data points where a user changes direction with an angle exceeding a threshold (we use 0.33rad). The velocity change rate shows the ratio of consecutive data points where the user exceeds a speed variation threshold (we use 26%). Finally, the stop rate represents the ratio of data points with an inferred velocity lower than a threshold (we use 0.89m/s). These threshold values are chosen in accordance with findings from Zheng et al. (2008)[3] and are meant to discriminate between moving and being stationary.

We only used local data (i.e., observations from the same trip) in fitting GP models for these trips and thus our only input is t_u (normalized Unix time). Accordingly, we use the RBF kernel, a simpler form of the earlier-defined RQ kernel:

$$k_{\text{RBF}}(x, x') = \Omega^2 \exp\left(-\frac{(x - x')^2}{2\ell^2} \right) \tag{11}$$

Fig. 4a reveals that slower and shorter trips exhibit the lowest average RMSE, indicating higher prediction accuracy, whereas fast, distant trips present the highest RMSE. Correspondingly, Fig. 4b's boxplot of optimized lengthscale parameters for each trip cluster indicates that faster trips, likely associated with highway drives, necessitate larger lengthscales, reflecting less frequent changes in mobility patterns. In contrast, slower to medium-speed trips generally require shorter lengthscales, suggestive of more variable and less predictable movements. Notably, medium-speed trips display the widest range of lengthscales, underscoring a significant variability in temporal correlation. This can be ascribed to the heterogeneity of those trips, which often involve a combination of transportation modes, each with distinct velocities and movement patterns. Such diversity introduces numerous non-smooth transitions, or 'kinks,' into the data, which the Gaussian Process (GP) must adeptly capture.

The variability in lengthscales observed for slower and shorter trips may be attributed to the wide spectrum of moving behaviors that may involve a range of transportation modes used from walking to bicycling and driving. On one end, they may reflect walking behavior that may be taken with a clear destination in mind or just wandering. On the other end, they likely reflect driving on urban streets, frequently encountering stops due to congestion or traffic signals. This dichotomy is corroborated by the high heading change and stop rates as detailed in Table 2. Despite the complexities introduced by such diverse activities, the slower to medium-speed trips achieve lower median RMSE values, suggesting that the GP model is effectively modeling these trip types. This indicates that the GP framework is robust enough to handle the stochasticity inherent in a wide range of trips, accurately reflecting the nuanced shifts in mobility without a significant loss in predictive performance.

6.2. Robustness Checks: Gap imputation

The goal of this experiment is to assess the performance of our model against other imputation methods in a variety of missingness conditions. In doing so, we conduct a robustness analysis on the gap length. Our benchmark methods include simple exponential smoothing (SES), exponential smoothing (ES), the Holt-Winters (Holt) method, ARIMA, SARIMAX, and an RBF-ARD kernel. The first three (ES, SES, and Holt) work similarly—they weigh the average of past observations in producing forecasts, giving exponentially lower weights as the time gap increases. SES is most suitable for forecasting data with no clear trend or seasonal pattern, while ES allows for forecasting with a trend. The Holt-Winters method uses three types of exponential smoothing to model the level (the typical value), the trend (the slope), and the seasonality (repeating patterns) of data. ARIMA and SARIMAX are both linear regression models fit for forecasting univariate time series data. While ARIMA can handle data with a trend, its extension SARIMAX can also handle exogenous variables and seasonal components. Each of these methods is regarded as a reliable, well-studied alternative commonly

[3] See "Parameter Selection" on Zheng et al. (2008).

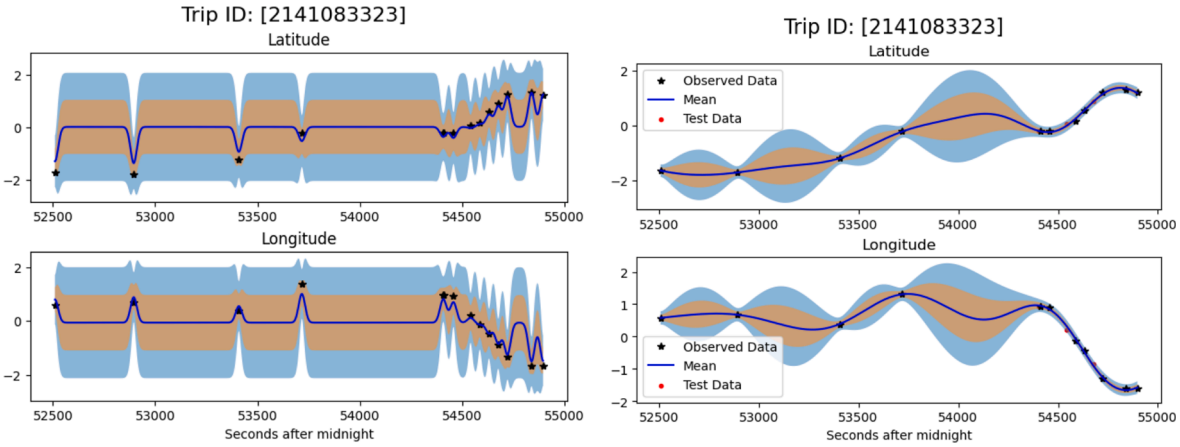

Fig. 3. Examples of trips fit with an underestimated lengthscale (left) and appropriate lengthscale (right).

Table 2
Summary of trip clusters.

Cluster	Avg. Vel. [m/s]	Distance [m]	Duration [s]	Heading Change Rate	Velocity Change Rate	Observations	Stop Rate
Slower, shorter trips	9.29	8,088	1,062	0.0019	0.0024	22.79	0.0007
Medium speed/ distance trips	13.94	29,693	2,362	0.0007	0.0008	49.86	0.0002
Fast, distant trips	17.86	59,299	3,449	0.0005	0.0006	141.8	0.0001

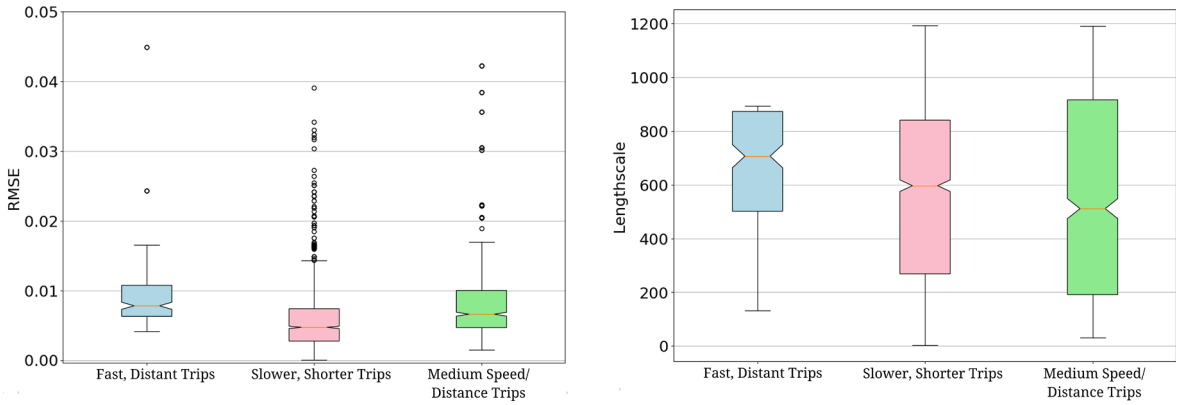

Fig. 4. (a) Boxplot of total RMSE for trips in different mobility metric clusters (b) Boxplot of optimized lengthscales for each trip cluster.

used to impute missing values in time series, and therefore make a useful comparison. The inclusion of the RBF-ARD kernel is to demonstrate that an off-the-shelf kernel implementation does not perform as well as our composite formulation. Table A1 in the Appendix describes and quantifies the parameters of each method.

Our results demonstrate that our model outperforms existing methods in all scenarios with varying levels of missingness. Specifically, we simulate gaps in the selected 50 users by reserving a subset of their data for testing, and we choose this subset such that the temporal occupancy of their training data meets a lower target temporal occupancy (i.e., a decimal between 0 and the current temporal occupancy) according to the temporal resolution being tested. We test six different temporal resolutions (τ): one week, one day, six hours, one hour, thirty minutes, and fifteen minutes. Algorithm 2 describes the full process to remove points from users' data.[4]

[4] We also provide a step-by-step breakdown of this algorithm in the Appendix.

Algorithm 2: gap simulation in trajectory data

Input: *Trajectory data; Bin length* τ
Output: *Trajectory data with gaps; New* q_τ
bins ← *Create an array of integers spanning from* $t_{u,1}$ *to* $t_{u,N}$ *with step size* τ
bins_dict ← *Use dictionary comprehension to map each data point to the relevant bin*
non_empty_bins_dict ← *Make a similar dictionary with only the non-empty bins*
target_ocp ← **np.random.uniform** $(0, q_{curr})$ // *Take decimal floor of randomly chosen value between 0 and the current temporal occupancy as the target occupancy*
while $(q_\tau >$ ***target_ocp)*** ***do*** // *Until temporal occupancy falls below the target level*
    np.random.choice(non_empty_bins_dict.keys()) ← *Randomly choose a bin to remove and remove all values in this bin from original data*
    bins_dict ← *Update the original dictionary*
    non_empty_bins_dict ← *Update the non-empty bins dictionary*
new_ocp ← *Calculate the new temporal occupancy with the gapped data*
Return Output // *The gapped trajectory data as well as the new* q_τ

In the context of plotting and analyzing the results of various models with differing levels of data sparsity, we utilized Dynamic Time Warping (DTW) as an ancillary metric. DTW, a method established by Müller (2007), is suitable for measuring the similarity between two temporal sequences, accommodating variations in speed and timing. DTW values give us an indication of the dissimilarity between sequences; a lower DTW value indicates a closer match, whereas a higher value suggests a greater disparity between the compared sequences.

From the analysis depicted in Fig. 5, we discern an inverse-linear relationship between temporal occupancy and DTW values, indicating that as temporal occupancy diminishes, DTW values — reflecting the disparity between testing set outcomes and model predictions — correspondingly escalate. This trend illustrates a progressive, rather than precipitous, decline in the model's predictive accuracy with sparser datasets. Notably, this inverse-linear relationship appears less distinct at smaller gap lengths, as illustrated when comparing Fig. 5a for $\tau = 1$ week and Fig. 5d for $\tau = 5$ minutes, suggesting a more stable model performance within these intervals. Despite the non-linear trends at the extremities of the temporal occupancies, potentially stemming from limited sample sizes, we do not observe a specific threshold beyond which the model's performance deteriorates drastically. Instead, the model's accuracy gradually tapers off with reduced temporal occupancy. This analysis emphasizes the model's consistent performance across different scenarios while indicating that ensuring a minimum level of temporal occupancy—adjusted according to the specific temporal resolution of the data—is crucial for achieving the best predictive accuracy from the model.

For the benchmark algorithms, we used only the monotonically increasing t_u as input since they do not have a straightforward way to deal with categorical or binary variables. For each individual, we trained an optimal benchmark model by maximum likelihood estimation (MLE) whenever applicable. In cases when we could not use MLE (i.e., determining the order parameters for ARIMA and SARIMAX), we used the grid search method to identify the optimal parameters (see Table A2 in Appendix).

We analyzed the accuracy of derived mobility metrics after the imputation of gapped training data compared to before simulating any gaps. Table 3 presents the median error results in each bin length. The boxplots of these error results can be found in the Appendix (Figs. A1-A7). The same 50 users' data was evaluated across the different gap lengths, allowing for an apples-to-apples comparison. In all tests, our model outperformed the other models: The average error was always the lowest, and the 5th and 95th percentile bounds tended to be closer. The metrics we compared across the different methods included the number of distinct locations, the radius of gyration, the straight-line distance traveled, real entropy, random entropy, and uncorrelated entropy. The number of distinct locations simply counted how many unique combinations of coordinate pairs existed in the imputed and original datasets. The straight-line travel distance traveled by an individual is computed as the sum of the distances traveled (Williams et al., 2015) and can be computed as

$$d_{SL} = \sum_{j=2}^{m} dist(r_{j-1}, r_j) \tag{12}$$

The radius of gyration indicates the characteristic travel distance of a user during a period (Song et al., 2010b) and is given by the equation

$$r_g = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(r_i - r_{cm})^2}, \tag{13}$$

where r_i represents the $i = 1, \cdots, n$ locations recorded for the user and $r_{cm} = \frac{1}{n}\sum_{i=1}^{n} r_i$ is the center of mass of the period's trajectory. Finally, the concept of entropy is used to assess a user's predictability. The real entropy of is given by the equation (Song et al., 2010b)

$$E_{real} = -\sum_{T'} P(T')\log_2[P(T')] \tag{14}$$

where $P(T')$ is the probability of finding a particular time-ordered subsequence T' in the trajectory T. Therefore, the real entropy depends not only on the frequency of visitation, but also the order in which the locations were visited, and the time spent at each location, thus capturing the entire spatiotemporal order present in a user's mobility. The random entropy captures a user's degree of

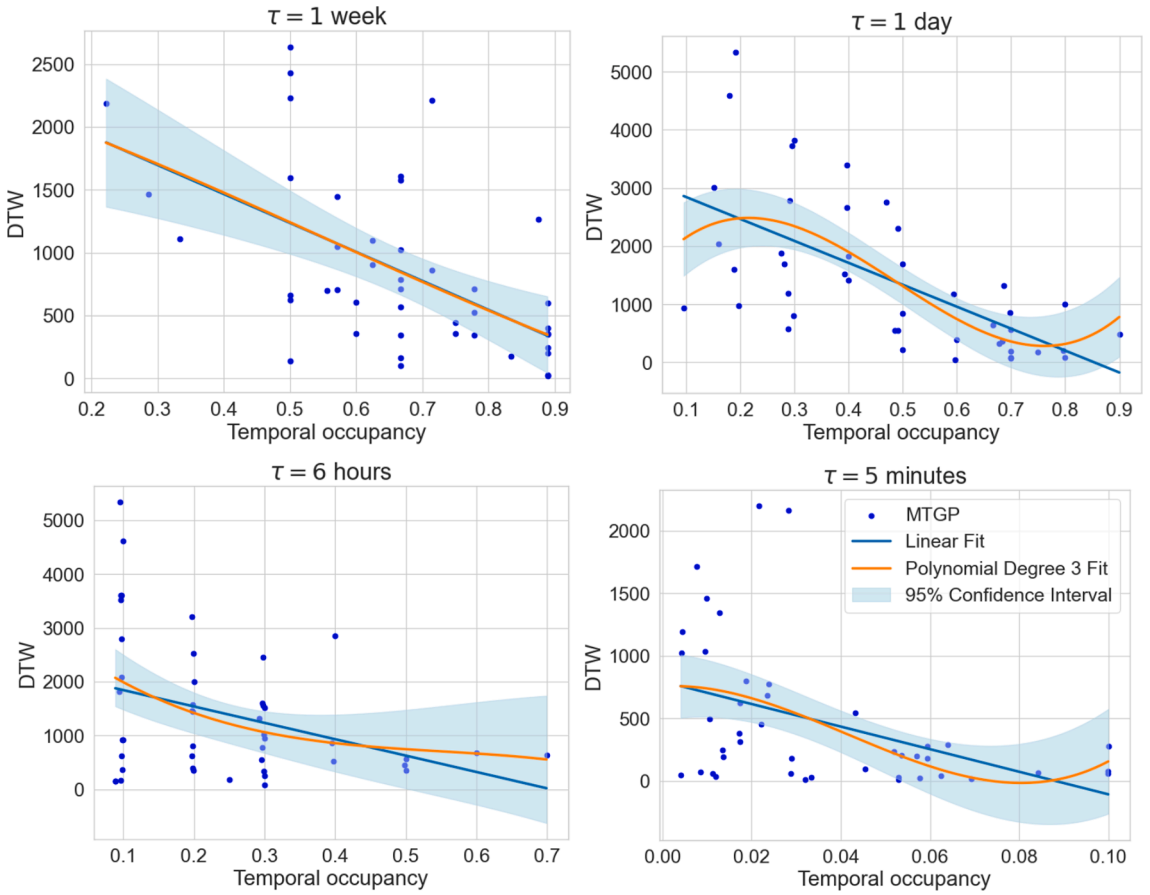

Fig. 5. Scatterplots of DTW and temporal occupancy for various gap lengths. (a) $\tau = 1$ week, (b) $\tau = 1$ day, (c) $\tau = 6$ hours, (d) $\tau = 5$ minutes.

predictability if each of their distinct locations were visited with equal probability. The (temporal) uncorrelated entropy characterizes the heterogeneity of a user's visitation patterns (Eagle and Pentland, 2009). The two are given by the equations

$$E_{rand} = \log_2[L] \tag{15}$$

$$E_{unc} = -\sum_{j=1}^{L} P(j)\log_2[P(j)] \tag{16}$$

where L is the number of distinct locations a user visits, $P(j)$ is the historical probability that a location was visited by the user.

We see that, on average, our model outperforms the rest of the benchmarks by a significant margin in estimating the number of locations visited, the radius of gyration, the straight-line travel distance, real entropy, random entropy, and uncorrelated entropy. Moreover, on these metrics, the distance between our model and the second-best model is significant. The standard RBF kernel multiplied across all input dimensions does not capture the variability of trips in sparse regions, performing worse than simpler smoothing algorithms even in low sparsity regimes. This highlights the importance of using appropriate kernel specifications while dealing with high-dimensional, mixed-type datasets.

Fig. 6 offers some perspective on the results presented here. In this example, a perfect predictor would have the red points align exactly with the blue points. The MTGP method (with the proposed kernel) better captures the variability of a user's location over time than Exponential Smoothing—hence why it is more accurate in metrics like straight-line travel distance (Equation (12)) and radius of gyration (Equation (13)). We also note that, in general, MTGP seems less accurate in capturing route choice of users, but more accurate in learning destination choice. This is what we attribute to the performance we see on metrics like real and uncorrelated entropy (Equations (14) and (16), respectively).

7. Discussion

The use of passively-generated mobile data for transportation applications and decision-making is an irreversible trend. And yet, such data exhibits significant and varying levels of missingness compared to actively solicited data such as household travel surveys

Table 3
Median error with respect to the testing sets.

Time Gap	Method	Number of Locations	Radius of Gyration	Straight-Line Travel Distance	Random Entropy	Real Entropy	Uncorrelated Entropy
1 week	***MTGP***	**26**	**−0.07**	**205.029**	**0.045**	**0.278**	**0.153**
	RBF	−801	−0.835	−888.441	−9.647	−9.323	−9.527
	SES	−801	−0.835	−888.441	−9.574	−9.323	−9.527
	Holt	−801	−0.835	−888.441	−9.574	−9.323	−9.527
	ES	−778	−0.621	−643.909	−4.989	−7.726	−4.955
	ARIMA	−801	−0.835	−888.441	−9.276	−9.323	−9.527
	SARIMAX	−801	−0.835	−888.441	−9.647	−9.323	−9.527
1 day	***MTGP***	**33.5**	**−0.245**	**236.805**	**0.036**	**0.227**	**0.117**
	RBF	−1050	−0.909	−1303.06	−10.038	−9.612	−9.806
	SES	−1050	−0.871	−1303.06	−9.309	−9.612	−9.806
	Holt	−1050	−0.846	−1303.06	−9.223	−9.61	−9.803
	ES	−1027	−0.718	−768.678	−4.878	−7.907	−5.225
	ARIMA	−1050	−0.834	−1303.06	−8.506	−9.609	−9.803
	SARIMAX	−1050	−0.909	−1303.06	−10.038	−9.612	−9.806
6 h	***MTGP***	**34**	**−0.187**	**−13.641**	**0.042**	**0.237**	**0.155**
	RBF	−956.5	−0.645	−1223.47	−9.809	−9.493	−9.751
	SES	−954	−0.645	−1177.23	−9.139	−9.493	−9.608
	Holt	−952	−0.645	−1171.79	−8.893	−9.493	−9.533
	ES	−929	−0.4	−768.56	−4.895	−7.869	−5.066
	ARIMA	−952	−0.645	−1178.65	−8.317	−9.493	−9.608
	SARIMAX	−956.5	−0.645	−1223.47	−9.901	−9.493	−9.751
1 h	***MTGP***	**38.5**	**−0.074**	**389.308**	**0.053**	**0.29**	**0.157**
	RBF	−902	−0.94	−1319.47	−9.818	−9.548	−9.761
	SES	−901.5	−0.761	−1262.95	−7.989	−9.546	−9.642
	Holt	−901.5	−0.761	−1262.95	−7.041	−9.543	−9.642
	ES	−878.5	−0.627	−711.119	−4.8	−7.816	−5.174
	ARIMA	−898.5	−0.761	−1262.76	−6.801	−9.545	−9.613
	SARIMAX	−830.5	−0.644	−1161.14	−6.435	−9.455	−9.449
30 min	***MTGP***	**21**	**−0.435**	**123.606**	**0.048**	**0.314**	**0.166**
	RBF	−624.5	−1.36	−1043.86	−9.052	−8.954	−9.161
	SES	−614	−1.175	−1025.83	−7.405	−8.954	−9.006
	Holt	−614	−1.167	−1025.83	−6.872	−8.953	−8.952
	ES	−591	−1.145	−707.361	−4.099	−7.07	−4.445
	ARIMA	−614	−1.214	−1027.68	−6.282	−8.949	−8.952
	SARIMAX	−624.5	−1.36	−1043.86	−9.277	−8.954	−9.161
15 min	***MTGP***	**22**	**−0.299**	**−7.116**	**0.048**	**0.323**	**0.161**
	RBF	−670	−2.15	−1112.56	−8.925	−8.871	−9.125
	SES	−660	−1.71	−1162.11	−7.34	−8.871	−8.994
	Holt	−660	−2.099	−1162.07	−6.435	−8.871	−8.849
	ES	−637	−2.056	−513.035	−3.96	−6.771	−4.497
	ARIMA	−659	−1.931	−1168.42	−6.048	−8.871	−8.851
	SARIMAX	−670	−2.15	−1199.07	−9.39	−8.871	−9.146
5 min	**MTGP**	**21**	−0.824	**47.301**	**0.056**	**0.301**	**0.156**
	RBF	−896	−1.396	−1441.06	−9.791	−9.302	−9.571
	SES	−896	−1.274	−1391.03	−6.757	−9.302	−9.571
	Holt	−893	−1.012	−1391.03	−6.555	−9.302	−9.394
	ES	−872	**−0.744**	−666.535	−4.313	−6.643	−4.984
	ARIMA	−896	−1.339	−1391.03	−6.754	−9.302	−9.495
	SARIMAX	−896	−1.396	−1441.06	−9.809	−9.302	−9.571

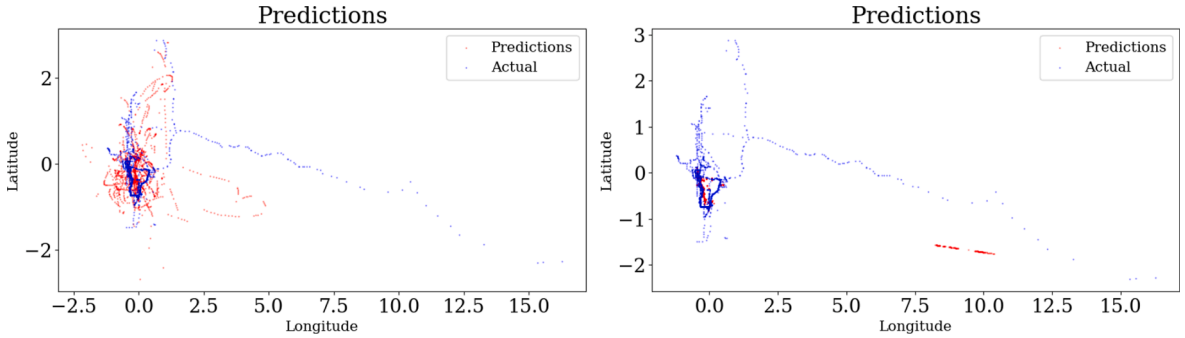

Fig. 6. Predictions (red) and ground truth test data (blue) for a user tested at $\tau = 6$ hours. MTGP with the proposed kernel (left) outperforms exponential smoothing (right). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

with GPS loggers or smartphone data collection components. Left untreated, the resulting mobility metrics are biased (McCool et al., 2022). To correct missingness in the mobile data, we have developed a novel framework using multi-task Gaussian processes. More specifically, our framework leveraged the correlations between users' coordinates (thus multi-task), and allowed individual-level differences in data characteristics resulting from fundamentally different trip generation processes (e.g., different modes of transportation used). We introduced and demonstrated the effectiveness of RQ and PER kernels in the context of mobile data for human mobility modeling, and highlighted the ARD extension of these kernels for fitting high-dimensional data (whether continuous or binary). Our experiments highlighted the importance of specifying a fitting composite kernel for our domain, as using the generic RBF kernel resulted in sub-optimal predictions due to not being able to capture multi-level periodicities. In the results, we showed that trips of different types (e.g., slower to faster trips) are associated with different, optimally estimated kernel parameters (i.e., the lengthscale) and provided guidelines on different parameter initializations for different kinds of trips. More importantly, we showed that our model outperformed six existing methods for correcting missingness in mobile data and assessed the temporal sensitivity of our model, confirming that our model achieves enhanced predictive performance when trained on a diverse temporal dataset.

While the current study is motivated by the need to directly address the significant sparsity issue exhibited in passively-generated mobile data (McCool et al., 2022), the implications of this work can be significant in several aspects. For one, it directly addresses the bias issue induced by data sparsity and the resulting, corrected trajectory data can be viewed as pseudo-ground truth from which mobility metrics can be derived. In another, it opens us the opportunity to create city-wide simulations of human mobility patterns using the generated mobile data directly. This is in contrast with the current process where parametric models are first estimated using the household travel survey data (which often represents less than 1% of a region's population[5]) and then those models are extrapolated to a synthetic population. As noted earlier, the latter method is not scalable as travel surveys must be frequently collected and models must be updated. An added disadvantage is that those models do a poor job of capturing the nonlinearities embedded in the data.

Transportation behaviors, whether they refer to individual travel behaviors as studied in this paper or community- or region-level traffic phenomena, are highly complex and non-linear. Thus, expecting a universal model to capture all is likely unrealistic. Instead, context-dependent modeling is much needed for the field of transportation. Context-dependent modeling requires the development of flexible modeling frameworks that can capture not only heterogeneity but also adapt to changing contexts. We demonstrate that GP-based methods are one suitable framework for this purpose in the context of modeling individual mobility patterns.

Our work underscores the critical role of kernel specification in capturing the nuances exhibited by individual mobility patterns. In our study, we used a domain-based approach, i.e., using our domain knowledge in the travel behavior literature to guide the selection and the composition of the kernels. Through our experiments, we show that standard, off-the-shelf kernels, such as the Radial Basis Function (RBF) kernel, frequently underperform in comparison to more complex kernels across a range of imputation scenarios. The process of specifying kernels is akin to a traditional, parametric modeling process where one would have to start with a hypothesized model structure. One may argue that even in data-driven modeling, one would still have to start with a modeling framework or architecture that define how input data are taken in and evolves over different layers, and whether there are feedback loops or not. Just as these models often require rigorous cross-validation to determine the most suitable hyperparameters, Gaussian Processes similarly benefit from a structured approach to kernel selection. Addressing this issue, notable advancements have been made in the development of automated methods for kernel determination. Duvenaud et al. (2013) have proposed a method for automating the construction of kernel expressions, thereby facilitating the identification of suitable kernel structures directly from the data. In a similar vein, Wilson and Adams (2013) introduced a framework that learns expressive covariance functions for GPs through spectral mixture kernels capable of automatically adapting to the structure of the data. These innovations are significant steps forward in simplifying the kernel selection process, thereby enhancing both the performance and the broader applicability of the model across various domains.

[5] In urban regions, the sampling rate is typically less than 0.1%.

Nevertheless, our study also recognizes certain limitations. One limitation is the challenge in selecting kernels that accurately emulate the data's inherent features, which is compounded by the issue of model interpretability. Although our model offers improved interpretability over deep learning (DL) models, which are often criticized for their 'black box' nature, it lacks the transparency of traditional statistical methods. The inherent complexity of Gaussian processes, particularly when employing sophisticated kernels such as Rational Quadratic (RQ) and Periodic (PER), introduces significant challenges in terms of interpretability. While model interpretability is not of paramount significance to our study (since the study objective is simply fixing missingness in raw mobile trajectories), potential applications of GPs in other transportation applications (such as those reviewed earlier in Section 2 for forecasting travel demand or traffic flows) would require higher levels of model interpretability. Thus, balancing model complexity with interpretability is an essential area for future research that involve GPs or other big data methods.

Another future direction of this work is to tackle computational complexity—the time it takes to train a model scales cubically with the size of the covariance matrix, which is a function of the size of the data. We outline a few ideas to explore this front. One is to accelerate the process of GP parameter optimization through CUDA (Compute Unified Device Architecture), a parallel computing platform and API created by NVIDIA that allows the software to use certain types of graphical processing units (GPUs) for general-purpose processing. Doing so would allow future models to add additional layers in a GP framework (i.e., deep GPs), predicting covariates like velocity and bearing before producing a location output. We can also pre-process the input data to drastically reduce the number of times we have to re-evaluate Equation (9) above. An example is provided by Lee et al. (2018).

There is also the need to incorporate the underlying built environment including multi-modal road networks into the model, such that bodies of water, buildings, or locations of different types can be recognized. Methods like map-matching may be employed to post-process model results such that the predictions can only be made within allowable regions. However, this approach would have a trade-off with computational complexity—though scalable map-matching algorithms have been proposed in recent years (Fiedler et al., 2019; Zeidan et al., 2020), convenient implementations are not yet widely available. Alternatively, one may also explore additional selection and identification of composite kernels (like Equation (8)) to potentially account for those contextual effects.

There are numerous applications of passively-generated mobile data in the existing literature, which are exclusively based on uncorrected trajectory data. It would thus be interesting to see whether the existing findings still stand if the corrected trajectory data is used. In a previous paper by one of the authors, we showed that not removing the oscillation phenomenon in the data leads to the non-negligible overestimation of the regularity of individuals' mobility (Wang and Chen, 2018). We similarly expect improvements in accuracy for downstream studies that use sparsity-corrected passively-generated mobile data. Future efforts could also be made to unify existing literature on applications of GPs to transportation problems, such as the works of Batista et al. (2022) and Gammelli et al. (2020), highlighting shared model properties, cautioning on common modeling mistakes, and establishing standards of benchmarking.

CRediT authorship contribution statement

Ekin Uğurel: Writing – review & editing, Writing – original draft, Visualization, Validation, Methodology, Investigation, Formal analysis. **Xiangyang Guan:** Writing – original draft, Investigation. **Yanchao Wang:** Writing – original draft, Investigation. **Shuai Huang:** Writing - review & editing, Methodology, Investigation, Conceptualization. **Qi Wang:** Investigation, Data curation, Writing – review & editing, Conceptualization. **Cynthia Chen:** Writing – review & editing, Writing – original draft, Supervision, Project administration, Methodology, Investigation, Funding acquisition, Data curation, Conceptualization.

Acknowledgments

The authors are grateful to the funding support from the National Science Foundation for the project titled as "A whole-community effort to understand biases and uncertainties in using emerging big data for mobility analysis" (award numbers 2114260 & 2114197). EK, XG, and CC are also supported by the Center for Teaching Old Models New Tricks (TOMNET), a University Transportation Center sponsored by the US Department of Transportation through Grant No. 69A3551747116.

Appendix

Initialization Strategy & Parameter Optimization

The optimization problem of a GP is non-convex. Therefore, kernel parameter initialization can help avoid bad local optima, leading to better model estimation and more accurate prediction results. We achieve this in two ways. First, if the gap is short enough, we take advantage of training points near the gap to infer trip characteristics. These include metrics like average velocity, total distance traveled, trip duration, and stop rate, among others. Our first set of experiments suggest that trips with differing characteristics tend to land at different optimal parameters. Therefore, we suggest initializing the lengthscale parameter of continuous temporal dimensions (Unix time in our case) according to the characteristics of the trip with missing data.

If using a monotonically-increasing temporal dimension as an input, one alternative we have identified is to initialize the lengthscale parameter with the average length of gap between observations in the training data. Ak et al. (2018) employed this strategy without any optimization. We however do further optimize the parameter beyond this initial heuristic using the Adaptive Moment Estimation algorithm (Kingma and Ba, 2017), a preferred choice among various machine learning frameworks due to its computational efficiency and ability to manage sparse gradients on noisy data.

We initialize binary variables (i.e., one-of-k encoded categorical variables) with a lengthscale of 1. This is largely done to avoid model misspecification during the optimization stage as lengthscale parameters are constrained to be nonnegative. In our implementation, a small amount of noise (called jitter) is added to the diagonal of the covariance matrix. This is done to ensure numerical stability when performing matrix operations like inversion and decomposition. Simultaneously, however, jitter can lead to parameter values dipping below 0 at the very first iteration of the optimization algorithm.[6]

Algorithm to determine training data for experiments in Section 6.2

To choose the training dataset in Section 6.2, we used Algorithm 2, which simulates gaps in trajectory data based on the current temporal occupancy with respect to the bin length being tested. The following is a step-by-step breakdown of that algorithm:

1. **Binning the Trajectory Data**: The algorithm starts by creating an array of integers called **bins** that span from $\mathbf{t_{(u,1)}}$ to $\mathbf{t_{(u,N)}}$ with a specified step size $\boldsymbol{\tau}$. This step is effectively dividing the entire temporal range of the trajectory data into bins of equal length.
2. **Mapping Data Points to Bins**: The **bins_dict** is then created using dictionary comprehension to map each data point to the relevant bin. This means that each data point in the trajectory is associated with a specific bin based on its timestamp.
3. **Selecting Non-Empty Bins**: Another dictionary, **non_empty_bins_dict**, is created to store only the non-empty bins. This step involves filtering out bins that have data points associated with them.
4. **Setting Target Occupancy**: The algorithm then sets a target occupancy level (**target_ocp**) by randomly selecting a value between 0 and the current temporal occupancy (**q_curr**). The selected value is used to determine the desired temporal occupancy.
5. **Gap Simulation Loop**: The algorithm enters a loop to simulate gaps in the trajectory data until the temporal occupancy falls below the target level (**target_ocp**). Within this loop:
   - o It randomly chooses a bin from the non-empty bins using **np.random.choice**.
   - o All data points associated with the chosen bin are removed from the original data, and the **bins_dict** and **non_empty_bins_dict** are updated accordingly.
6. **Calculating New Temporal Occupancy**: After the loop, the algorithm calculates the new temporal occupancy (**new_ocp**) with the gapped data.
7. **Return Output**: Finally, the algorithm returns the gapped trajectory data, as well as the new temporal occupancy ($\mathbf{q\_\tau}$).

Detailed imputation accuracy metrics for robustness experiments

Mobility metrics

Figs. A1–A7

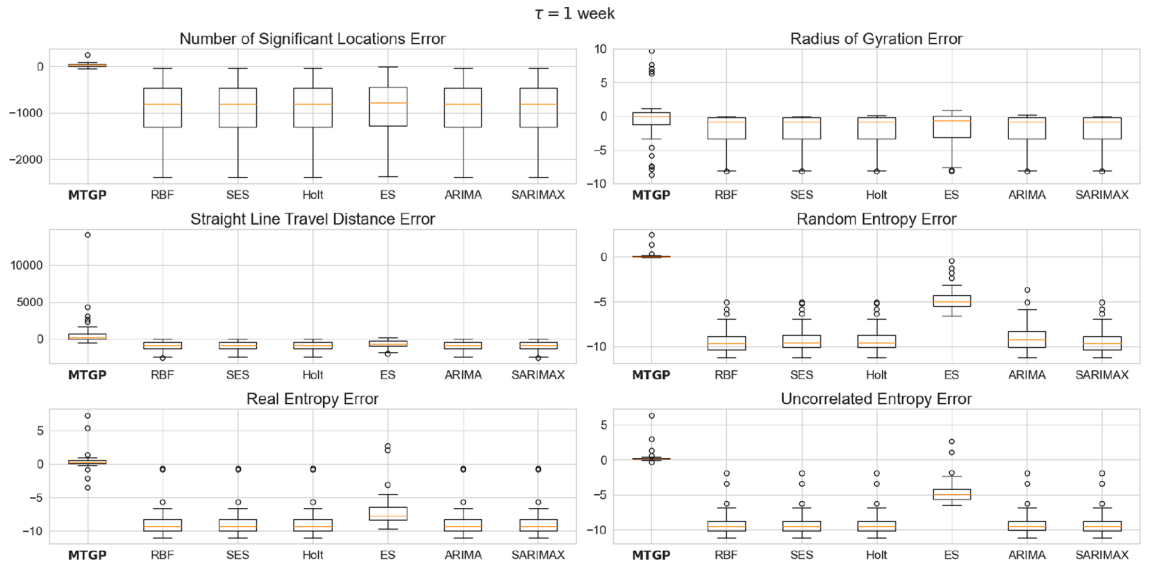

Fig. A1. Mobility error metrics for $\tau = 1$ week

[6] If the learning rate is small, the first iteration of the optimization algorithm may result in a value that is very close to 0, and the associated jitter may cause the parameter to flip signs.

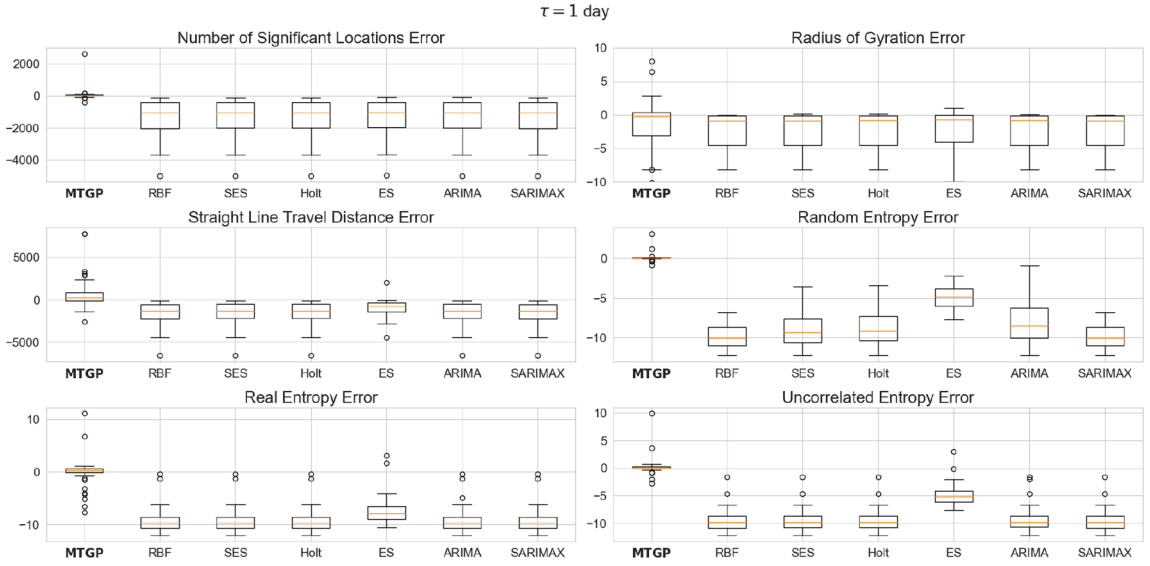

Fig. A2. Mobility error metrics for $\tau = 1$ day

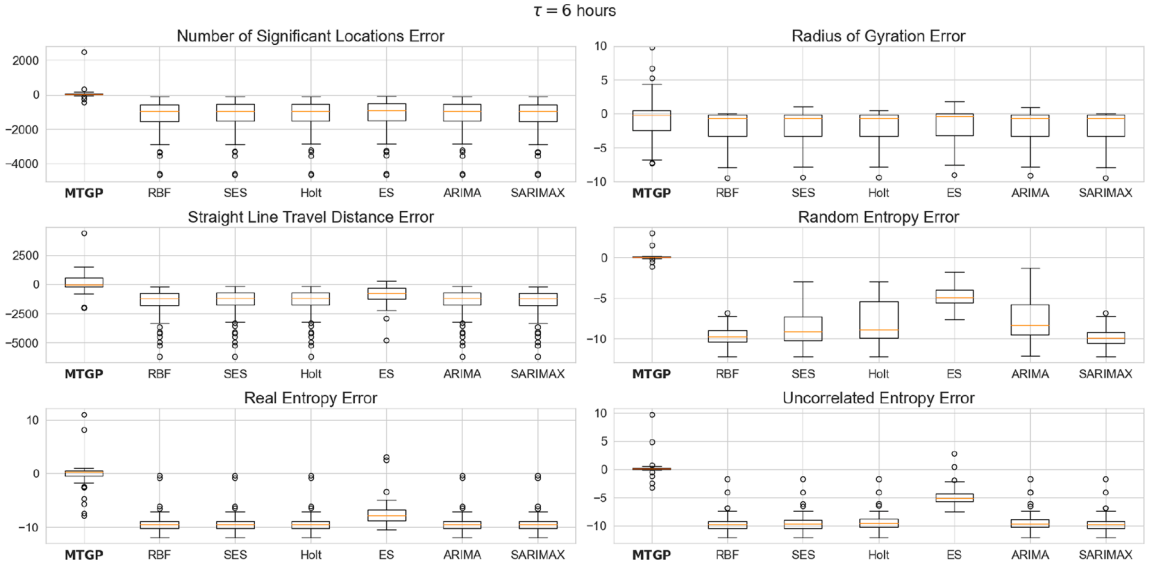

Fig. A3. Mobility error metrics for $\tau = 6$ hours

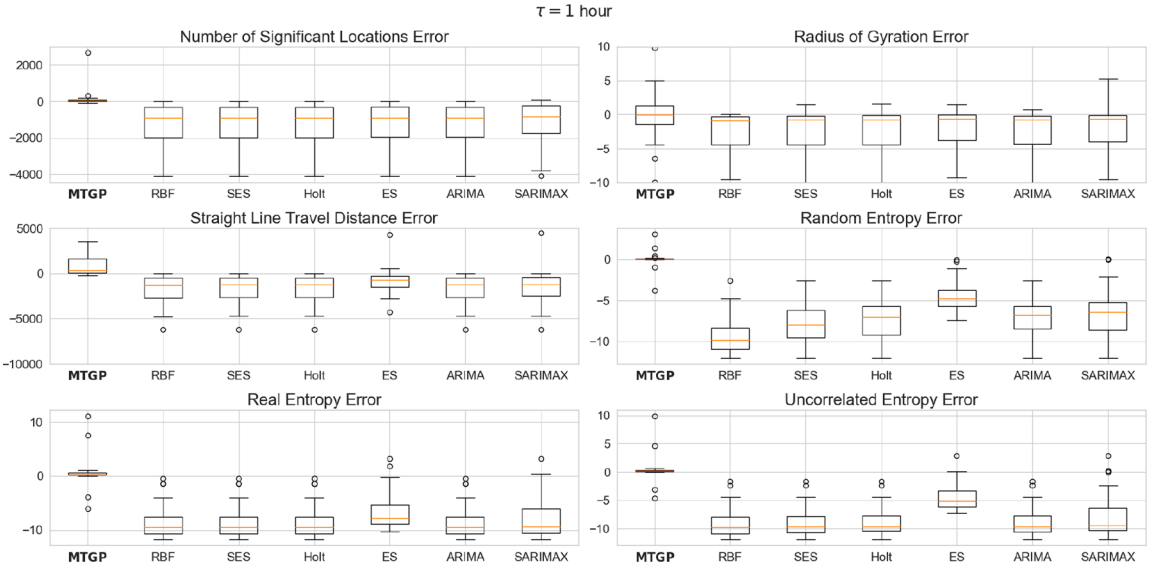

Fig. A4. Mobility error metrics for $\tau = 1$ hour

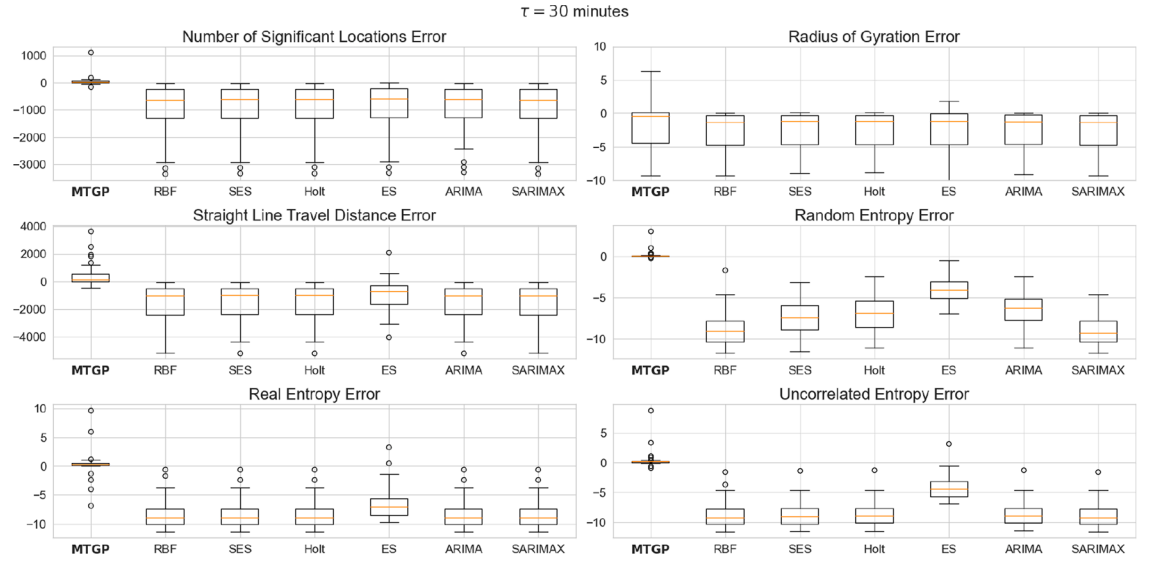

Fig. A5. Mobility error metrics for $\tau = 30$ minutes

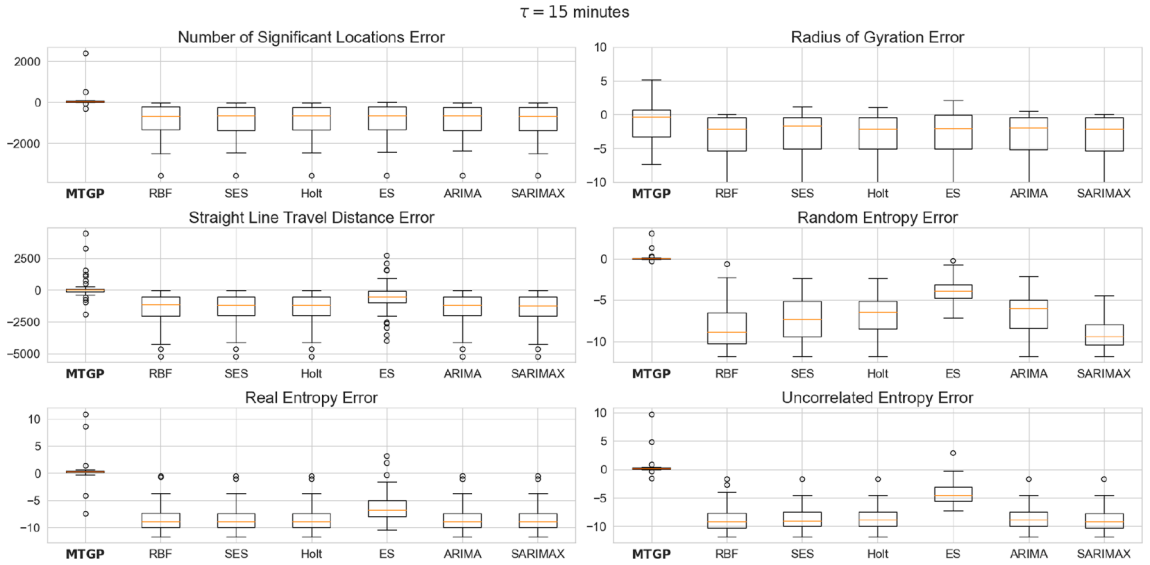

Fig. A6. Mobility error metrics for $\tau = 15$ minutes

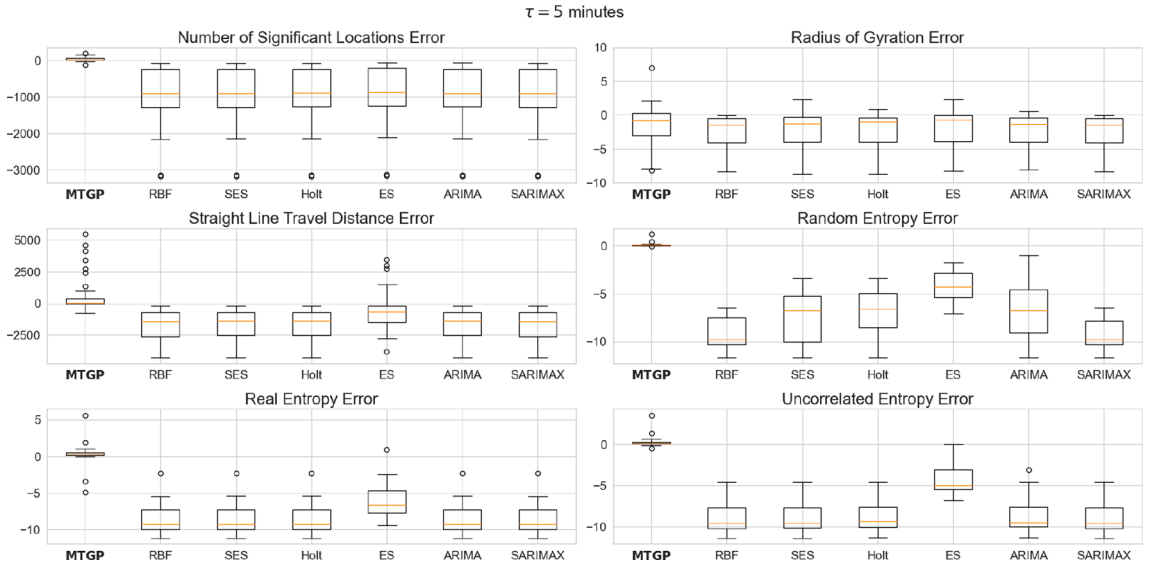

Fig. A7. Mobility error metrics for $\tau = 5$ minutes

Classical error metrics

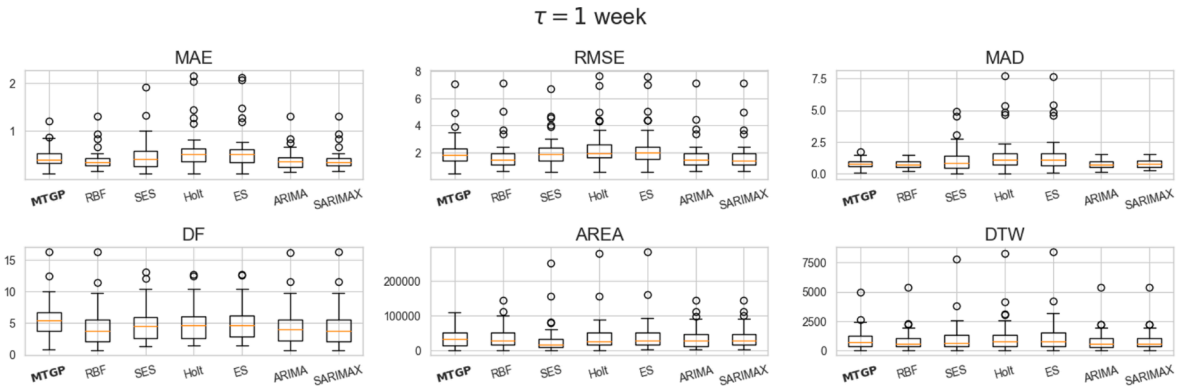

Fig. A8. Classical error metrics for $\tau = 1$ week

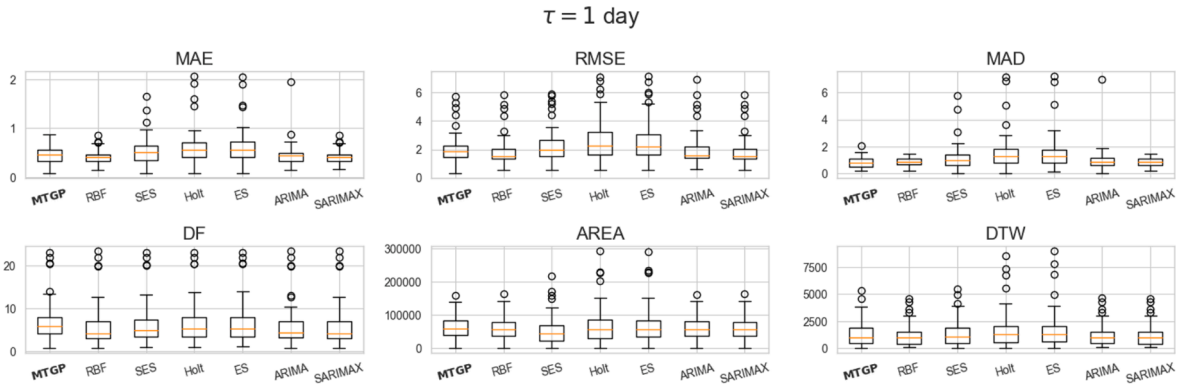

Fig. A9. Classical error metrics for $\tau = 1$ day

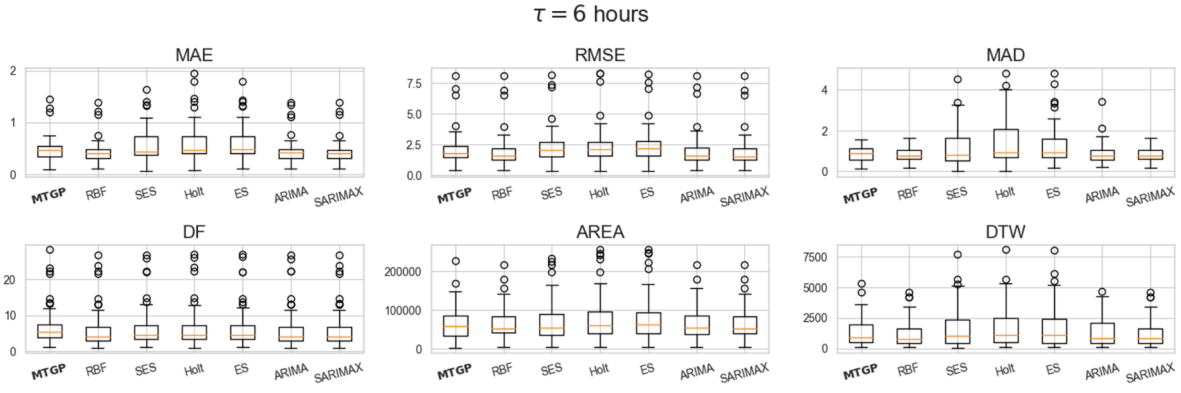

Fig. A10. Classical error metrics for $\tau = 6$ hours

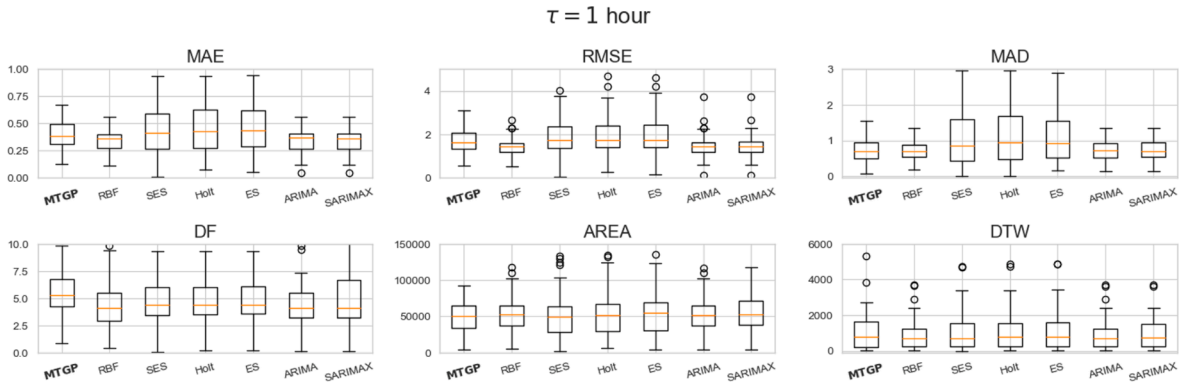

Fig. A11. Classical error metrics for $\tau = 1$ hour

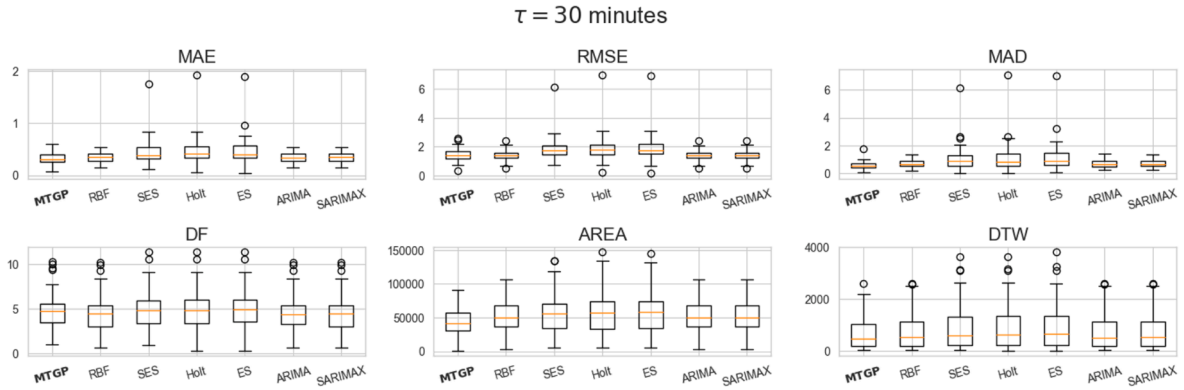

Fig. A12. Classical error metrics for $\tau = 30$ minutes

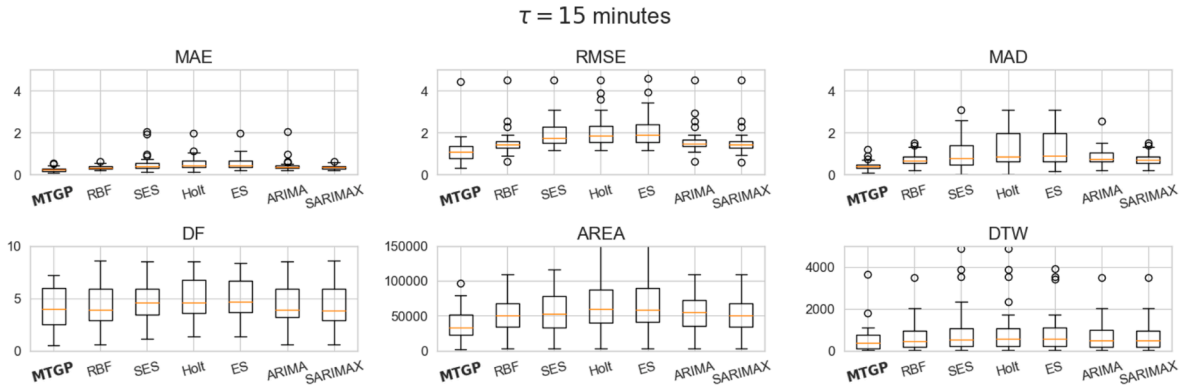

Fig. A13. Classical error metrics for $\tau = 15$ minutes

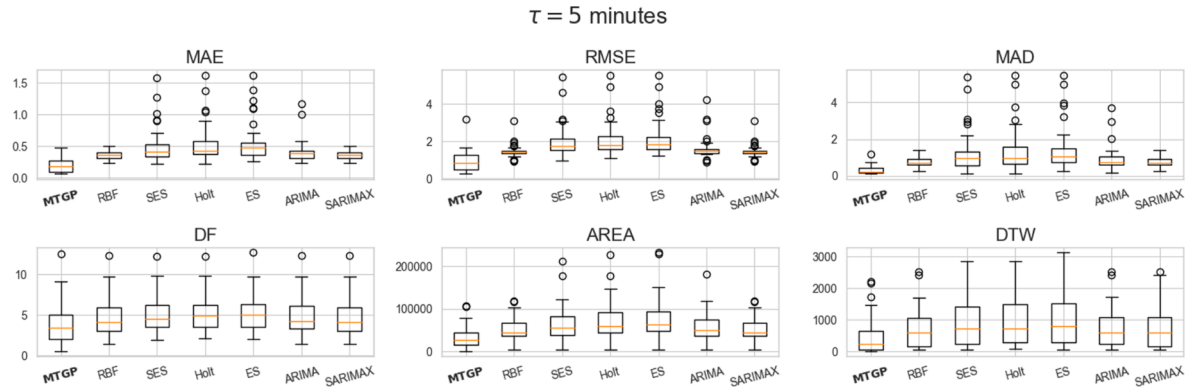

Fig. A14. Classical error metrics for $\tau = 5$ minutes

Benchmark methods and model parameters

Table A1
Parameters of benchmark methods.

Method	Parameter Count	Parameters	Notes
SES	1	α	Controls the weight given to the most recent observation when estimating the baseline of the time series
Holt-Winters	3	α	
		β_s	Determines the weight given to the recent change in the level of the data
		β_d	Determines the dampening of the trend.
ES	4	α	Same as above
		β	Determines the weight given to the recent change in the level of the data
		γ	Controls the weight assigned to the seasonal component
		m	Represents the length of the seasonal cycle (i.e., the number of seasonal periods to model)
ARIMA	5–10	p	Represents the number of lagged (past) observations used to model the current value of the time series
		d	Determines the number of differencing operations applied to the time series data to make it stationary
		q	Controls the number of lagged forecast errors (residuals) used to model the current value of the time series
SARIMAX	8–20	p, d, q	Same as above
		P	Represents the number of seasonal autoregressive terms in the model
		D	Refers to the number of seasonal differences applied to the time series data to make it stationary
		Q	Controls the number of seasonal moving average terms in the model
		s	Determines periodicity (number of periods in season)
MTGP-RBF	16	ℓ	Controls how far the model can extrapolate beyond the training data (using Euclidian distance). We assign a separate lengthscale for each input dimension (14 total).
		η	The weight of a kernel component (output variance)
		ε	Homoscedastic Gaussian noise
MTGP (ours)	21	ℓ, η, ε	Same as above ($2 \times \eta$)
		p	Determines the distance between repetitions of the function (2 total)
		α	Controls the relative weighting of large-scale and small-scale variations (2 total)

Table A2
Parameters used during robustness experiments.

Method	Parameter	Value
Preprocessing	Max Speed (km/h)	200
	Spatial Radius for Compression (km)	0.3
	Uncertainty Filter (m)	100
Adam	Learning Rate for Adam	0.3
	Max Number of Training Iterations	150
Periodic Kernels	Initial Period Length for $K_{PER,1}$	1440
	Initial Period Length for $K_{PER,2}$	10080
Rational Quadratic Kernels	Initial Lengthscale for t_u	$\frac{1}{2N}\sum_{i=1}^{N} t_{u,i} - t_{u,i-1}$
	Initial Lengthscale for $[t_{ss}, \cdots, t_{pm}]$	1
Search Range for ARIMA	Order of the AR,p	$[0, 3]$
	Order of the differencing,d	$[0, 2]$
	Order of the MA,q	$[0, 3]$

(*continued on next page*)

Table A2 (*continued*)

Method	Parameter	Value
Search Range for SARIMAX	Order of the AR,p	[0, 3]
	Order of the differencing,d	[0, 2]
	Order of the MA,q	[0, 3]
	Order of the seasonal AR,P	[0, 3]
	Order of the seasonal differencing,D	[0, 2]
	Order of the seasonal MA,Q	[0, 3]
	Order of the periodicity,s	24

Max speed threshold sensitivity analysis

To determine the optimal choice for a max speed threshold during the data preprocessing stage, we created a kernel density estimation (KDE) plot (Fig. A15) to visualize the number of observations eliminated by different speed limits. The KDE plot shows that the 200 km/h speed limit captures all of the points that the 300 km/h speed limit captures and some more. Because the speeds we calculate are based on "as the crow flies" distances, it is better to be more conservative in this limit—therefore, the additional points captured by the 200 km/h limit are most likely erroneous and have actual speeds much higher than 200 km/h (which is already high). This suggests that the 200 km/h speed limit is a better limit for eliminating outliers and reducing noise in the dataset. The 100 km/h speed limit, on the other hand, eliminates too many observations and reduces the data quality. Therefore, we chose the 200 km/h speed limit as the optimal threshold for our model (Fig. A15).

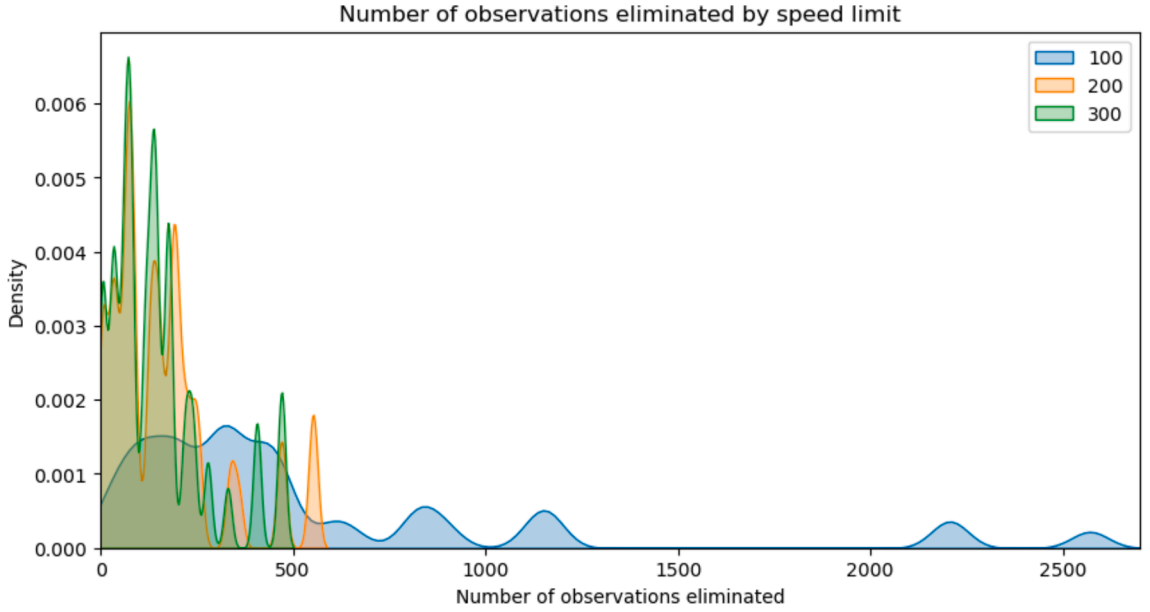

Fig. A15. The number of observations eliminated by varying speed limits across the 50 users' data analyzed in Section 6.2.

References

Abkowitz, M.D., 1981. An analysis of the commuter departure time decision. Transportation 10, 283–297. https://doi.org/10.1007/BF00148464.
Ak, Ç., Ergönül, Ö., Gönen, M., 2018. Structured Gaussian Processes with Twin Multiple Kernel Learning, in: Proceedings of The 10th Asian Conference on Machine Learning. Presented at the Asian Conference on Machine Learning, PMLR, pp. 65–80.
Ban, X. (Jeff), Chen, C., Wang, F., Wang, J., Zhang, Y., United States. Federal Highway Administration, 2018. Promises of Data from Emerging Technologies for Transportation Applications: Puget Sound Region Case Study (No. FHWA-HEP-19-026).
Barbosa, H., de Lima-Neto, F.B., Evsukoff, A., Menezes, R., 2015. The effect of recency to human mobility. EPJ Data Sci. 4, 21. https://doi.org/10.1140/epjds/s13688-015-0059-8.
Barbosa, H., Barthelemy, M., Ghoshal, G., James, C.R., Lenormand, M., Louail, T., Menezes, R., Ramasco, J.J., Simini, F., Tomasini, M., 2018. Human mobility: Models and applications. Phys. Rep. 734, 1–74. https://doi.org/10.1016/j.physrep.2018.01.001.
Batista, S.F.A., Cantelmo, G., Menéndez, M., Antoniou, C., 2022. A Gaussian sampling heuristic estimation model for developing synthetic trip sets. Comput.-Aided Civ. Infrastruct. Eng. 37, 93–109. https://doi.org/10.1111/mice.12697.
Ben-Moshe, B., Elkin, E., Levi, H., Weissman, A., 2011. Improving Accuracy of GNSS Devices in Urban Canyons 6.
Bonilla, E.V., Chai, K.M., Williams, C., 2008. Multi-task Gaussian Process Prediction 8.
Chen, C., Ma, J., Susilo, Y., Liu, Y., Wang, M., 2016. The promises of big data and small data for travel behavior (aka human mobility) analysis. Transp. Res. Part C Emerg. Technol. 68, 285–299. https://doi.org/10.1016/j.trc.2016.04.005.
Cipra, T., Trujillo, J., Robio, A., 1995. Holt-Winters Method with Missing Observations. Manag. Sci. 41, 174–178. https://doi.org/10.1287/mnsc.41.1.174.

Daisy, N.S., Millward, H., Liu, L., 2018. Trip chaining and tour mode choice of non-workers grouped by daily activity patterns. J. Transp. Geogr. 69, 150–162. https://doi.org/10.1016/j.jtrangeo.2018.04.016.
Douglas, D.H., Peucker, T.K., 1973. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. Cartogr. Int. J. Geogr. Inf. Geovisualization 10, 112–122. https://doi.org/10.3138/FM57-6770-U75U-7727.
Duvenaud, D.K., 2014. Automatic Model Construction with Gaussian Processes. University of Cambridge.
Duvenaud, D., Lloyd, J.R., Grosse, R., Tenenbaum, J.B., Ghahramani, Z., 2013. Structure Discovery in Nonparametric Regression through Compositional Kernel Search.
Eagle, N., Pentland, A.S., 2009. Eigenbehaviors: identifying structure in routine. Behav. Ecol. Sociobiol. 63, 1689. https://doi.org/10.1007/s00265-009-0830-6.
Fiedler, D., Čáp, M., Nykl, J., Žilecký, P., Schaefer, M., 2019. Map Matching Algorithm for Large-scale Datasets.
Frias-Martinez, V., Soguero, C., Frias-Martinez, E., 2012. Estimation of urban commuting patterns using cellphone network data, in: Proceedings of the ACM SIGKDD International Workshop on Urban Computing, UrbComp '12. Association for Computing Machinery, New York, NY, USA, pp. 9–16. https://doi.org/10.1145/2346496.2346499.
Gammelli, D., Peled, I., Rodrigues, F., Pacino, D., Kurtaran, H.A., Pereira, F.C., 2020. Estimating latent demand of shared mobility through censored Gaussian Processes. Transp. Res. Part C Emerg. Technol. 120, 102775 https://doi.org/10.1016/j.trc.2020.102775.
Gardner, J., Pleiss, G., Weinberger, K.Q., Bindel, D., Wilson, A.G., 2018. GPyTorch: Blackbox Matrix-Matrix Gaussian Process Inference with GPU Acceleration, in: Advances in Neural Information Processing Systems. Curran Associates, Inc.
Gong, X., Huang, Z., Wang, Y., Wu, L., Liu, Y., 2020. High-performance spatiotemporal trajectory matching across heterogeneous data sources. Future Gener. Comput. Syst. 105, 148–161. https://doi.org/10.1016/j.future.2019.11.027.
González, M.C., Hidalgo, C.A., Barabási, A.-L., 2008. Understanding individual human mobility patterns. Nature 453, 779–782. https://doi.org/10.1038/nature06958.
Hao, Q., Chen, L., Xu, F., Li, Y., 2020. Understanding the Urban Pandemic Spreading of COVID-19 with Real World Mobility Data. 26th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min. KDD 2020, 3485–3492.
Hills, T.T., Todd, P.M., Lazer, D., Redish, A.D., Couzin, I.D., 2015. Exploration versus exploitation in space, mind, and society. Trends Cogn. Sci. 19, 46–54. https://doi.org/10.1016/j.tics.2014.10.004.
Huo, J., Cox, C.D., Seaver, W.L., Robinson, R.B., Jiang, Y., 2010. Application of Two-Directional Time Series Models to Replace Missing Data. J. Environ. Eng. 136, 435–443. https://doi.org/10.1061/(ASCE)EE.1943-7870.0000171.
Idé, T., Kato, S., 2009. Travel-Time Prediction using Gaussian Process Regression: A Trajectory-Based Approach, in: Proceedings of the 2009 SIAM International Conference on Data Mining (SDM), Proceedings. Society for Industrial and Applied Mathematics, pp. 1185–1196. https://doi.org/10.1137/1.9781611972795.101.
Kim, D., Park, K., Park, Y., Ahn, J.-H., 2019. Willingness to provide personal information: Perspective of privacy calculus in IoT services. Comput. Hum. Behav. 92, 273–281. https://doi.org/10.1016/j.chb.2018.11.022.
Kingma, D.P., Ba, J., 2017. Adam: A Method for Stochastic Optimization. https://doi.org/10.48550/arXiv.1412.6980.
Kitamura, R., 1988. An evaluation of activity-based travel analysis. Transportation 15. https://doi.org/10.1007/BF00167973.
Kohn, R., Ansley, C.F., 2012. Estimation, Prediction, and Interpolation for ARIMA Models with Missing Data. J. Am. Stat. Assoc.
Kitamura, R., Kermanshah, M., 1983. Identifying Time and History Dependencies of Activity Choice. Transp. Res. Rec. 9.
Kitamura, R., Van Der Hoorn, T., 1987. Regularity and irreversibility of weekly travel behavior. Transportation 14, 227–251. https://doi.org/10.1007/BF00837531.
Le, T.V., Oentaryo, R., Liu, S., Lau, H.C., 2017. Local Gaussian Processes for Efficient Fine-Grained Traffic Speed Prediction. IEEE Trans. Big Data 3, 194–207. https://doi.org/10.1109/TBDATA.2016.2620488.
Lee, J., Bahri, Y., Novak, R., Schoenholz, S.S., Pennington, J., Sohl-Dickstein, J., 2018. Deep Neural Networks as Gaussian Processes. https://doi.org/10.48550/arXiv.1711.00165.
Liu, Z., Lyu, C., Huo, J., Wang, S., Chen, J., 2022. Gaussian Process Regression for Transportation System Estimation and Prediction Problems: The Deformation and a Hat Kernel. IEEE Trans. Intell. Transp. Syst. 23, 22331–22342. https://doi.org/10.1109/TITS.2022.3155527.
Liu, G., Onnela, J.-P., 2021. Bidirectional imputation of spatial GPS trajectories with missingness using sparse online Gaussian Process. J. Am. Med. Inform. Assoc. 28, 1777–1784. https://doi.org/10.1093/jamia/ocab069.
McCool, D., Lugtig, P., Schouten, B., 2022. Maximum interpolable gap length in missing smartphone-based GPS mobility data. Transportation. https://doi.org/10.1007/s11116-022-10328-2.
Merrill, N.H., Atkinson, S.F., Mulvaney, K.K., Mazzotta, M.J., Bousquin, J., 2020. Using data derived from cellular phone locations to estimate visitation to natural areas: An application to water recreation in New England, USA. PLoS One 15, e0231863.
Müller, M. (Ed.), 2007. Dynamic Time Warping, in: Information Retrieval for Music and Motion. Springer, Berlin, Heidelberg, pp. 69–84. https://doi.org/10.1007/978-3-540-74048-3_4.
Nasernejad, P., Sayed, T., Alsaleh, R., 2021. Modeling pedestrian behavior in pedestrian-vehicle near misses: A continuous Gaussian Process Inverse Reinforcement Learning (GP-IRL) approach. Accid. Anal. Prev. 161, 106355 https://doi.org/10.1016/j.aap.2021.106355.
Pappalardo, L., Simini, F., Barlacchi, G., Pellungrini, R., 2022. scikit-mobility : A Python Library for the Analysis, Generation, and Risk Assessment of Mobility Data. J. Stat. Softw. 103 https://doi.org/10.18637/jss.v103.i04.
Rasmussen, C., Ghahramani, Z., 2000. Occam' s Razor, in: Advances in Neural Information Processing Systems. MIT Press.
Rasmussen, C.E., Williams, C.K.I., 2006. Gaussian processes for machine learning, Adaptive computation and machine learning. MIT Press, Cambridge, Mass.
Ren, H., Ruan, S., Li, Y., Bao, J., Meng, C., Li, R., Zheng, Y., 2021. In: Virtual Event Singapore, pp. 1410–1419. https://doi.org/10.1145/3447548.3467238.
Rodrigues, F., Henrickson, K., Pereira, F.C., 2019. Multi-Output Gaussian Processes for Crowdsourced Traffic Data Imputation. IEEE Trans. Intell. Transp. Syst. 20, 594–603. https://doi.org/10.1109/TITS.2018.2817879.
Snelson, E., Ghahramani, Z., 2005. Sparse Gaussian Processes using Pseudo-inputs. Advances in Neural Information Processing Systems. MIT Press.
Song, C., Koren, T., Wang, P., Barabási, A.-L., 2010a. Modelling the scaling properties of human mobility. Nat. Phys. 6, 818–823. https://doi.org/10.1038/nphys1760.
Song, C., Qu, Z., Blumm, N., Barabási, A.-L., 2010b. Limits of Predictability in Human Mobility. Science 327, 1018–1021. https://doi.org/10.1126/science.1177170.
Spectus, 2022a. Privacy-First Approach and Policy. URL: https://spectus.ai/privacy/privacy-policy (accessed Feb 18 2024).
Spectus, 2022b. Sensitive points of interest policy. URL: https://spectus.ai/privacy/spoi-policy/ (accessed Feb 18 2024).
Steentoft, A., Lee, B.-S., Schläpfer, M., 2023. Quantifying the uncertainty of mobility flow predictions using Gaussian processes. Transportation. https://doi.org/10.1007/s11116-023-10406-z.
Storm, P.J., Mandjes, M., van Arem, B., 2022. Efficient evaluation of stochastic traffic flow models using Gaussian process approximation. Transp. Res. Part B Methodol. 164, 126–144. https://doi.org/10.1016/j.trb.2022.08.003.
Sulis, P., Manley, E., Zhong, C., Batty, M., 2018. Using mobility data as proxy for measuring urban vitality. J. Spat. Inf. Sci. 2018, 137–162. https://doi.org/10.5311/JOSIS.2018.16.384.
Sun, H., Yang, C., Deng, L., Zhou, F., Huang, F., Zheng, K., 2021. In: Virtual Event Queensland Australia, pp. 1734–1743. https://doi.org/10.1145/3459637.3482284.
Teixeira, D. do C., Almeida, J.M., Viana, A.C., 2021. On estimating the predictability of human mobility: the role of routine. EPJ Data Sci. 10, 49. https://doi.org/10.1140/epjds/s13688-021-00304-8.
Titsias, M., 2009. Variational Learning of Inducing Variables in Sparse Gaussian Processes, in. In: Proceedings of the Twelth International Conference on Artificial Intelligence and Statistics. Presented at the Artificial Intelligence and Statistics, pp. 567–574.
Wang, F., Chen, C., 2018. On data processing required to derive mobility patterns from passively-generated mobile phone data. Transp. Res. Part C Emerg. Technol. 87, 58–74. https://doi.org/10.1016/j.trc.2017.12.003.

Wang, Z., Zhang, S., Yu, J.J.Q., 2020. Reconstruction of Missing Trajectory Data: A Deep Learning Approach. In: In: 2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC). Presented at the 2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC), pp. 1–6. https://doi.org/10.1109/ITSC45102.2020.9294402.

Williams, N.E., Thomas, T.A., Dunbar, M., Eagle, N., Dobra, A., 2015. Measures of Human Mobility Using Mobile Phone Records Enhanced with GIS Data. PLoS One 10, e0133630.

Wilson, A.G., Adams, R.P., 2013. Gaussian Process Kernels for Pattern Discovery and Extrapolation.

Xie, Y., Zhao, K., Sun, Y., Chen, D., 2010. Gaussian Processes for Short-Term Traffic Volume Forecasting. Transp. Res. Rec. 2165, 69–78. https://doi.org/10.3141/2165-08.

Yuan, Y., Zhang, Z., Yang, X.T., Zhe, S., 2021. Macroscopic traffic flow modeling with physics regularized Gaussian process: A new insight into machine learning applications in transportation. Transp. Res. Part B Methodol. 146, 88–110. https://doi.org/10.1016/j.trb.2021.02.007.

Zeidan, A., Lagerspetz, E., Zhao, K., Nurmi, P., Tarkoma, S., Vo, H.T., 2020. GeoMatch: Efficient Large-scale Map Matching on Apache Spark. ACMIMS Trans. Data Sci. 1, 1–30. https://doi.org/10.1145/3402904.

Zheng, Y., Li, Q., Chen, Y., Xie, X., Ma, W.-Y., 2008. Understanding mobility based on GPS data, in. In: Proceedings of the 10th International Conference on Ubiquitous Computing. Presented at the UbiComp08: the 10th International Conference on Ubiquitous Computing, ACM, Seoul Korea, pp. 312–321. https://doi.org/10.1145/1409635.1409677.

Zheng, V.W., Zheng, Y., Xie, X., Yang, Q., 2012. Towards mobile intelligence: Learning from GPS history data for collaborative recommendation. Artif. Intell. 184–185, 17–37. https://doi.org/10.1016/j.artint.2012.02.002.

Zhou, H., Wang, H., Zhou, Y., Luo, X., Tang, Y., Xue, L., Wang, T., 2020. Demystifying Diehard Android Apps. In: IEEE/ACM International Conference on Automated Software Engineering (ASE). Presented at the 2020 35th IEEE/ACM International Conference on Automated Software Engineering (ASE), pp. 187–198.