Contents lists available at ScienceDirect

Transportation Research Part B

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

Learning to generate synthetic human mobility data: A physics-regularized Gaussian process approach based on multiple kernel learning

Ekin Uğurel [a], Shuai Huang [b], Cynthia Chen [a,b,*]

[a] THINK Lab, Civil & Environmental Engineering, University of Washington, Seattle, WA 98105, United States of America
[b] Industrial and Systems Engineering, University of Washington, Seattle, WA 98105, United States of America

ARTICLE INFO

Dataset link: https://github.com/ekinugurel/physics-regularized-MTGP

Keywords:
Passively-generated mobile data
Gaussian process
Multiple kernel learning
Multi-task learning
Travel behavior

ABSTRACT

Passively-generated mobile data has grown increasingly popular in the travel behavior (or human mobility) literature. A relatively untapped potential for passively-generated mobile data is synthetic population generation, which is the basis for any large-scale simulations for purposes ranging from state monitoring, policy evaluation, and digital twins. And yet, this significant potential may be hindered by the growing sparsity or rate of missingness in the data, which stems from heightened privacy concerns among both data vendors and consumers (users of service platforms generating individual mobile data). To both fulfill the great potential and to address sparsity in the data, there is a need to develop a flexible and scalable model that can capture individual heterogeneity and adapt to changes in mobility patterns. We propose a conditional-generative Gaussian process framework that learns kernel structures characterizing individual mobile data and can provably replicate observed patterns. Our approach integrates physical knowledge to regularize the framework such that the generated data obeys constraints imposed by the built and natural environments (such as those on velocity and bearing). To capture travel behavior heterogeneity at the individual level, we propose a data-driven multiple kernel learning approach to determine the optimal composite kernel for every user. Our experiments demonstrate that: (1) the impact of kernel choice on mobility metrics derived from synthetic data is non-negligible; (2) physics-regularization not only reduces model bias but also improves uncertainty estimates associated with the predicted locations; and (3) the proposed method is robust and generalizes well to varying individuals and modes of travel.

1. Introduction

The ubiquity of sensor-equipped mobile devices has enabled the generation of passively-generated large-scale time- and location-stamped data (hereafter called "mobile data"), including GPS traces, geo-tagged posts from social media platforms, and call detail records (Chen et al., 2016). These datasets are typically massive in size, encompassing millions of residents and travelers within a region, and longitudinal, spanning periods from days to months, and even years. These two attributes are fundamentally different from household travel surveys which are snapshots of a region's travel patterns with a small sample (typically less than 1% of a region's population) and have been traditionally used for travel behavior and demand forecasting studies. These big mobile datasets offer great promises in moving travel behavior or human mobility studies forward in various applications. Existing studies of the past

* Correspondence to: 201 More Hall Box 352700 Seattle, WA 98195-2700, United States of America.
E-mail address: qzchen@uw.edu (C. Chen).

https://doi.org/10.1016/j.trb.2024.103064
Received 15 January 2024; Received in revised form 11 August 2024; Accepted 26 August 2024
Available online 2 September 2024

15 years using these passively-generated mobile data have mostly focused on identifying models that can capture both regularities and variations of human mobility patterns (Gonzalez et al., 2008; Yuan and Raubal, 2012). The emergence of the COVID pandemic as well as AI in the last several years have stimulated additional studies that seek to identify cause and effect relationships (e.g., whether a change in the built environment leads to changes in mobility patterns; Liu et al. (2024)) and to understand changes in mobility patterns to support timely policy making (e.g., social distancing policy-making during pandemic requires rapid assessment of changes in people's mobility patterns; Morris et al. (2021)).

An emerging application of passively generated mobile data is the creation of synthetic populations for large-scale simulations of individual mobility patterns (Deng et al., 2021). These applications are wide-ranging and include state monitoring, regional or targeted policymaking, and long-term investment decisions. Using passively-generated mobile data, which are not only massive but also longitudinal, for this purpose has the potential to transform the current state of the art in the field that are primarily of two kinds. The first kind is the long-standing Activity-based Models (ABM) which are essentially a set of linked econometric models whose parameters are estimated using household travel survey data (Pendyala et al., 1997; Bhat and Koppelman, 1999; Timmermans and Arentze, 2011). Estimated econometric models reflect average trends and thus direct application of those models at the individual level result in large amounts of errors. Furthermore, ABM models are often estimated on small cross-sectional datasets and thus are static and cannot capture how behaviors evolve over time. The second kind are data-driven models that leverage population-level probabilistic distributions estimated from the big mobile data. These models capture individual heterogeneity through the expression of a distribution and fundamentally they cannot account for or adapt to nuances that are unique to an individual. These nuances may include factors unique to the individual (e.g., socio-demographics), the trip itself (e.g., the specific modes of transportation being used), and the underlying physical environment at the moment (e.g., road networks regulate average velocity and bearing). Fundamentally, neither of the two kinds of the existing studies touches upon uncovering the underlying data generation process at the individual level, which is a critical task for passively-generated mobile data to be used for synthetic population in large-scale simulations. This constitutes the central aim of our study.

The great potential of the passively-generated mobile data as stated above can be hampered by the fact that the consumer data privacy protection landscape has become increasingly user-centric, bestowing individuals with much greater flexibility in how their data is handled (i.e., easily allowing users to opt out of location-based data sharing agreements). This is due to a combination of factors both from the supply and demand sides: From the latter, consumers are now much more aware of the implications of unlimited data collection (i.e., hyper-targeted advertisements) and, as a result, wary of opting into agreements from which they do not receive any benefits (Kim et al., 2019). Simultaneously, companies like Apple have spearheaded efforts to promote user-centric privacy, implementing a range of new features in 2021's iOS 15 to help users better control and manage access to their data (Apple, 2021). Though the jury is still out on whether the goal of Apple's new initiatives were to maximize stakeholder returns or avoid anti-trust legislation (or both), the end result is the same: there has been a significant decrease in the amount of data generated, or a significant increase in sparsity. This emphasizes, from another perspective, the need to develop models that can uncover the underlying data generation process of passively-generated mobile data, as such models can generate synthetic data where consumer privacy will be less of an issue.

In this paper, we propose a framework that can be used to potentially generate synthetic individual mobile data that replicates real travel behavior. This approach can achieve two ends: It can augment existing datasets with sparsity, and alternatively, it can be substituted for real raw data while conducting analyses without compromising user privacy. In modeling travel behavior or mobility patterns, individual heterogeneity has been well-documented in numerous studies, as affected by many systematic factors such as demographic, socioeconomic, temporal, cultural, and built environment-related factors (Bayarma et al., 2007; Kitamura and Van Der Hoorn, 1987; McGuckin and Murakami, 1999; Nishii et al., 1988; Wallace et al., 2000), as well as spur of the moment factors (Lee and McNally, 2003, 2006). Thus, the challenge is to develop a flexible model framework that can adapt to individual uniqueness and easily scale up as more and new data come in. This model cannot take a parametric form that requires the pre-specification of a functional form. To this end, we propose a multi-task, multiple kernel Gaussian process framework that discovers optimal kernel specification for each user. Gaussian Processes (GPs) are a generalization of multivariate Gaussian distributions to infinitely many variables. A GP usually consists of two main components—a mean function that specifies the mean at any point in the output space, and a covariance (kernel) function that embeds a measure of similarity between any pair of observations in a multi-dimensional space. The most crucial component of a GP model is its kernel, as it determines the class of functions the GP relies upon (Rasmussen and Williams, 2006). To learn the form of the covariance function, we combine candidate kernels in a domain-inspired and data-driven manner to optimize model performance and use the Kronecker product to reduce the time it takes to invert a complex covariance matrix.

While powerful, unconstrained GPs can generate predictions that do not align with real-world conditions. Discrete observations of mobile data tend to satisfy certain physical bounds, which stem from the underlying street network and the broader built and natural environment. Specifically, we deal with two dimensions of physics: (1) Land-based human mobility has inherent limitations on the speed of travel, mostly related to the underlying street network as well as the time of day.[1] For example, a driver cannot travel 100 kilometers per hour in a residential street. Similarly, trips at night may have higher average velocities due to reduced traffic. (2) The direction of a user between consecutive data points is limited by the mode of travel and the built environment. If there is no paved road going east at a juncture, a person traveling by car would not be able to traverse that direction, just as a person

[1] This applies to water-based and air-based human mobility as well. However, because these trips are not captured in the dataset we use, we do not consider them in our methodology.

cannot walk into a lake. Thus, our approach integrates physical knowledge, such as segment velocities and bearings, into the GP inference process. Our conditional-generative framework, inspired by (but also differs from) the likes of Raissi et al. (2019), Lasserre et al. (2006) and Wang et al. (2022b), works by first sampling a set of locations as a function of time (conditional model) and then using a generative process to simulate physical variable observations at a certain time and location. However, our framework differs from existing methods in that it does not generate new training data before posterior estimation. Instead, it samples training data during estimation and generates synthetic physical inputs for model inference.

Our Contributions

- We propose a conditional-generative GP framework to generate synthetic individual mobile data that provably replicates observed travel patterns (Theorem 1);
- We formulate a flexible physics-regularization framework that permits the learning of composite kernels that can accommodate nonsmooth and nonstationary patterns in human mobility modeling;
- We develop a data-driven multiple kernel learning (MKL) approach to determine the best spatiotemporal kernel for each individual;
- Our experiments lead us to three findings: First, the impact of kernel choice on mobility metrics derived from synthetic data is non-negligible. Second, incorporating physics-based regularization not only diminishes model bias but also enhances the precision of uncertainty estimates in predicting locations. Third, the proposed method is robust and generalizes well to varying individuals and modes of travel.

2. Related work

Machine learning techniques like artificial neural networks (ANN) and kernel-based approaches have been particularly promising in modeling human mobility, with notable recent developments in map-matching (Jiang et al., 2023) and trajectory completion (Wang et al., 2020; Ugurel et al., 2024). The basic idea with many ANN-based models is to replace unknown functions with a neural network and then minimize a loss function with respect to the parameters of the network. Although ANN-based models have demonstrated empirical success in tackling complex problems, their performance can be inadequate for simpler problems if the ANN architecture is not tailored to the underlying domain (Wang et al., 2021b, 2022a). This goes hand-in-hand with the fact that the theory of ANNs has lagged behind its empirical success, with many ANN-based PDE solvers lacking rigorous theoretical guarantees, and convergence analyses often being limited to linear instances with constraints (Shin et al., 2020; Wang et al., 2022a). As Chen et al. (2021) points out, when compared to kernel methods, ANN methods suffer from limited theoretical foundations and present less favorable trade-offs between computational complexity and accuracy estimates. The generative nature of kernel methods can provide a more elucidated mathematical form of its model and can provide uncertainty quantification, probabilistic inference, and simulation.

Rasmussen and Williams (2006) wrote extensively on modeling and predicting complex interactions in space and time while accounting for spatial non-stationarity. The flexibility of Gaussian Processes in capturing nonlinearity and incorporating prior knowledge through kernel selection has demonstrated their efficacy in uncovering hidden patterns and predicting in dynamic geospatial contexts. The major limitation of using GPs is that the expert must choose a kernel to describe the underlying functional relationships in the data. An ill-fitting kernel can degrade GP model performance, leading to poor generalization and inadequate feature representation (Gönen and Alpaydın, 2011). In the human mobility domain, only a few previous works have explored preliminary methods on this kernel selection problem, while many are limited to choosing a kernel from an established off-the-shelf list of common kernels. Liu and Onnela (2021) proposed an additive kernel with a daily and a weekly temporal component as well as a distance-based spatial component to impute missing values in longitudinal mobile data, while Batista et al. (2022) focused primarily on the RBF and Matern kernels. The literature lacks a systematic framework established to account for heterogeneity in travel behavior at the individual level.

As a data-driven method, the performance of GP models can suffer when the training data does not sufficiently reflect the complexity of the system (Wang et al., 2022b). Left unconstrained, a GP's behavior may not satisfy expected system outcomes (Ugurel, 2023). Furthermore, GPs may produce predictions that are inconsistent with real-world conditions, defying the laws of physics or the constraints imposed by the built and natural environments. Physical models, built using physics knowledge expressed through differential equations, can characterize the underlying principles governing a system without needing large amounts of training data (Lapidus and Pinder, 1999). Thus, marrying data-driven methods with physical models not only provides insights into the system's mechanics but also enhances prediction accuracy by overcoming data limitations, which addresses the issues arising from unconstrained GPs.

Physics-regularization has been extremely popular in the context of ANNs (Raissi et al., 2019; Cuomo et al., 2022). Though several studies have proposed analogous approaches using GPs and other kernel methods (Wang et al., 2022b; Yang et al., 2019; Nevin et al., 2021; Chen et al., 2021), the methodology has not been developed as extensively. Specifically, the studies focusing on physics-regularized GPs have proposed methods for deriving physical knowledge for the estimation phase (i.e., to obtain the posterior). In this study, we use the human mobility domain to our advantage in overcoming this estimation problem, specifically by reducing the noise associated with direct estimation through proper data labeling (see Section 3.3). Conversely, the challenge of inference using this posterior, particularly when it involves physical variables that are dependent on both the inputs and the outputs, has not been extensively explored.

Within the transportation domain, Yuan et al. (2021) proposed a physics-regularized GP analogous to that of Wang et al. (2022b) to model macroscopic traffic flow, while Zhu et al. (2022) proposed a physics-regularized multi-output GP to model the numbers of

pickups, returns, and idle bikes of a large-scale bike-sharing system. Recently, Wu et al. (2024) used anisotropic Gaussian Processes to model congestion propagation in traffic flow data. These studies highlight the growing interest in applying physics-regularization in diverse contexts, yet there remains a gap in its application to individual-level mobility data where each individual has their unique spatial coverage and the data can extend for days, weeks or months. To our knowledge, this study is one of the first studies proposing a method to model the generation of GPS-generated human mobility traces while accounting for physical attributes associated with the built environment.

3. Methodology

3.1. Problem definition

Mathematically, the problem of uncovering the underlying data generation process for synthetic individual mobile data generation translates into two related research questions:

1. Given time, how do we infer (predict) spatial locations?
2. How do we infuse physics (i.e., velocity and bearing) into the inference problem from time to location, as stated above?

The input time can be represented by a set of variables, in addition to being simply a time index (e.g., the Unix time). Those additional time-related variables include periodic elements like the day of the week and weeks of the month. They capture varying rhythms that have long been identified in travel behavior literature (Hanson and Huff, 1988; Bayarma et al., 2007; Gonzalez et al., 2008): e.g., people usually go to their workplaces and return home every day and other trips like grocery shopping, going out with friends, or attending family gatherings happen less frequently. Additionally, there are behavioral patterns correlated with certain weeks of a month or months of a year (i.e., Christmas, Thanksgiving, etc.). We denote with $\mathbf{T}$ the matrix of temporal variables, where $t_{u,i}$ is the uth temporal dimension at the ith observation. The physics-related variables such as average velocity and bearings are to capture the constraints that the built environment has on one's movement pattern. We denote with $\mathbf{P}$ the matrix of physics-related variables, where v_i and β_i is the average velocity and bearing at ith observation respectively. The output variable location $\mathbf{Y}$ is represented with latitude λ and longitudeϕ[2]:

$$\mathbf{T} = \begin{bmatrix} t_{1,1} & \cdots & t_{d,1} \\ \vdots & \ddots & \vdots \\ t_{1,n} & \cdots & t_{d,n} \end{bmatrix} = \begin{bmatrix} \mathbf{t}_1 \\ \vdots \\ \mathbf{t}_n \end{bmatrix}, \quad \mathbf{P} = \begin{bmatrix} v_1 & \beta_1 \\ \vdots & \vdots \\ v_n & \beta_n \end{bmatrix}, \quad \mathbf{Y} = \begin{bmatrix} y_{\lambda,1} & y_{\phi,1} \\ \vdots & \vdots \\ y_{\lambda,n} & y_{\phi,n} \end{bmatrix}. \tag{1}$$

Human movements, as reflected in individual-level mobile phone trajectory data, are highly non-linear, context-dependent, and heterogeneous. Underlying the observed mobile data is the wide range of transportation modes used, each with different underlying physics (i.e., average velocity, acceleration behavior). When more than one mode of transportation is used, there is also a change of modes that need to be captured. Individual movement behaviors are also heterogeneous. While previous works have explored "universal" models on human movement patterns (Gonzalez et al., 2008; Song et al., 2010), later studies have revealed many other models that can capture human movement patterns and no single model will outperform others across contexts. These individual-level complexities suggest that making rigid assumptions about the shape or form of the underlying distribution of an individual's mobile data is unlikely a fruitful approach. A non-parametric, data-driven method holds greater promise, being more flexible and less constrained by predefined assumptions. Such an approach is capable of accommodating complex patterns and more adeptly capturing the inherent relationships present within the data. Thus, to answer the first research question stated above, we propose a multi-task Gaussian Process that learns individualized kernels to infer a set of locations $\mathbf{Y}$ from a set of temporal variables $\mathbf{T}$. As noted earlier, identifying kernels, or more specifically the correct structure for a combination of kernels, is critical as a fitting kernel can seamlessly capture complex dependencies in mobile data while a non-fitting one will be inadequate. In this paper, we address this multiple kernel learning problem in the context of mobile trajectory data (see Section 3.4).

To answer the second research problem stated above, we need information on the physics-related variables, or $\mathbf{P}$. And yet, $\mathbf{P}$ is not directly observed. More specifically, this issue of non-observability is different in the estimation problem where the kernels are learned from $\mathbf{T}$ to $\mathbf{Y}$ versus the inference problem where $\mathbf{Y}$ is inferred based on $\mathbf{T}$. In the estimation problem, since $\mathbf{T}$ and $\mathbf{Y}$ are both observed in the training data, $\mathbf{P}$ can be calculated. This calculation, of course, is unavoidably tampered with noise, especially when the time gap between two consecutive observations is large. We address this issue by a sample labeling procedure that reduces bias in the estimation of physical variables (see Section 3.3). In the inference problem, $\mathbf{Y}$ is not available, since it is to be inferred. Thus, we must first estimate $\mathbf{P}$ by considering its dependency on the yet-to-be-predicted output variable $\mathbf{Y}$, as well as on $\mathbf{T}$. We achieve this by formulating a model to predict the unobserved physical constraints $\mathbf{P}$ as a function of $\mathbf{T}$ and $\mathbf{Y}$. We then incorporate these generated constraints as observed inputs to aid in the inference of locations. We detail this strategy in Section 3.5.

The rest of Section 3 is organized as such: Section 3.2 translates the defined problem into one that can be addressed with Gaussian processes; Section 3.3 describes how to incorporate $\mathbf{P} \rightarrow \mathbf{Y}$ in the modeling framework; Section 3.4 discusses our greedy kernel learning approach to learn individual kernel structures; Section 3.5 ties together Sections 3.1 through 3.4 and describes our model inference strategy.

[2] For simplicity, we project λ and ϕ to a two-dimensional grid such that a user's location at time t is $\mathbf{y}_t = [y_\lambda, y_\phi]$. As travel distances between consecutive data points in mobile datasets tend to be short, this does not make a significant difference compared with extensions to spherical geometry.

3.2. Multi-task Gaussian process for mobile data

We first focus on modeling the relationship $\mathbf{T} \rightarrow \mathbf{Y}$. Consider the task of learning a function $f_j : \mathbb{R}^d \rightarrow \mathbb{R}$ from a training set $\mathcal{D} = (\mathbf{T}, \mathbf{Y})$ where j refers to either latitudes ϕ or longitudes λ. The basic form of our learning problem is

$$y_{ji} = f_j(\mathbf{t}_i) + \varepsilon_{ji}, \tag{2}$$

where f_j is a systematic function mapping inputs $\mathbf{t}_i$ to output y_{ji}, and $\varepsilon_{ji} \sim \mathcal{N}(0, \delta_j^2)$ are independent random variables for noise associated with the jth task. We place a GP prior on f_j such that $f_j \sim \mathcal{GP}(m(\cdot), k(\cdot,\cdot))$, where $m(\cdot) = \mathbb{E}[f_j(\cdot)]$ is the mean function (hereafter set to $\mathbf{0}$ assuming proper normalization), and $k(\cdot,\cdot)$ is the covariance (or kernel) function.

Assumption 1. The set $\mathbf{f}_j = [f_j(\mathbf{t}_1), \ldots, f_j(\mathbf{t}_n)]^\top$ follows a multivariate normal distribution such that $p(\mathbf{f}_j|\mathbf{T}) = \mathcal{N}(\mathbf{0}, \mathbf{K}_t)$, where $\mathbf{K}_t$ is the covariance matrix such that $[\mathbf{K}_t]_{i,g} = k_t(\mathbf{t}_i, \mathbf{t}_g)$, where subscript$_t$ is used to indicate temporal correlations and subscript$_g$ is an index like subscript$_i$.

In our approach, we relate multiple tasks (each multivariate normal distributed) by leveraging the correlations between them. Thus, we specify the covariance matrix for all n observations and two tasks as

$$\mathbf{K}_t = \mathbf{K}^t(\mathbf{T}, \mathbf{T}) \otimes \mathbf{K}^f(\mathbf{y}_\lambda, \mathbf{y}_\phi), \tag{3}$$

where $\mathbf{K}^t$ is the $n \times n$ covariance matrix of the training times using any valid PSD kernel, $\otimes$ is the Kronecker product, and $\mathbf{K}^f$ is a 2×2 PSD matrix containing the inter-task covariances, learned by jointly minimizing the negative marginal log-likelihood (Bonilla et al., 2007). The dimension of $\mathbf{K}_t$ for two tasks is then $2n \times 2n$. A key property of this model is that the joint distribution over the entire output space is not block-diagonal with respect to tasks. That is, observations of one task can affect the predictions on another task.

Assumption 2. The inferred location $f_j(\mathbf{t}_*)$ of a new input $\mathbf{t}_*$ conditioned on the training data is assumed to be distributed with the following form:

$$p(f_j(\mathbf{t}_*)|\mathbf{t}_*, \mathbf{T}, \mathbf{Y}, \delta_j^2) \sim N(\boldsymbol{\mu}_*, \boldsymbol{\sigma}_*^2), \tag{4}$$

where

$$\begin{aligned} \boldsymbol{\mu}_* &= (\mathbf{k}_j^f \otimes \mathbf{k}_*)(\mathbf{K}^f \otimes \mathbf{K}^t + \mathbf{D} \otimes \mathbf{I})^{-1}\mathrm{vec}(\mathbf{Y}) \\ \boldsymbol{\sigma}_*^2 &= (\mathbf{k}_j^f \otimes \mathbf{k}_{**}) - (\mathbf{k}_j^f \otimes \mathbf{k}_*)(\mathbf{K}^f \otimes \mathbf{K}^t + \mathbf{D} \otimes \mathbf{I})^{-1}(\mathbf{k}_j^f \otimes \mathbf{k}_*). \end{aligned} \tag{5}$$

Here, $\mathbf{k}_j^f$ selects the jth column of $\mathbf{K}^f$, $\mathbf{k}_* = k(\mathbf{t}_*, \mathbf{T})$ is the vector of covariances between the test point and the training set, $\mathbf{D}$ is a 2×2 diagonal matrix with the variances of the noise processes for latitude and longitude δ_j^2, and $\mathbf{k}_{**} = k(\mathbf{t}_*, \mathbf{t}_*)$. Finally, we minimize the negative marginal log-likelihood (MLL) of the output vectors with respect to the training data in determining the optimal hyperparameters.

$$-\log(p(\mathbf{Y}|\mathbf{T}, \Theta)) = \frac{1}{2}[\mathrm{vec}(\mathbf{Y})^\top \boldsymbol{\Sigma}^{-1}\mathrm{vec}(\mathbf{Y}) + \log(\det(\mathbf{K}_t)) + |\Theta|\log(2\pi)], \tag{6}$$

where Θ is the set of model parameters, $|\Theta|$ denotes the cardinality of the set of model parameters, which include both kernel-specific parameters as well as the weights associated with each kernel component (discussed further in Section 3.4), $\boldsymbol{\Sigma} = \mathbf{K}^f \otimes \mathbf{K}^t + \mathbf{D} \otimes \mathbf{I}$, and $\det(\mathbf{K}_t)$ is the determinant of the $\mathbf{K}_t$ matrix. We emphasize that the elements of $\mathbf{K}^f$ are treated as free parameters which are learned through Eq. (6). The negative marginal log-likelihood provides a target function for kernel learning. The first component, $\mathrm{vec}(\mathbf{Y})^T \Sigma^{-1}\mathrm{vec}(\mathbf{Y})$, estimates the model fit, while the second and the third terms, $\log(\det(\mathbf{K}_t)) + |\Theta|\log(2\pi)$, act as the regularization (Rasmussen and Ghahramani, 2000).

3.3. Physics-regularized GP

Only using the relationship $\mathbf{T} \rightarrow \mathbf{Y}$ to generate synthetic data can be problematic, as unconstrained GPs may produce predictions that are inconsistent with real-world conditions. We can indeed regularize the framework we have thus described through measurements of velocity and bearing, which are realizations of physics-based constraints acting on every user. Incorporating physical knowledge helps reduce the risk of overfitting and improves the likelihood of model generalization since the physical knowledge provides data augmentation. We derive observations of these variables with

$$v_\lambda = \frac{dy_\lambda}{dt}, \quad v_\phi = \frac{dy_\phi}{dt}, \quad v = \sqrt{v_\lambda^2 + v_\phi^2} \tag{7}$$

$$\beta = \arctan\left(\frac{v_\phi}{v_\lambda}\right) = \arctan\left(\frac{dy_\phi}{dy_\lambda}\right) \tag{8}$$

where v_λ and v_ϕ denote the velocity in the horizontal and vertical direction, respectively, v denotes the overall velocity, and β denotes the bearing.

One issue with using physical variables is the noise that may be introduced by irregular sampling of training data. That is, if time gaps between data points are large, the estimated segment velocities and bearings may not be representative of the distribution of

physics within that period. To get around this, Wang et al. (2022b) proposed to sample the physical observations from the posterior of the GP modeling $\mathbf{T} \rightarrow \mathbf{Y}$, thereby retaining control over where $\mathbf{P}$ is observed by utilizing a continuous function. A drawback of this approach is that the resulting likelihood function is difficult to solve, necessitating the calculation of the evidence lower bound (ELBO). Additionally, it significantly increases the computational demand, raising the complexity of the conditional GP from $O(n^3)$ to $O(n^3 + m^3)$, where m represents the number of sampled locations for $\mathbf{P}$.

Alternatively, Álvarez et al. (2009) and Alvarez et al. (2013) propose a different strategy that sidesteps the problem by embedding physical knowledge directly into the kernel's structure, rather than incorporating it as part of the training data. While this method simplifies the model, it also imposes limitations on the selection of kernels. Specifically, it restricts the choice to simpler and smoother kernel functions, which in turn constrains the ability to incorporate complex physical knowledge into the model through more sophisticated and adaptable kernels.

To mitigate the introduction of noise from data points with extensive time gaps, we simply assign each data point the same label until the time gap between the next observation exceeds a threshold (in our case, we used 1 h). This essentially labels continuous trajectories with the same unique identifier. For the first point of each unique trajectory, we assign a velocity and bearing of zero as we cannot observe the physical variables from the most recent trajectory (which would be > 1 h ago). This avoids having to drop training data with noisy physical variables without complicating the estimation procedure.

After ensuring the training data is efficiently sampled, adding physical variables to the posterior of the temporal GP translates to sampling from a larger covariance matrix that also considers physical correlations between the physical variables and their interactions with the temporal variables.

Assumption 3. Let $\mathbf{X} = \begin{bmatrix}\mathbf{T} & \mathbf{P}\end{bmatrix}$ such that a data vector $\mathbf{x}_i$ has both temporal and physical variables. Given the set of physics observations $\mathbf{P}$, the set $\mathbf{f}_j = [f_j(\mathbf{x}_1), \dots, f_j(\mathbf{x}_n)]$ follows a normal distribution such that $p(\mathbf{f}_j|\mathbf{X}) = \mathcal{N}(\mathbf{0}, \mathbf{K}_{comp})$, where $\mathbf{K}_{comp}$ is the covariance matrix with entries $[\mathbf{K}_{comp}]_{i,g} = k(\mathbf{x}_i, \mathbf{x}_g)$.

The inclusion of the physical variables is compatible with the multi-task kernel structure, as the kernel in Eq. (3) becomes $\mathbf{K}_{comp} = \mathbf{K}^x(\mathbf{X}, \mathbf{X}) \otimes \mathbf{K}^f(\mathbf{y}_\lambda, \mathbf{y}_\phi)$. This then changes the probability of observing a location $f_j(\mathbf{x}_*)$ at unobserved input $\mathbf{x}_*$ to be conditional on not only time but physics as well.

Assumption 4. The inferred location $f_j(\mathbf{x}_*)$ of a new input $\mathbf{x}_*$ conditioned on the training data is assumed to be distributed with the following form:

$$p(f_j(\mathbf{x}_*)|\mathbf{x}_*, \mathbf{T}, \mathbf{P}, \mathbf{Y}, \delta_j^2) \sim \mathcal{N}(\boldsymbol{\mu}_{**}, \boldsymbol{\sigma}_{**}^2), \tag{9}$$

where $\boldsymbol{\mu}_{**}$ and $\boldsymbol{\sigma}_{**}^2$ are defined as before but using the larger covariance matrix $\mathbf{K}_{comp}$.

This process transforms the MLL shown in Eq. (6) to

$$-\log(p(\mathbf{Y}|\mathbf{T}, \mathbf{P}, \Theta)) = \frac{1}{2}[\mathrm{vec}(\mathbf{Y})^\top \boldsymbol{\Sigma}^{-1} \mathrm{vec}(\mathbf{Y}) + \log(\det(\mathbf{K}_{comp})) + |\Theta| \log(2\pi)]. \tag{10}$$

The impact of including physical variables is non-negligible, both for estimation and inference (the latter of which we discuss in Section 3.5). Only looking at the temporal learning problem $\mathbf{T} \rightarrow \mathbf{Y}$ risks letting the GP remain too unconstrained, leading to illogical travel behavior predictions or unfeasible routes. For example, a well-fit GP with no physics-regularization tends to predict a user hovering around an activity location (rather than staying stationary while not on a trip), potentially leading to a false "walking trip" interpretation. On the other hand, a well-fit GP with physics-regularization largely solves this issue, as the model is statistically informed on whether the user is moving or remaining stationary.

3.4. Greedy algorithm for multiple kernel learning

Multiple kernel learning comprises two steps: the first step is to select a set of base kernels that are suitable to the context of the study (in this case, it is human mobility patterns) and the second step is to identify the optimal combination of the base kernels.

3.4.1. Selecting base kernels

The selection of base kernels is based on two key features of human mobility patterns. The first one relates to that observations of human mobility at nearby time intervals tend to be geographically close to each other. This concept extends Tobler's first law of geography (Tobler, 1970), which states that things in close proximity are more closely related than those far apart. Such dependencies can be captured rather straightforwardly using smooth functions derived from a variety of kernels—in this work, we consider the squared exponential (SE), rational quadratic (RQ), and Matern (MAT) 5/2 kernels:

$$k_{SE} = \eta \exp\left(-\frac{|x_i - x_g|^2}{2l^2}\right), \tag{11}$$

$$k_{RQ} = \eta \left(1 + \frac{|x_i - x_g|^2}{2\alpha l^2}\right)^{-\alpha}, \tag{12}$$

$$k_{MAT_{5/2}} = \eta \left(1 + \frac{\sqrt{5}|x_i - x_g|}{l} + \frac{5|x_i - x_g|^2}{3l^2}\right) \exp\left(-\frac{\sqrt{5}|x_i - x_g|}{l}\right), \tag{13}$$

where $|x_i - x_g|$ represents the Euclidian distance between any pair of inputs x_i and x_g; η is a scale parameter[3]; the lengthscale l determines the smoothness of the function; and the scale mixture α determines the relative weight of large- and small-scale variations in the data.

The SE kernel is characterized by infinite differentiability and is defined by a single lengthscale parameter. It is suitable for capturing smooth and gradual variations in the data. The RQ kernel introduces an additional scale mixture parameter that controls the smoothness of the kernel function and therefore can model both short- and long-range dependencies in the data. It can capture various patterns, from smooth variations to abrupt changes, making it suitable for datasets with diverse characteristics. Finally, the Matern 5/2 kernel exhibits differentiability up to the second order and is often favored when there is prior knowledge that the underlying process has a roughness level between that of the SE and the Matern 3/2 kernels. The Matern 5/2 kernel can capture moderate levels of non-smoothness in the data while still maintaining a certain degree of smoothness.

The second feature as noted in Section 3.1 is that individuals' travel behavior tends to center around a few important locations, such as home and work and there tend to be rhythms or cyclical patterns of different lengths with respect to how individuals conduct activities. To account for these, we further introduce categorical variables to capture cyclical patterns of different lengths. We represent calendar-based structures like days, weeks, and months as binary variables using a one-of-k encoding. For example, as the days of the week can take one of seven values $\{Mo, Tu, We, Th, Fr, Sa, Su\}$, a one-of-k encoding of We will correspond to $\{0, 0, 1, 0, 0, 0, 0\}$. We embed multiple categorical inputs in a GP framework by multiplying the same kernel across the one-hot encodings:

$$k_{RQ_{cat}} = \prod_{u=1}^{d} k_{RQ_u}. \tag{14}$$

Eq. (14) is a product kernel containing a unique lengthscale parameter for each input dimension being multiplied across the function. A small optimal lengthscale for a categorical variable implies a relatively low correlation among data in that category, while a large optimal lengthscale implies a high correlation. In addition to the one-hot encoding strategy, we also use the periodic kernel (Eq. (15)), which allows GPs to model functions that repeat themselves:

$$k_{PER} = \exp\left(-\frac{2\sin^2(\pi|x_i - x_g|/p)}{l^2}\right), \tag{15}$$

where the period length p determines the distance between repetitions of the function. Due to its sinusoidal nature, the periodic kernel is a better fit for continuous input dimensions that are unbounded in the input space.

3.4.2. Learning multiple kernel structures

The basic rationale of MKL builds on the possibility to create composite kernels comprising base kernels. So, the task at hand is kernel function discovery, i.e., finding the optimal structure for the composite kernel function that comprises a set of base ones. Kernel function discoveries have been done in other domains that aim to find concrete analytic forms of the kernels (Ong et al., 2002; Barla et al., 2003; Wilson, 2014; Gibbs, 1998), but there have been no specialized kernel functions developed for individual mobile data. In this paper, we develop a multiple kernel learning algorithm that is data-driven.

A GP's kernel is only valid if it has a PSD covariance matrix (Rasmussen and Williams, 2006). The direct sum or the tensor product of two valid kernels is also a valid kernel (for proof, we refer the reader to Section 4.2.4 in Rasmussen and Williams (2006)). To put it another way, the set of PSD kernels is closed under sum and product operations. 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 which each depend 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 a sum of one-dimensional functions, and hence the function $f(\mathbf{x}, \mathbf{y}) = f_x(\mathbf{x}) + f_y(\mathbf{y})$.

Our MKL learning is essentially learning about six components: target function, the base learner, the learning method, the functional form, the training method, and the computational complexity (Gönen and Alpaydın, 2011). We use the negative marginal log-likelihood (Eq. (6)) as our target function to assess the model's goodness of fit. However, simply comparing MLLs after optimizing kernel parameters would result in a bias in favor of more complex models. We avoid this overfitting issue by approximating the integral of the marginal likelihood over all free parameters using the Bayesian information criterion (BIC) (Schwarz, 1978):

$$\mathrm{BIC}(n, |\Theta|) = |\Theta| \log(n) - \log(p(\mathbf{Y}|\mathbf{X}, \Theta)). \tag{16}$$

Our greedy multi-kernel learning is essentially iteratively adding or multiplying new kernels until the BIC is minimized. The functional form of the derived kernel expressions can be both linear (i.e., additive) or non-linear (i.e., product)—each component has a weight parameter that determines its relevance and proportion of importance. We use the Adaptive Moment Estimation algorithm (Kingma and Ba, 2014) as the training method, which optimizes both the kernel weight parameters $\boldsymbol{\eta}$ and the base kernel parameters simultaneously. The computational complexity is a function of the algebraic operations involved, which determines the

[3] The scaling parameter can also be thought of as a weight in the composite kernel, determining the influence of each base kernel.

number of times the composite covariance matrix has to be inverted. We initialize the algorithm with a set of base kernels B, the maximum number of steps M, and a set of algebraic operations A.[4] This optimization procedure can be formally stated as:

$$\begin{aligned} &\underset{\boldsymbol{\eta},\mathbf{K},\Theta}{\operatorname{argmin}} \quad -\log(p(\mathbf{Y}|\mathbf{X},\Theta,\boldsymbol{\eta},k_{curr}))+|\Theta|\log(n), \\ &s.t. \quad k_{curr}=\sum_{h=1}^{M}\eta_h k_h \quad \text{where} \quad k_h=\begin{cases} k_{prev}+k_{base}, & \text{if addition at step } h \\ k_{prev}\times k_{base}, & \text{if multiplication at step } h \end{cases} \\ &\sum_{h=1}^{M}\eta_h=1, \quad \eta_h\geq 0 \quad \forall h\in 1,\dots,M. \end{aligned} \tag{17}$$

As illustrated in Fig. 1, we first estimate the BIC of each kernel in B and choose the lowest one. We then begin the greedy procedure: we apply each operation in the set of algebraic operations A to the chosen kernel and every available kernel in B (that is, they are added or multiplied) and choose the combination with the lowest BIC. This procedure is repeated until no new kernels result in a lower BIC than that of the existing kernel or until we reach M. The convergence properties of this algorithm depend largely on the complexity of the training data—if the data includes a large timeframe and the user displays multiple periodicities, trends, or random variations, the algorithm may not converge until we reach M. Otherwise, only one or two steps can be sufficient in describing the user's behavior.

3.5. Model inference

So far, we have dealt with the estimation of the posterior for the GP model. However, inference using physical variables presents a significant obstacle as it is a random variable determined by both temporal and spatial factors (locations), the latter of which are unobserved. Thus, prior to generating a set of synthetic locations $\mathbf{Y}_{gen}$ using the conditional model, we must specify a model to predict the unobserved physical constraints $\mathbf{P}_{gen}$ as a function of time and space. For the rest of the paper, $\mathbf{T}$, $\mathbf{P}$, and $\mathbf{Y}$ will be represented with $\mathbf{T}_{cond}$, $\mathbf{P}_{cond}$, and $\mathbf{Y}_{cond}$ to distinguish the fact that they belong to the conditional model, while sets relating to the generative model will have the subscript "gen".

Inspired by Lasserre et al. (2006), we propose a conditional-generative inference model. The generative component is for realizations of physical variables beyond the training data. Unlike LFM and the physics-regularized GP proposed by Wang et al. (2022b), the physical variables in our method are determined partially by the output of the conditional model (coordinates). Therefore, we first generate a set of noisy locations $\bar{\mathbf{Y}}_{gen}=[\bar{\mathbf{y}}_{gen,1},\dots,\bar{\mathbf{y}}_{gen,m}]^\top$ induced by a set of times $\mathbf{T}_{gen}$ using the conditional multi-task GP $f^{\bar{y}}\sim\mathcal{GP}(\mathbf{0},\mathbf{K}_t)$ via Eqs. (4) and (5), where $[\mathbf{K}_t]_{i,g}=k_t(\mathbf{t}_i,\mathbf{t}_g)$.

To estimate $\mathbf{P}_{gen}$, we require a second posterior that takes in spatial and temporal observations $\mathbf{Z}=\begin{bmatrix}\mathbf{Y}_{cond} & \mathbf{T}_{cond}\end{bmatrix}$ and approximates a function $f^p:\mathbf{Z}\rightarrow\mathbf{P}_{cond}$. This is achieved by defining another multi-task GP ("hidden GP" in Fig. 2) for the physical variables $f^p\sim\mathcal{GP}(\mathbf{0},\mathbf{K}_p)$ where $[\mathbf{K}_p]_{i,g}=k_{comp}(\mathbf{z}_i,\mathbf{z}_g)$. Sampling from the posterior distribution

$$\mathbf{P}_{gen}\sim p(\mathbf{T}_{gen})p(f^p|\bar{\mathbf{Y}}_{gen},\mathbf{Y}_{cond},\mathbf{T}_{cond},\mathbf{P}_{cond}) \tag{18}$$

where we use the marginal independence of $\mathbf{T}_{gen}$ to simplify the joint conditional distribution. The set of generated physical variables are then incorporated as inputs to the physics-regularized GP model described in Section 3.3. Finally, we sample from the physics-regularized GP $f^y\sim\mathcal{GP}(\mathbf{0},\mathbf{K}_{comp})$ where $[\mathbf{K}_{comp}]_{i,g}=k_{comp}(\mathbf{x}_i,\mathbf{x}_g)$.

Fig. 2 shows the workflow for the proposed model, which begins with learning the functional form of the physical and temporal kernels k_p and k_t via the proposed MKL algorithm. These are then multiplied using the Kronecker product into the composite kernel k_{comp}. This learned kernel is used two-fold: First to embed the physics knowledge as a function of time and space using a GP ("Physics Embedding"), then to generate synthetic data points given $\mathbf{T}_{gen}$ and estimated physical variables $\mathbf{P}_{gen}$. In Fig. 2, the Greedy MKL and Physics Embedding modules are estimation problems, while the Physics-informed multi-task GP inference module shows the inference problems.

Theorem 1. *The empirical distribution functions used for sampling the generated synthetic data $\mathbf{Y}_{gen}$ and $\mathbf{P}_{gen}$ are constructed only using the basis vectors generated from the conditional and hidden GPs, respectively.*

Proof. Let $F_{\mathbf{Y}_{gen}}$ and $F_{\mathbf{P}_{gen}}$ denote the empirical distribution functions for $\mathbf{Y}_{gen}$ and $\mathbf{P}_{gen}$, respectively. Due to Mercer's theorem,[5] each sample of $\mathbf{P}_{gen}$ and $\mathbf{Y}_{gen}$ is a linear combination of the basis functions determined by the covariance matrices $\mathbf{K}_p$ and $\mathbf{K}_{comp}$. Because $F_{\mathbf{Y}_{gen}}$ and $F_{\mathbf{P}_{gen}}$ are constructed from these samples, they will also be linear combinations of these basis vectors generated from the conditional and hidden GPs. □

[4] For now, we only allow addition and multiplication as other operations do not always result in PSD kernels.

[5] For details, see Section 4.3 of Rasmussen and Williams (2006).

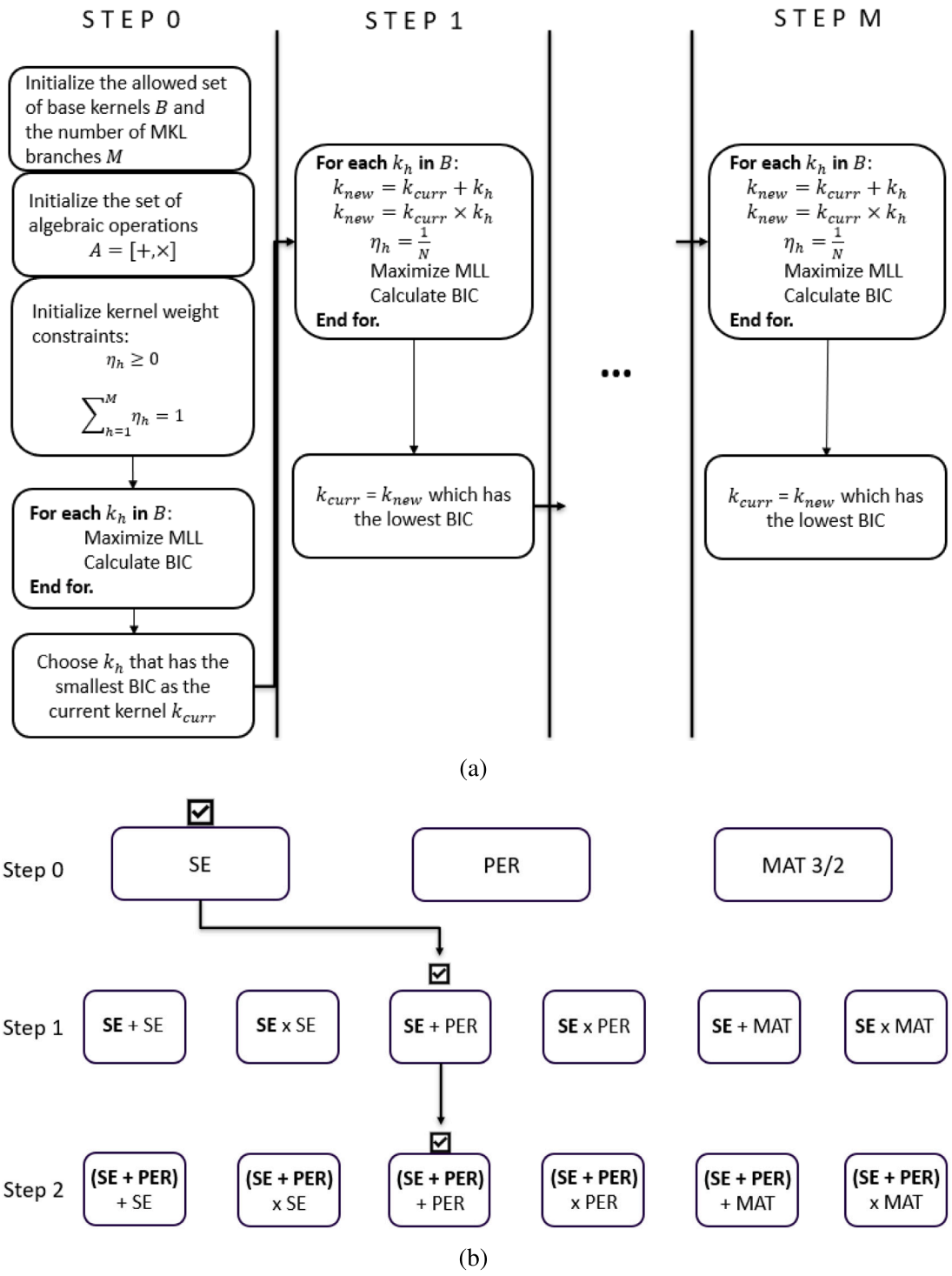


Fig. 1. (top) Proposed algorithm for greedy multiple kernel learning; (bottom) example greedy learning tree with three steps (checkmark denotes lowest BIC option).

The implication of Theorem 1 is three-fold. First, the generated $\mathbf{P}_{gen}$ and $\mathbf{Y}_{gen}$ reflect the patterns, trends, and noise characteristics present in the observed data $\mathbf{P}_{cond}$ and $\mathbf{Y}_{cond}$. Second, the model is not just statistically informed but also physically informed. This means that any generated data point respects the underlying physical laws or constraints that have been encoded through the model

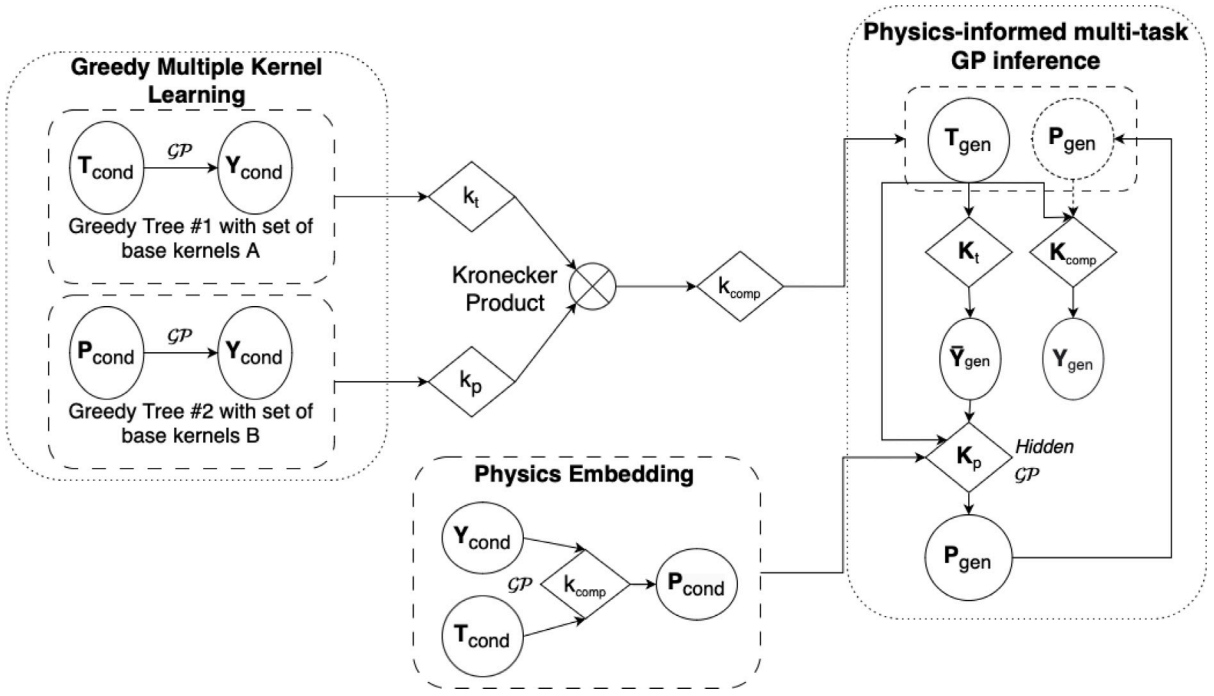


Fig. 2. Physics-regularized GP inference for passively-generated mobile data. Circles indicate variables, while diamonds indicate estimation (kernel function) or inference (kernel matrix) processes. A dashed line around a circle indicates an unobserved variable.

framework. These constraints might represent known relationships between variables that must be maintained, such as the laws of motion in a physical system. Third, by adhering to the learned relationships and physical constraints, the synthetic data generation process is not just mimicking the observed data but generalizing from it. This enables the model to create data points that are plausible under the model's understanding of the domain, even if they were not explicitly present in the training data.

4. Numerical experiments

In this section, we document the results of four experiments that evaluate the formulation we have thus presented. In Sections 4.1 and 4.2, we discover the best-fitting covariance structure for temporal and physical features to create a composite kernel, respectively. In Section 4.3, we compare the performance of this composite kernel against alternative formulations over many users in a passively-collected mobile dataset. Finally, to analyze our framework's generalizability, we conduct another set of experiments in Section 4.4 on Microsoft Asia's GeoLife dataset using trips of varying modes. All class and function objects relevant to this experiment can be found on GitHub: https://github.com/ekinugurel/physics-regularized-MTGP. Additionally, Appendix B outlines a set of algorithms we use throughout the experiments.

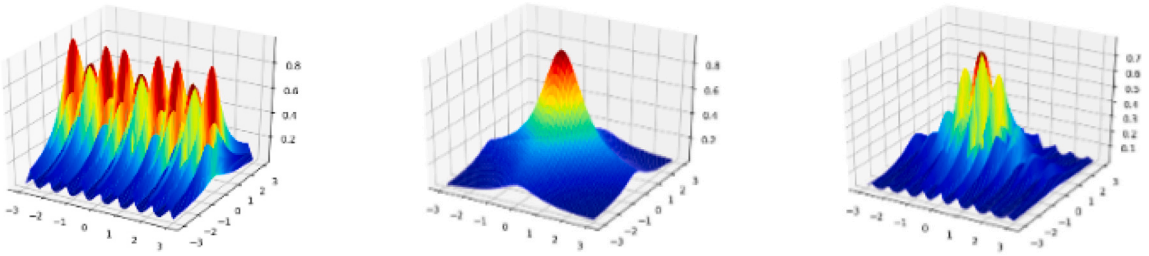
Before all experiments, we perform data standardization (z-score normalization) such that each feature is centered at 0 with a standard deviation of 1. For example, $v_i \in \mathbf{v}$ (the ith observation of velocity in $\mathbf{v}$) can be standardized with

$$\tilde{v}_i = \frac{v_i - \bar{v}}{s_{\mathbf{v}}} \tag{19}$$

where $\bar{v}$ is the sample's average velocity and $s_{\mathbf{v}}$ is the standard deviation of the sample's velocity. As different variables have different units, doing so simplifies the process of optimization via gradient descent–allowing the algorithm to converge more efficiently by ensuring that the scales of all features are comparable. This standardization also aids in preventing certain variables with larger magnitudes from disproportionately influencing the optimization process, promoting a more balanced and effective model training.

4.1. Temporal structure discovery using multiple kernel learning

We show the process of discovering the temporal structure associated with an individual's mobile data. Specifically, we conduct a test on data over three weeks, using the first two weeks for training the model, and measure its accuracy using the last week. We run the MKL algorithm on the training set with the following base kernels: SE (Eq. (11)), PER (Eq. (15)), and RQ (Eq. (12)). We initiate two instances of the PER kernel: one initialized with $p = 1440$ (representing a day) and the other initialized with $p = 10{,}080$ (representing a week). After a sensitivity analysis, we settle on $M = 3$ (as more steps do not result in lower BICs) and initialize the kernel weight constraints. The resulting kernel is a set of rational quadratic kernels multiplied on all input dimensions and one



(a) $k_{PER,0} \times k_{RQ,1}$ (b) $k_{RQ,1} \times k_{RQ,2}$ (c) $k_{PER,0} \times k_{RQ,1} \times k_{RQ,2}$

Fig. 3. Components of the single-task temporal kernel in a 3-D space.

Table 1
Optimal kernel parameters for long gap imputation example.

Parameter	Value	Explanation
Initial $p_{a,1}$	1440 min = 1 day	Initial value for period length in PER_a
Initial $p_{b,1}$	10,080 min = 1 week	Initial value for period length in PER_b
η_a	0.27	Optimal weight of kernel component a
η_b	0.73	Optimal weight of kernel component b
$p_{a,1}$	629 min = 0.44 days	Optimal period length of PER_a
$p_{b,1}$	5290 min = 3.67 days	Optimal period length of PER_b
$l_{a,RQ,t_d=3}$	39.4[a]	Lengthscale of RQ_a for $t_d = 3$
$l_{a,RQ,t_{am}=1}$	18.4[a]	Lengthscale of RQ_a for $t_{am} = 1$
$l_{b,RQ,t_d=6}$	55.6[a]	Lengthscale of RQ_b for $t_d = 6$
$l_{b,RQ,t_{am}=1}$	18.7[a]	Lengthscale of RQ_b for $t_{am} = 1$

[a] Denotes the highest lengthscale among categorical/binary variable.

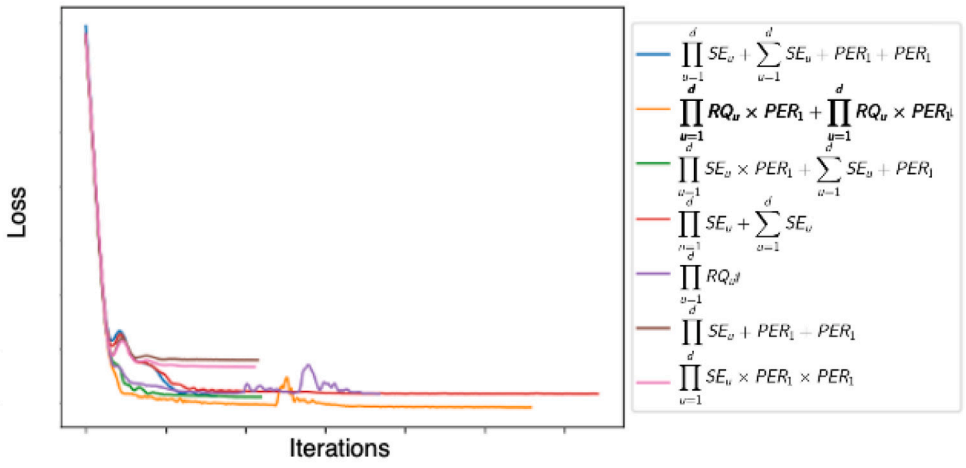
periodic kernel on the Unix time dimension. Fig. 3 visualizes components of the discovered kernel in a single-task space. To evaluate the success of our algorithm, we compare this kernel with similar kernels that can be constructed using the above base kernel set.

We measure model accuracy with three mobility metrics: spatial burstiness B_s, representing the distribution pattern of travel distances between consecutive stay points (Kim and MacEachren, 2014); radius of gyration r_g, indicating the characteristic travel distance of a user during a period (Song et al., 2010); and home location inaccuracy ΔH, measuring the haversine distance between user's imputed and actual home location. Our results are detailed in Fig. 4, while the optimal kernel parameters are shown in Table 1. From Fig. 4a, the convergence behavior of each kernel is discernible, with most of them stabilizing as the number of iterations increases, indicating that the models are reaching an optimal point in terms of the parameters being estimated. A few kernels exhibit some volatility in their loss values, even in later iterations, which could suggest a less stable model fit. Fig. 4b shows the generated traces using the discovered kernel and the temporal GP based on the training data. Sigma 1 and Sigma 2 represent 1 and 2 standard deviations in the normalized distribution. The coefficients of variation using de-normalized generated traces are 0.127 for the latitudes and 0.026 for the longitudes, respectively, suggesting that GP does a good job in estimating the variance of the predictions.

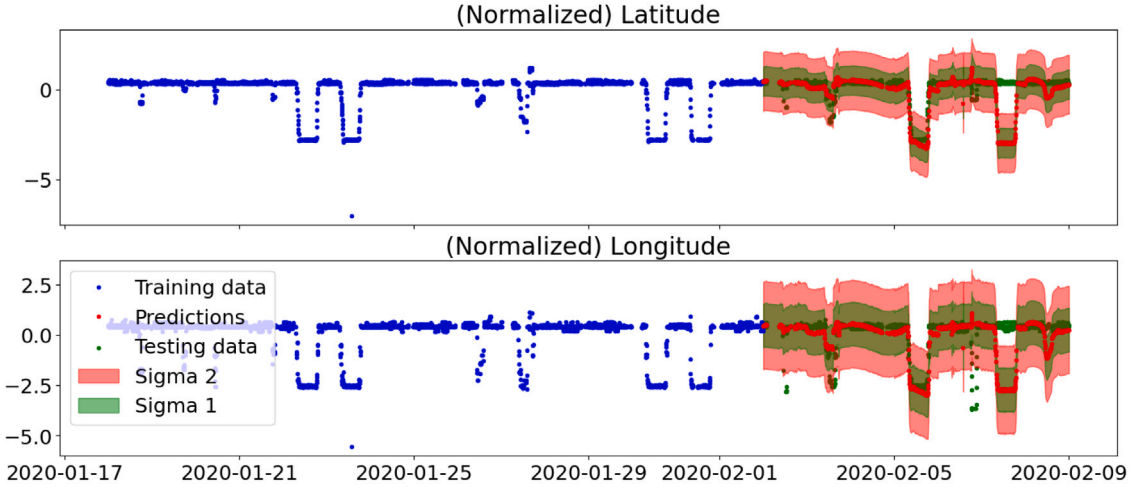
Fig. 4c suggests that there is a trade-off in the performance of a composite kernel depending on the mobility metric being prioritized and that the impact of the chosen kernel is non-negligible in this context. Notably, while certain kernels may not be optimal for all metrics, they still provide some degree of insight into the mobility patterns. A kernel that may have a higher loss or error for one metric might still be useful for understanding certain aspects of mobility through other metrics. This may be because the mobility metrics analyzed here are not necessarily aligned with the objective function of the MKL process. We discuss ideas for future studies on this topic in Section 5.

In practical applications, composing a kernel would then depend on the specific aspect of mobility one aims to capture. If the goal is to have a balanced model across various metrics, then kernels that show moderate performance across the board might be preferred. However, if the objective is to excel in one particular metric, such as predicting home location with high accuracy, then the kernel with the lowest error for ΔH would be chosen, despite potential compromises in other areas.

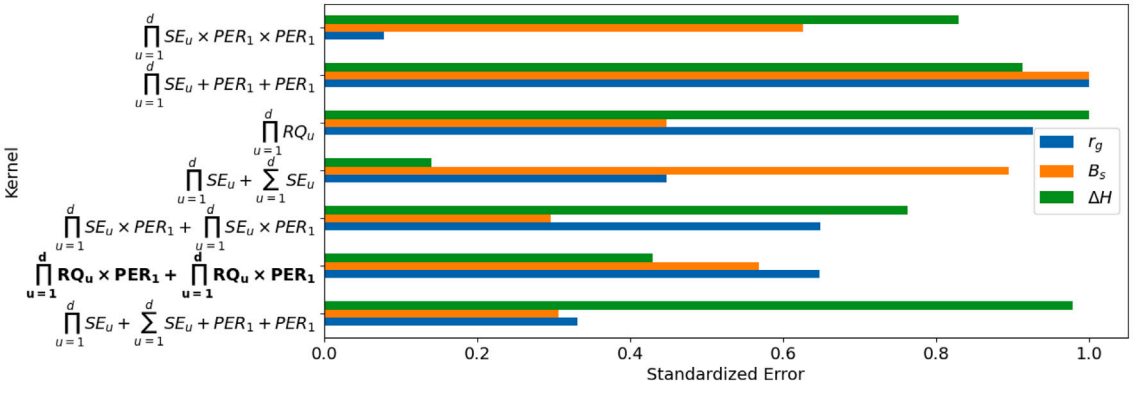
Analyzing optimal kernel parameters highlights the capacity of our model to uncover individual mobile data generation processes. The last four rows in Table 1 show the highest characteristic lengthscale along each categorical dimension—a high smoothing parameter for an input dimension suggests low variation along that dimension in the function being modeled, and vice versa. We notice that the example user has high correlations in their behavior on Thursdays ($t_d = 3$), Sundays ($t_d = 6$) and in the AM peaks ($t_{am} = 1$) compared to the other days of the week and the PM peak, respectively. The optimal period lengths also suggest that the user's periodicities occur at shorter intervals than initially assumed—the first periodic kernel is optimized at 0.44 days and the second at 3.67 days. Relatedly, the daily kernel component influences the model less than the weekly component judging by the weights η_a and η_b.



(a)



(b)



(c)

Fig. 4. (a) Loss function progression and convergence behavior of different functional forms for a composite kernel; (b) Generating one week of trips using discovered kernel and the temporal GP; (c) Derived mobility metrics by kernel. Bold denotes the optimal kernel. r_g denotes radius of gyration, B_s denotes spatial burstiness, and ΔH denotes distance between user's imputed and actual home location.

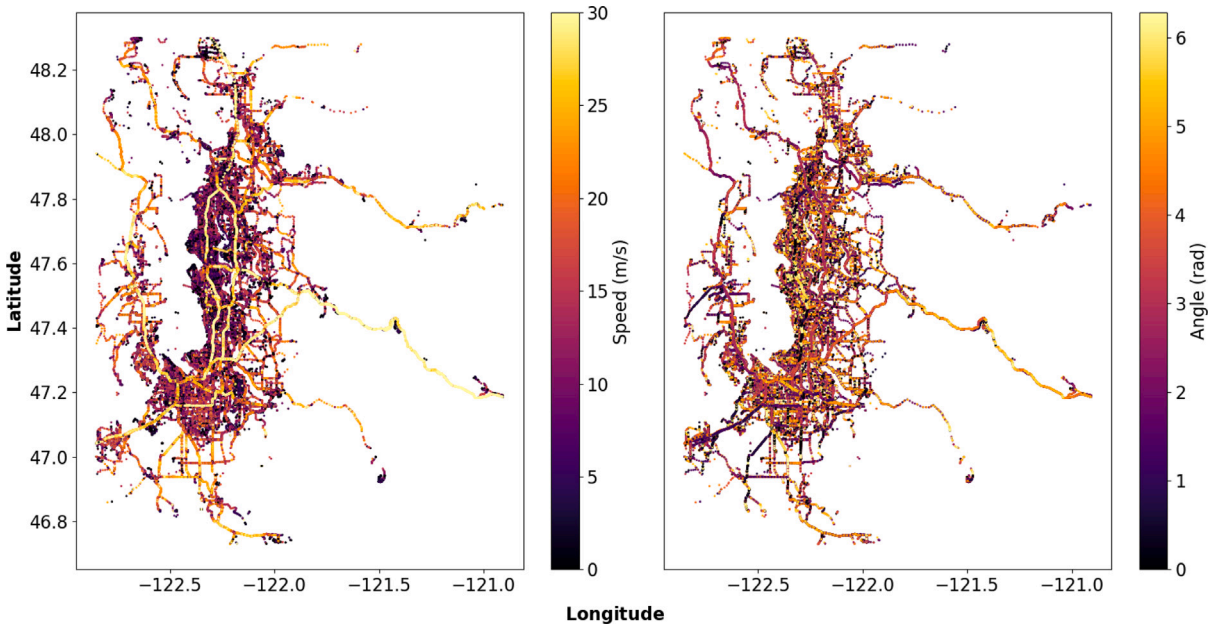


Fig. 5. Speed and bearing constraints in King County, Washington.

4.2. Learning the physical kernel

In this section, we discuss the process of learning the optimal physical kernel k_p. Fig. 5 shows discrete observations of average segment speeds and angular directions observed from 10 users' trips over six months. We note that one can deduce roadway types by the average speed observed on them (e.g., the clear vertical lines on the left are SR-99 and I-5 respectively).

To learn the form of the covariance function for physics-based variables, we apply the greedy MKL algorithm on the training set with the following base kernels: SE (Eq. (11)), and RQ (Eq. (12)), MAT (Eq. (13)). This is one realization of the $\mathbf{P}_{cond} \rightarrow \mathbf{Y}_{cond}$ GP shown on the left side of Fig. 2. The resulting kernel (Eq. (20)) has an additive form and uses the RQ (Eq. (12)) and MAT 5/2 (Eq. (13)) kernels. Fig. 6 shows predictions on one week of velocities and bearings after the covariance matrix associated with Eq. (20) was estimated for two weeks of training data.

$$
\begin{aligned}
k_p = {} & \eta_{RQ,1}\left(1+\frac{|\mathbf{x}_i-\mathbf{x}_g|^2}{2\alpha_1 l_{RQ,1}^2}\right)^{-\alpha_1} \\
& + \eta_{MAT,1}\left(1+\frac{\sqrt{5}|\mathbf{x}_i-\mathbf{x}_g|}{l_{MAT,1}}+\frac{5|\mathbf{x}_i-\mathbf{x}_g|^2}{3l_{MAT,1}^2}\right)\exp\left(-\frac{\sqrt{5}|\mathbf{x}_i-\mathbf{x}_g|}{l_{MAT,1}}\right)+\eta_{RQ,2}\left(1+\frac{|\mathbf{x}_i-\mathbf{x}_g|^2}{2\alpha_2 l_{RQ,2}^2}\right)^{-\alpha_2} \\
& + \eta_{MAT,2}\left(1+\frac{\sqrt{5}|\mathbf{x}_i-\mathbf{x}_g|}{l_{MAT,2}}+\frac{5|\mathbf{x}_i-\mathbf{x}_g|^2}{3l_{MAT,2}^2}\right)\exp\left(-\frac{\sqrt{5}|\mathbf{x}_i-\mathbf{x}_g|}{l_{MAT,2}}\right).
\end{aligned}
\tag{20}
$$

4.3. Passively-collected mobile dataset

We randomly select a subset of 33 users with at least 10,000 observations and 3 months of data (from first data point until last). Within the selected 33, we only retain data from the month with the most observations. We minimize Eq. (10) using the mean and covariance structures in Eq. (9) to evaluate the performance of the physics-regularized GP formulation against two alternative formulations: One using only physical variables, and another using only temporal variables. We note that the former is unrealistic in practice, as variables like speed and bearing have to be derived from coordinates (which are assumed missing in our tests). However, we show the results as they illuminate the motivation for integrating the two types of variables.

We use the root mean square error (RMSE) and the mean standardized log loss (MSLL) as our performance metrics—the former is the root of the squared residual between the mean prediction and the target, averaged over the test set; the latter is a probabilistic measure that takes into account the uncertainty of the predictions. The MSLL incorporates the logarithmic transformation of the

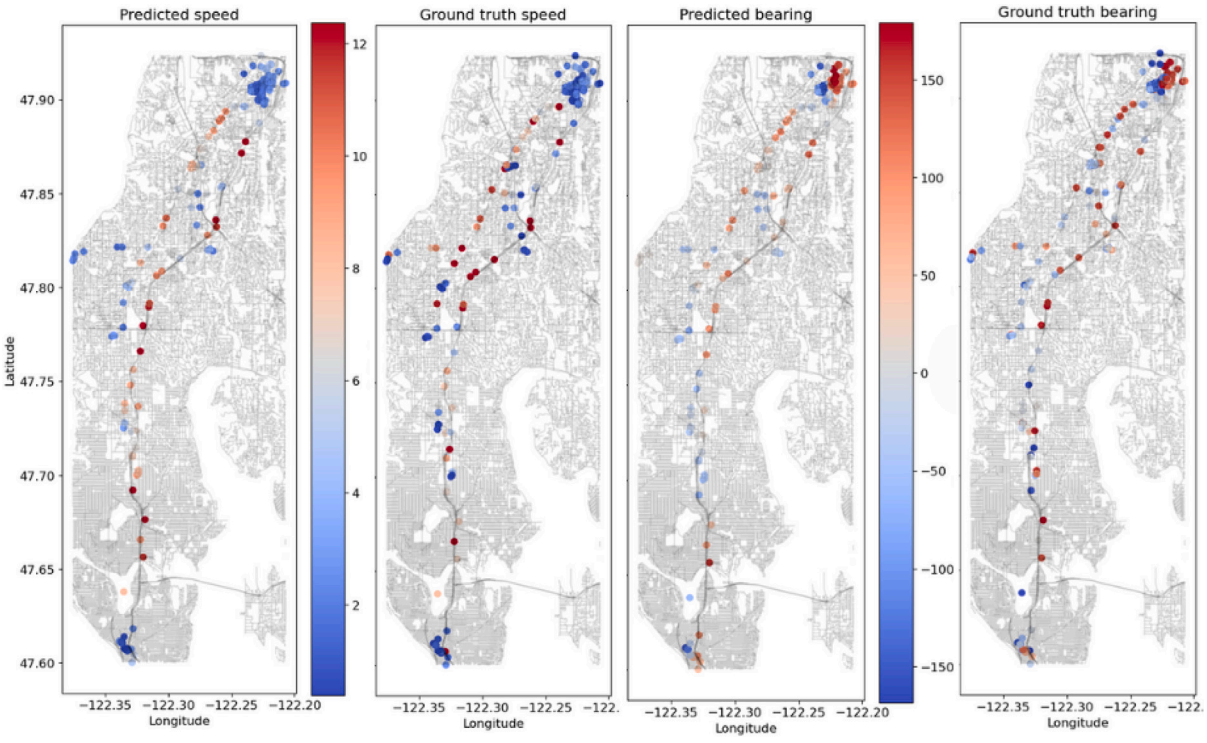


Fig. 6. Speed and bearing predictions and true observations for a Spectus user.

predicted and observed values, normalized by the associated uncertainty. This allows for a more nuanced assessment of the model's performance, as it captures the relative differences between predicted and observed values in a probabilistic manner. We use these metrics to compare our predictions to various testing sets, allowing us to comment on the model's generalizability and the impact of the physics-regularization on bias and variance.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}, \tag{21}$$

$$MSLL = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{\log(\hat{y}_i + 1) - \log(y_i + 1)}{\sigma_i}\right)^2. \tag{22}$$

where y_i and $\hat{y}_i$ are respectively the true and predicted values of the dependent variable at index i and σ_i is the predicted standard deviation.

We remove consecutive chunks of data to test the model's accuracy in imputing time series, progressively increasing the size of the training set in each iteration while keeping the size of the test set constant. Specifically, we divided the dataset into three sequential folds for training and testing, ensuring chronological order is maintained. The first split (S1) uses the first and the second weeks of the month as training and testing sets, respectively; the second set (S2) then uses the first two weeks to train and the third week to test. Finally, the third set (S3) uses the first three weeks to train and the last week to test. This approach simulates a real-world scenario where a model is trained on an increasing amount of historical data and tested on the subsequent period. It also allows us to observe the performance of the model as it is exposed to more data over time.

Table 2 shows the median performance of each model across the three splits. The Physics-regularized GP outperforms the individual physics-only and temporal-only models in each testing chunk with respect to RMSE and MSLL. The increase in RMSE from S1 to S2 for all models and from S2 to S3 for the Temporal GP indicates a decrease in predictive accuracy as the training set size increases, possibly due to overfitting or increased model complexity. However, the Physics-regularized GP maintains consistently low RMSE values across all splits, demonstrating better generalization as more data is included. The steady performance improvement in this model suggests that the integration of both physical and temporal inputs helps to counteract overfitting and improve predictive accuracy as the dataset grows.

The MSLL is consistently lower for the Physics-regularized GP in all splits, indicating not only better prediction accuracy but also better calibrated uncertainty estimates. The consistent performance of the Physics-regularized GP across the splits in terms

Table 2
Models and median test set metrics for 33 users from Spectus.

Models	Metrics								
	RMSE			MSLL			RT		
	S1	S2	S3	S1	S2	S3	S1	S2	S3
Physical GP	0.105	0.128	0.130	0.767	0.816	0.855	**6.69**	**13.9**	**30.3**
Temporal GP	0.107	0.119	0.116	0.885	0.808	0.874	7.99	17.4	38.5
Physics-regularized GP	**0.103**	**0.119**	**0.113**	**0.882**	**0.779**	**0.849**	11.5	22.8	50.2

Note: S1, S2, and S3 represent three different time-series splits.

of MSLL suggests that it remains reliable and well-calibrated regardless of the amount of training data. Although the Physics-regularized GP estimates more input dimensions simultaneously than the other two models, its Kronecker product structure keeps the increase in training time relatively modest. The physical-only model has the fastest runtime, but the added computational cost of the Physics-regularized GP is justified by its improvements in both prediction and uncertainty calibration.

4.4. GeoLife dataset

To test the generalizability of our framework across multiple modes and populations, we conduct an experiment on Microsoft Asia's GeoLife dataset (Zheng et al., 2008, 2009, 2010), specifically using the subset of all GeoLife users who had mode labels available. The essence of this experiment is to assess how well our model can generate unobserved trips based on a training set of similar trips. Rather than a time-series-based experimental design where the training and testing sets are temporally sequential, we use a stratified sampling strategy (shown within Algorithm 5 in Appendix B) applied to all of one user's trips. That is, we cluster trips based on their start and end locations, and sample out of the set of similar trips to build the training set. We do this for two reasons: (1) in GeoLife, users may appear in the dataset over the course of years and data continuity is low (i.e., we do not see trips from the same user in consecutive days), making a time-series split more difficult; (2) Ensuring that the training set of trips is similar to the testing set allows us to better isolate the effect of physics-regularization, as a pair of start/end locations can have multiple routes (with presumably different underlying physics).

We assess the proposed exact GP formulation (PIMTGP) against three alternatives: a sparse GP model (SparsePIMTGP), an exact temporal GP using multiple kernel learning but without physics-regularization (TempMTGP), and a Long Short-Term Memory (LSTM) network (Hochreiter and Schmidhuber, 1997). Unlike probabilistic models, LSTM networks do not inherently provide measures of uncertainty. To facilitate a comparison between LSTM predictions and that of the GPs, we conduct Monte Carlo (MC) simulations with a dropout rate of 0.5 across 70 simulations during inference. This approach introduces randomness during inference to approximate the model estimation uncertainty of the LSTM. It is important to note that this does not equate to the prediction uncertainty provided by the GP models, which is derived from their probabilistic nature. Therefore, while MC dropout helps in assessing the robustness of the LSTM predictions, the comparison in probabilistic measures between LSTMs and GPs should be interpreted with caution as they reflect different aspects of uncertainty. We use the `Torch` implementation of LSTM with a mean squared error loss, optimized via Adam (Kingma and Ba, 2014) as with all other models. We choose hyperparameters via cross-validation and limit the number of MC simulations to 70 to maintain scalability.

The key idea of the sparse GP model is that we can learn a memory-efficient representation of the $n \times n$ covariance matrix using c inducing points, which capture most of the variation in the dataset (where $c \ll n$) (Snelson and Ghahramani, 2005). The benchmark formulation uses a variational approximation where the predictive distribution is given by

$$p(\mathbf{f}(\mathbf{x}_*)) = \int_o p(f(\mathbf{x}_*|\mathbf{o}))q(\mathbf{o})d\mathbf{o} \tag{23}$$

where $\mathbf{o}$ represents the function values at the c inducing points, $q(\mathbf{o}) = \mathcal{N}(\mathbf{m}, \mathbf{S})$. Here, $\mathbf{m} \in \mathbb{R}^c$ and $\mathbf{S} \in \mathbb{R}^{c \times c}$ are learnable parameters. Its multitask extension leverages the linear model of coregionalization (LMC) which assumes that each task j is the linear combination of some latent functions $\mathbf{g}(\cdot) = [g^{(1)}(\cdot), \ldots, g^{(Q)}(\cdot)]$ and

$$f_j(\mathbf{X}) = \sum_{v=1}^{Q} a^{(v)} g^{(v)}(\mathbf{X}), \tag{24}$$

where $a^{(v)}$ are learnable parameters.

Fig. 7 shows the performance of each model across seven evaluation metrics and three modes of travel–bike, walk, and bus. RMSE and MSLL are defined as before. The mean absolute error (MAE) and the median absolute deviation (MAD) are defined as follows,

$$MAE = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i|, \tag{25}$$

$$MAD = \text{median}(|y_i - \hat{y}_i|). \tag{26}$$

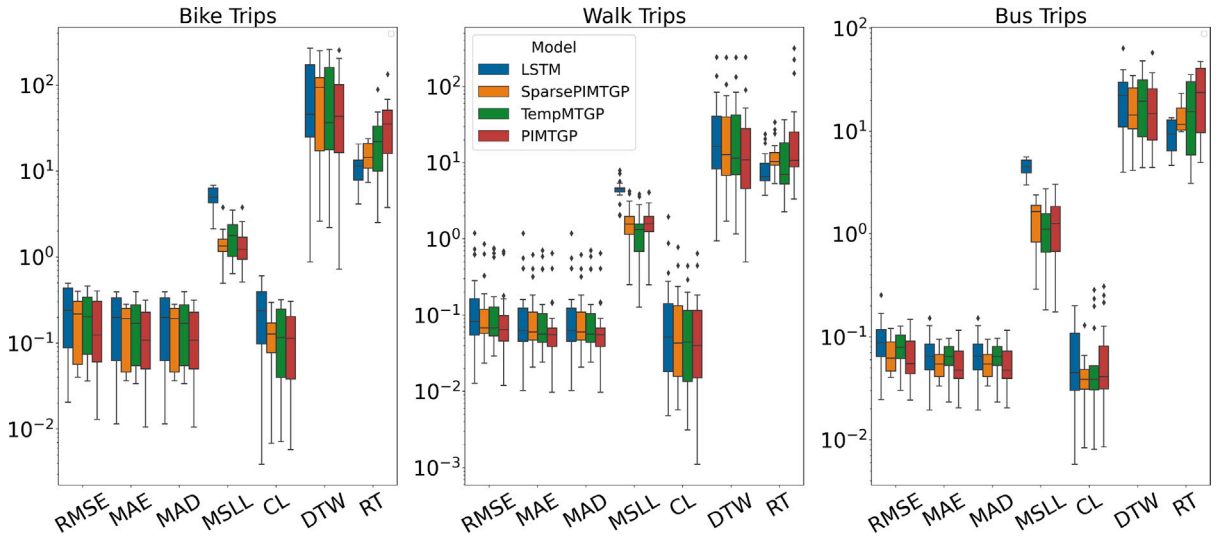


Fig. 7. Evaluation metrics for all models being compared, which include Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Median Absolute Deviation (MAD), Mean Standardized Log Loss (MSLL), Curve Length (CL), Dynamic Time Warping (DTW), and Training Runtime (RT). Y-axis is shown in log-scale. The lower the better for all metrics.

We also assess model performance with metrics specific to curves (or trajectories). These metrics better capture differences in predicted and actual route choice, an important variable in travel behavior research. The curve length (CL) measures the arc-length distance along the curve from the origin, capturing total travel distance in a given trip. Meanwhile, dynamic time warping (DTW) incorporates timestamps of trajectories when considering sequence alignment, capturing differences in speed of travel (Ugurel et al., 2024). A lower DTW value indicates a closer match whereas a higher value suggests a greater disparity between the compared sequences. Finally, we compare the training runtime (RT) of all methods.

From Fig. 7, we observe a few things: PIMTGP tends to outperform the other models in the classical error metrics (RMSE, MAE, and MAD) across all modes. In the probabilistic MSLL metric, we observe a clear distinction between GP-based models and the LSTM network. The MKL-equipped TempMTGP seems to be competitive with PIMTGP in this probabilistic measure, highlighting the importance of using Multiple Kernel Learning. In the trajectory comparison metrics (CL and DTW), the outcomes are fairly similar, with TempMTGP and PIMTGP tending to perform slightly better. Finally, taking a look at the runtimes paints a clear picture–the more complex models take longer to train but also tend to perform better at individual data generation. It is also worthy to note that the SparsePIMTGP has similar runtimes to the LSTM model while maintaining the core of our framework, opening a path to future scalable implementations. We believe this is a promising research avenue.

5. Discussion

To realize the potential of creating synthetic population for large scale simulations of individual mobility patterns, we need to develop a flexible and scalable model that can capture individual heterogeneity and also adapt to changes over time. These two properties: flexible and scalable, require nonparametric models whose forms can easily adapt. In this paper, we propose a physics-regularized multi-task Gaussian Process (GP) model for learning kernel structures that characterize individual movement patterns. Though there are other choices for nonparametric models such as deep neural networks, GP models are another promising method that offers several benefits over deep learning-based models. First, GPs are generally more interpretable than deep learning architectures, 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). A related second point is that kernel methods are more mathematically mature than their deep learning counterparts. Especially in exact GPs, all calculations are done in closed-form—this signals a potential for theoretical bounds on generalization. Additionally, the probabilistic framework of GPs facilitates robust uncertainty quantification, motivating the use of different metrics to capture this added benefit (Mohamed et al., 2022).

As noted earlier, the study is motivated by an emerging application of passively-generated mobile data that will have great potential when realized, a problem that can hamper the great potential, and the fact that few have looked into the problem at hand. That emerging application is synthetic population generation for large-scale simulations of cities for various needs such as state monitoring, policy formulation and evaluations, and building digital twins. Existing methods such as activity-based models or probability models are inflexible, unable to capture heterogeneity exhibited in individual mobility patterns and adapt to changing patterns over time. The reality now and in the future is that the quality of data is degrading (substantial sparsity or missingness associated with the data) with rising awareness to preserve privacy on the part of both data vendors and consumers. To fulfill the great potential that passively-generated mobile data has for synthetic population generation and to address the increasing issue of sparsity, an understanding of the underlying data generation process behind the observed individual mobile data is needed. This is the key theme of our paper: the development of a flexible modeling framework that can adapt to heterogeneity at the individual level.

We summarize our contributions as follows: We propose a conditional-generative GP framework to generate synthetic individual mobile data that integrates physical knowledge and provably replicates observed travel patterns. To account for heterogeneity at the individual level, we develop a data-driven multiple kernel learning approach to determine the most suitable spatiotemporal kernel for each user. The learned multiple kernel GP models can be viewed as mechanisms to uncover underlying data generation processes that result in the observed big mobile data, which in turn can be used to generate synthetic data mimicking the properties of collected data.

Moving forward, our research opens several avenues for advancement. Future studies can aim to explore alternative formulations for the MKL objective function that allow for the prioritization of different mobility metrics in the synthetic data generation process, tailoring the model to specific applications. Furthermore, new methods in one-step ahead forecasting strategies (i.e., predicting the next data point in a time series based on the observed values up to the current time point) infused with our framework could provide real-time insights and predictions, enhancing the model's applicability in dynamic environments. Finally, to address the computational burden often associated with GP models, studies should further investigate approximation techniques (as we do with the SparsePIMTGP) as well as algebraic methods, such as exploiting the grid structure in inputs, to streamline the model's efficiency without sacrificing accuracy. These approaches will be key to unlocking the production-level (i.e., scalable) potential of such methods.

CRediT authorship contribution statement

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

Data availability

The authors are not able to share the Spectus data due to privacy reasons. The GeoLife data is available in the Github link provided for this article: https://github.com/ekinugurel/physics-regularized-MTGP.

Acknowledgments

The authors are grateful to the funding support from the National Science Foundation, United States of America for the project titled as "A whole-community effort to understand biases and uncertainties in using emerging big data for mobility analysis" (award number 2114260). EU and CC are also supported by the Center for Teaching Old Models New Tricks (TOMNET), United States of America, a Tier 1 University Transportation Center through Grant No. 69A3551747116, as well as the Center for Understanding Future Travel Behavior and Demand(TBD), United States of America, a National University Transportation Center through grant 69A3552344815 and 69A3552348320.

Appendix A. List of notation used

See Table 3.

Table 3
Variable dictionary.

	General notation
λ	Longitude
ϕ	Latitude
v	Velocity
β	Bearing
Subscripts i and g	Indices for observations
Subscript j	Index for task
Subscript u	Index for input dimension
	GP notation
$\mathbf{Y}$	Matrix of outputs
$\mathbf{T}$	Matrix of temporal variables
$\mathbf{P}$	Matrix of physical variables
$\mathbf{X}$	Matrix of inputs including $\mathbf{T}$ and $\mathbf{P}$
ε	Gaussian white noise
d	Number of input dimensions
n	Number of training data points
m	Number of inferred data points
c	Number of inducing points
k	Covariance function of a GP
$\mathbf{K}$	Covariance matrix
$\mathbf{K}^f$	Inter-task covariance matrix
$\mathbf{k}$	Vector of covariances
$\otimes$	Kronecker product
$\boldsymbol{\mu}_*$	Posterior mean function of the multi-task GP
$\boldsymbol{\sigma}^2_*$	Posterior variance of the multi-task GP
$\boldsymbol{\mu}_{**}$	Posterior mean function of the physics-regularized multi-task GP
$\boldsymbol{\sigma}^2_{**}$	Posterior variance of the physics-regularized multi-task GP
Θ	Set of GP parameters
l_k	Lengthscale parameter of kernel k
α	Scale mixture parameter of an RQ kernel
p	Period length parameter for a PER kernel
	MKL notation
B	Set of base kernels
M	Number of MKL branches
A	Set of algebraic operations
η_h	Weight of kernel component h

Bold denotes vectors.
Capital Bold denotes matrices.

Appendix B. Dataset and software implementation

For Sections 4.1 through 4.3, we employ privacy-protected, passively-generated GPS data from Spectus, a U.S.-based data solution provider specializing in geospatial analytics. The dataset contains discrete GPS points of 2000 anonymous, opted-in individuals in the Greater Seattle Area between December 2019 and July 2020. The location data recorded includes timestamps, unique device identifiers, latitudes and longitudes, and a measure of data precision (i.e., a spatial radius for which the provider has 95% confidence in the reported coordinates). For the experiment in Section 4.3, we sequentially increased the number of training points in each split while keeping the number of testing points constant, as shown in Fig. 8.

For the experiment in Section 4.4, we use Microsoft Asia's GeoLife dataset, a comprehensive collection of location-based data gathered from GPS-enabled devices. This dataset spans from April 2007 to April 2012, amassed through the voluntary participation of users who carried GPS-enabled devices, such as smartphones or dedicated GPS trackers. These devices continuously recorded the users' location coordinates, capturing their movements and activities throughout the day. Besides being publically available, a huge benefit of this data source is that data points are more evenly distributed in time (as GPS-deviced were pinged regularly) giving a more accurate estimation of individuals' route choices. Our experiments specifically used the subset of all GeoLife users who had mode labels available. We discarded training sets with less than 100 points (due to numerical instabilities in matrix operations) and more than 2000 points (due to time constraints). We only considered trips made via buses, bikes, and by walking, as the other modes had too few trips to attain meaningful results.

We implemented our methodology in the Python programming language (van Rossum, 1995). We used GPyTorch (Gardner et al., 2018) to increase efficiency and speed up matrix inversions within our GP framework. Furthermore, we used pandas (Wes McKinney, 2010) to read and wrangle the mobile dataset from which we derived the GP models, NumPy (Harris et al., 2020) for a variety of array operations, and scikit-mobility (Pappalardo et al., 2019) to conduct data cleaning and preprocessing. Last but not least, we used matplotlib (Hunter, 2007) to produce all visualizations of our results.

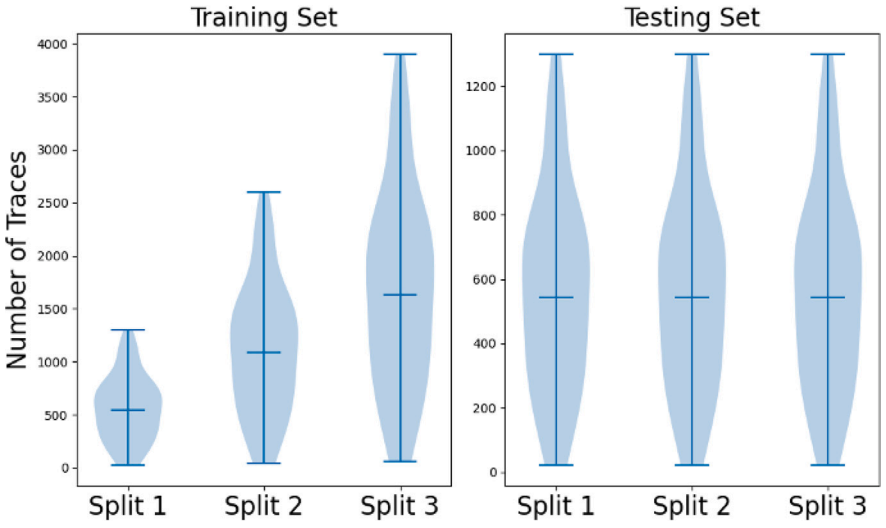


Fig. 8. Violin plots of training and testing set sizes across our passively-collected mobile data experiments.

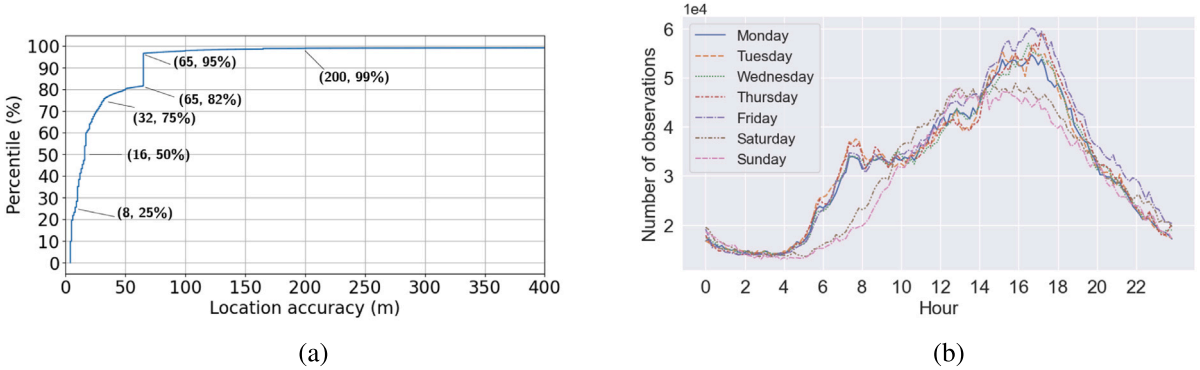


Fig. 9. Descriptive analysis of Spectus data for all 2000 users. (a) Cumulative distribution of location accuracy. (b) Distribution of observations within each day of the week.

Descriptive statistics

Most data points exhibit a high level of precision, with approximately 95% of all traces located within a 65-m radius of their true position (Fig. 9(a)). This represents a significant improvement compared to a previous study conducted in 2018 using a similar dataset, where only around 7% of observations had a precision worse than 1000 m (Ban et al., 2018). Regarding the temporal distribution of observations throughout the day (Fig. 9(b)), our findings align with that of Ban et al. (2018). We observe two distinct peaks on weekdays: one in the morning, around 7 am, and another in the evening, around 6 pm. On weekends, a single mid-day peak is evident. Moreover, the number of observations exhibits a positive trend as the hours progress, which holds true for every day.

We find trivial differences between the entire dataset and the 33 users sampled for the experiments in Section 4.3, confirmed by looking at various metrics. Fig. 10 highlights the distribution of location accuracies between the two sets, showing that the experiments sample does not contain any outliers. Fig. 11 shows the distribution of observation counts by time of day (normalized by the total count of observations) between the two datasets, where we observe slight differences in the weekday distributions. Finally, Fig. 12 shows the distribution of sampling rates over time in the dataset. We observe slightly higher intervals between consecutive observations in certain weeks, but overall the two distribution look fairly similar.

Data preprocessing and modeling considerations

We preprocess raw GPS traces to remove noisy data in two ways. First, we filter out rows using an upper bound on velocity (200 km/h, as the crow flies), which removes oscillations and physically-impossible jumps that may be present in raw mobile data. We further remove observations with a precision radius greater than 300 m. The potential of these noisy points to provide

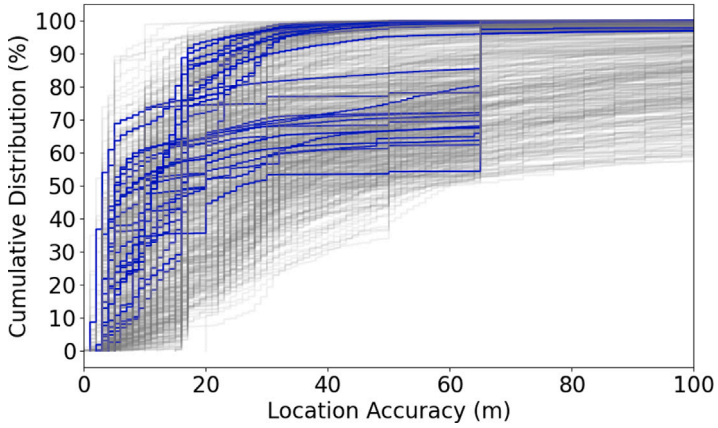


Fig. 10. Cumulative distribution of location accuracy for Spectus users in the experiments sample (blue) and the whole dataset (gray). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

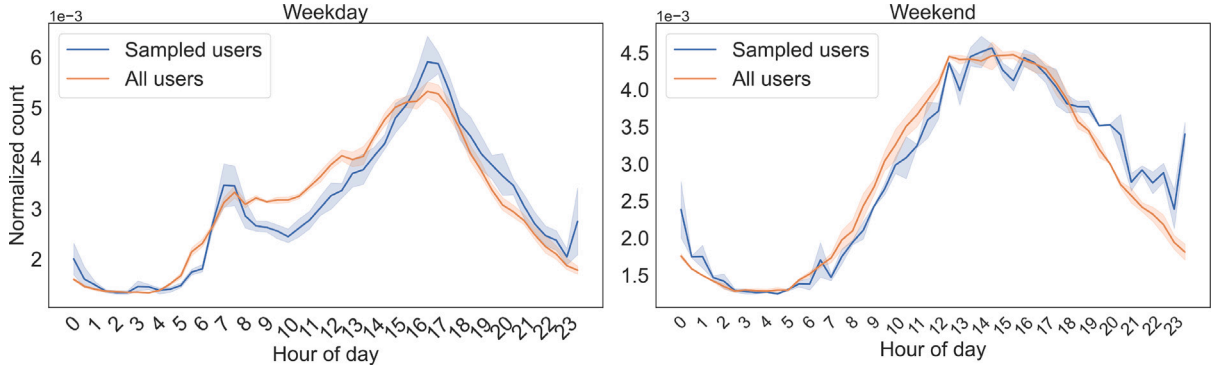


Fig. 11. Normalized observation counts by time of day for Spectus users in the experiments sample (blue) and the whole dataset (orange). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

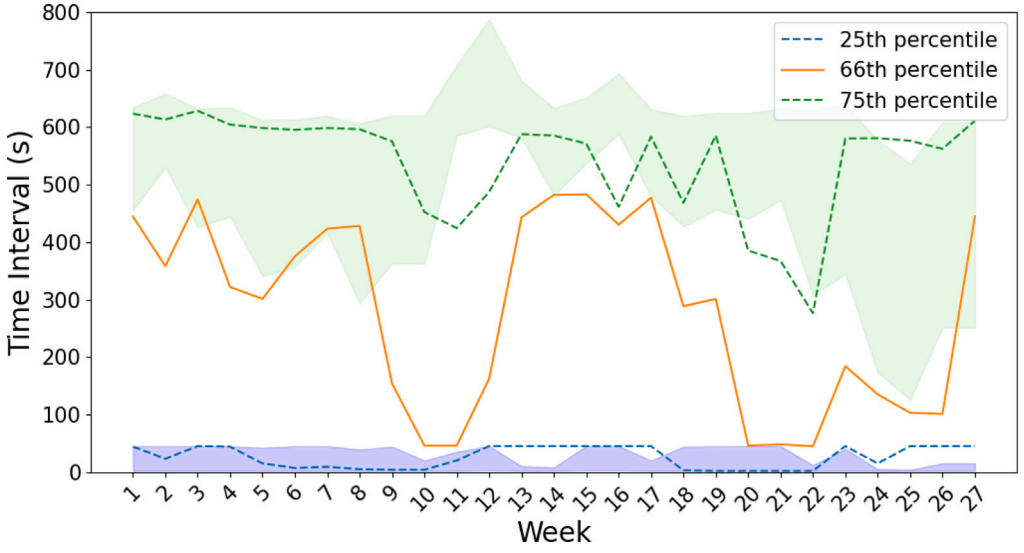


Fig. 12. Evolution of time interval between two consecutive points Week 1 to 27. Lines show the percentiles for the experiment sample, while the green and blue fillings shows the 70–80 and 15–35 percentile range for the entire dataset, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 4
Temporal and physical dimensions used in our experiments.

Variable	Notation	Type	Model inputs
Unix time (normalized)	$\mathbf{t}_u$	Continuous	$[0, 1, \ldots, n]$
Seconds (after midnight) - Sine	$\mathbf{t}_{ss}$	Continuous	$[0, \ldots, 1]$
Seconds (after midnight) - Cosine	$\mathbf{t}_{sc}$	Continuous	$[0, \ldots, 1]$
Day of week	$\mathbf{t}_d$	Categorical (Binary)	[0, 1]
AM peak[a]	$\mathbf{t}_{am}$	Binary	[0, 1]
PM peak[a]	$\mathbf{t}_{pm}$	Binary	[0, 1]
Average segment velocity	$\mathbf{v}$	Continuous	$[0, \infty]$
Bearing	$\boldsymbol{\beta}$	Continuous	[0, 360]

[a] Denotes a dimension that was not employed in the GeoLife experiments.

valuable location information is shadowed by the problems they cause for model calibration, which disrupts the continuity of smooth trajectories.

We also reduce the size of trajectories through a compression algorithm (Algorithm 2) that aims to generate an approximated trajectory that largely retains the shape of the original trajectory. If two points are within a certain Haversine distance (we use 100 m), we replace the two entries with their median—this is a variation of the Douglas-Peucker algorithm (Ugurel et al., 2024). We do this to ease the computational burdens of training our GP framework, which can be burdensome for large trajectory sets. Because many GPS points are close in time and space, this allows us to significantly reduce the size of our training sample without compromising accuracy.

A practical consideration with using a greedy strategy to learn the form of the best-fitting kernel is that of computational complexity. For this, we suggest leveraging "the median trick" (Schölkopf and Smola, 2002)—instead of minimizing the negative marginal log likelihood to learn the parameters of the kernel, they can be manually set using apriori knowledge of the domain and the data. For example, the lengthscale parameter can be set as the median gap between observations in the input space. This idea gets tricky with parameters that do not relate to the smoothness of the function (i.e., the outputscale or the period length), but these too can be experimented with in a systematic way to reduce the computational burden.

Predictor variables

Table 4 shows the temporal and physical variables we use as predictors in our experiments. As discussed in Section 3.4, we leverage various calendar-based structures to encode more regularity into the temporal GP. We normalize Unix time by equating a user's first observation to 0 and increasing it linearly thereafter. The second of the day is converted to cyclical encoding by calculating the sine and cosine values, wrapping smoothly around the boundaries (e.g., 23:59:59 and 00:00:00). Monday is the first day of the week and it is denoted with a 0. The AM peak variable is active in any observation between 6:00 AM and 10:00 AM, while the PM peak is between 3:00 PM and 7:00 PM (both in local time). The average segment velocity and bearing are calculated according to Eqs. (7) and (8), respectively.

Algorithms

Algorithms 1, 2, 3, and 4 are used inside Algorithm 5 (which is for the GeoLife experiments), while Algorithm 2 is also employed in the rest of the experiments.

Algorithm 1: Filter Trips

Input: Points
Output: Filtered points
Function *FilterTrips(points)*:
    Filter points outside Beijing based on longitude and latitude constraints;
    Remove trips going outside the boundary;
    Filter points if segment velocity between two consecutive points is greater than 300 km/h;
    return *Filtered points*;

Algorithm 2: Trajectory Data Compression

Input: Individual's trajectory data; 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**
  for $j < LenTraj + 1$ **do**
    $d_{ij} \leftarrow$ Measure Haversine distance between (ϕ_i, λ_i) and (ϕ_j, λ_j);
    if $d_{ij} > r$ **then**
      break // End current for loop;
    end
    Add ϕ_j and λ_j to lists $[\phi]$ and $[\lambda]$;
    $j \leftarrow j + 1$;
  end
  $(\phi, \lambda) \leftarrow$ 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 the next iteration;
end
return *Output* // The compressed trajectory data;

Algorithm 3: Elbow Method

Input: Data, max_clusters, plot
Output: Optimal number of clusters
Function *FindOptimalTripClusters(data, max_clusters)*:
  Initialize empty lists to store the inertia values and number of clusters;
  inertia_values ← [];
  num_clusters ← [];
  for *k* **to** *2* **to** *max_clusters + 1* **do**
    Apply K-means clustering;
    Compute inertia and number of clusters;
    Append inertia value and number of clusters to the lists;
  end
  Calculate differences in inertia values;
  Calculate differences ratio;
  Find the index of the elbow point;
  return *Optimal number of clusters*;

Algorithm 4: Train-Test Split

Input: DataFrame, k, random_state
Output: List of K folds
Function *TripLabelBasedKFoldSplit(df, k, random_state)*:
  Get unique trip IDs from the dataframe;
  Initialize KFold;
  List to hold the training and testing dataframes for each fold;
  Perform the k-fold split on the trip IDs;
  return *List of K folds*;

Algorithm 5: Trip Sampling

Input: Set of trips $Z_{w,e}$ for user w and mode e
Output: Three folds for training and testing
Function *TripSampling*($Z_{w,e}$)**:**
  FilteredTrips ← `FilterTrips(`$Z_{w,e}$, 300 *km/h*`)`;
  SimplifiedTrips ← `TrajectoryDataCompression(`*FilteredTrips,* 0.2 *km radius*`)`;
  if |*SimplifiedTrips*| < 10 **then**
    Continue to next set of trips;
  end
  else if 10 ≤ |*SimplifiedTrips*| ≤ 20 **then**
    End;
  end
  else
    k ← `ElbowMethod(KMeansClustering(`*SimplifiedTrips*`))`;
    Clusters ← `KMeansClustering(`*k, SimplifiedTrips*`)`;
    Sort Clusters by size in descending order;
    TopTwoClusters ← First two clusters in Clusters;
    if |*SimplifiedTrips*| > 20 **then**
      Shuffle SimplifiedTrips;
      SampledTrips ← First 20 trips in SimplifiedTrips;
    end
    else if |*SimplifiedTrips*| < 10 **then**
      MoreClusters ← `KMeansClustering(`*SimplifiedTrips,* $k+1$`)`;
      SimplifiedTrips;
      `// Recursion`
    end
    else
      End;
    end
  end
  `TrainTestSplit(`*SimplifiedTrips*`)`;
  `// Separating into three folds for training and testing`

Appendix C. Handling nonstationarity with GPs

Human mobility has various spatial and temporal scales (Alessandretti et al., 2020). Specifically, as a person moves, time determines their location at the level of minutes and seconds, while when that person is stationary at an activity location, time impacts their travel behavior at the level of hours. For modeling purposes, this implies that in a traditional time-series GP with a single temporal input, the lengthscale parameter of the GP will not be able to capture the impact of time evenly between the two states (moving and staying).

There are multiple ways to overcome this problem. The most common, which is the one we rely on in this paper, is to represent time using various calendar-based structures (e.g., seconds of the day, days of the week). This allows the GP to learn unique lengthscales across more degrees of freedom, providing the adaptive local smoothing necessary for non-stationary data. For example, while Unix time may have a small lengthscale (capturing second-level variations), other variables capturing the time of the day may have a lengthscale on the order of hours.

GPs are naturally well-suited to handle statistical nonstationarity as well. Duvenaud (2014) details various approaches to combining different kernels to attain nonstationary processes without having to detrend data. This tends to require usage of nonstationary kernels, like the Linear kernel (Eq. (27)), which produces different predictions if the data were moved while the kernel parameters were kept fixed.

$$k_{LIN} = \mathbf{x}^{\top}\mathbf{x}' \tag{27}$$

To showcase GP's flexibility in handling nonstationarity, we have included an example of modeling the Mauna Loa CO2 concentration dataset using a GP with feature engineering (and without detrending). This dataset exhibits an increasing mean in the observed values over years. The usual approach to modeling this type of data using linear models is to first detrend the data (i.e., as shown

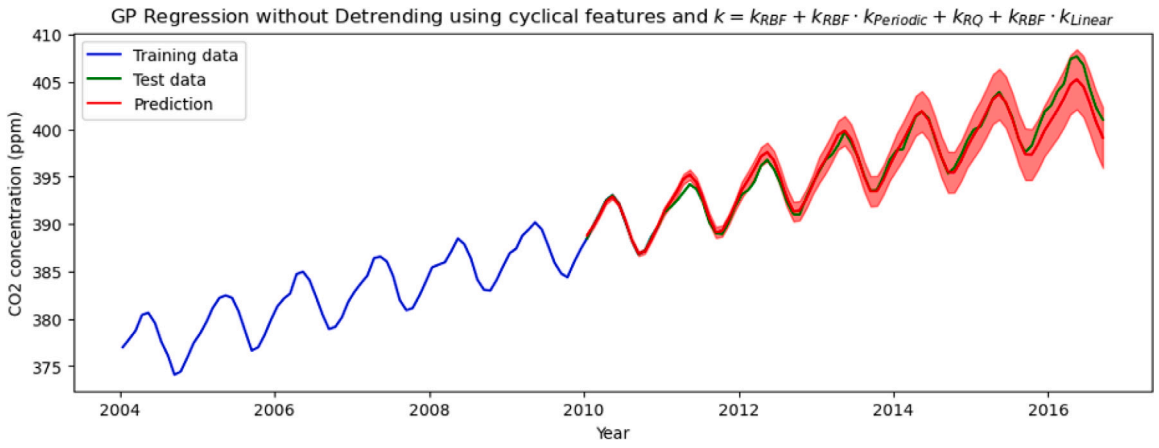


Fig. 13. GP regression using cyclical features and a composite kernel accurately captures nonstationary behavior in CO2 concentration of the Mauna Loa volcano.

here). However, we show that by creating cyclical features like months and years and leveraging the right kernel structure, this type of data can be modeled accurately using GPs (as seen in Fig. 13).

References

Alessandretti, L., Aslak, U., Lehmann, S., 2020. The scales of human mobility. Nature 587 (7834), 402–407.

Álvarez, M., Luengo, D., Lawrence, N.D., 2009. Latent force models. In: van Dyk, D., Welling, M. (Eds.), Proceedings of the Twelth International Conference on Artificial Intelligence and Statistics. In: Proceedings of Machine Learning Research, vol. 5, PMLR, Hilton Clearwater Beach Resort, Clearwater Beach, Florida USA, pp. 9–16, URL https://proceedings.mlr.press/v5/alvarez09a.html.

Alvarez, M.A., Luengo, D., Lawrence, N.D., 2013. Linear latent force models using Gaussian processes. IEEE Trans. Pattern Anal. Mach. Intell. 35 (11), 2693–2705.

Apple, 2021. Apple. URL https://www.apple.com/newsroom/2021/06/apple-advances-its-privacy-leadership-with-ios-15-ipados-15-macos-monterey-and-watchos-8/.

Ban, X.J., Chen, C., Wang, F., Wang, J., Zhang, Y., et al., 2018. Promises of Data from Emerging Technologies for Transportation Applications: Puget Sound Region Case Study. Technical report, United States. Federal Highway Administration.

Barla, A., Odone, F., Verri, A., 2003. Histogram intersection kernel for image classification. In: Proceedings 2003 International Conference on Image Processing (Cat. No. 03CH37429). Vol. 3, IEEE, pp. III–513.

Batista, S.F., Cantelmo, G., Menéndez, M., Antoniou, C., 2022. A Gaussian sampling heuristic estimation model for developing synthetic trip sets. Comput.-Aided Civ. Infrastruct. Eng. 37 (1), 93–109.

Bayarma, A., Kitamura, R., Susilo, Y.O., 2007. Recurrence of daily travel patterns: stochastic process approach to multiday travel behavior. Transp. Res. Rec. 2021 (1), 55–63.

Bhat, C.R., Koppelman, F.S., 1999. Activity-based modeling of travel demand. In: Hall, R.W. (Ed.), Handbook of Transportation Science. Springer US, Boston, MA, pp. 35–61.

Bonilla, E.V., Chai, K., Williams, C., 2007. Multi-task Gaussian process prediction. Adv. Neural Inf. Process. Syst. 20.

Chen, Y., Hosseini, B., Owhadi, H., Stuart, A.M., 2021. Solving and learning nonlinear PDEs with Gaussian processes. J. Comput. Phys. 447, 110668.

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. C 68, 285–299.

Cuomo, S., Di Cola, V.S., Giampaolo, F., Rozza, G., Raissi, M., Piccialli, F., 2022. Scientific machine learning through physics–informed neural networks: Where we are and what's next. J. Sci. Comput. 92 (3), 88.

Deng, T., Zhang, K., Shen, Z.-J.M., 2021. A systematic review of a digital twin city: A new pattern of urban governance toward smart cities. J. Manage. Sci. Eng. 6 (2), 125–134.

Duvenaud, D., 2014. Automatic Model Construction with Gaussian Processes (Ph.D. thesis). University of Cambridge.

Gardner, J., Pleiss, G., Weinberger, K.Q., Bindel, D., Wilson, A.G., 2018. Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration. Adv. Neural Inf. Process. Syst. 31.

Gibbs, M.N., 1998. Bayesian Gaussian Processes for Regression and Classification (Ph.D. thesis). Citeseer.

Gönen, M., Alpaydın, E., 2011. Multiple kernel learning algorithms. J. Mach. Learn. Res. 12, 2211–2268.

Gonzalez, M.C., Hidalgo, C.A., Barabasi, A.-L., 2008. Understanding individual human mobility patterns. Nature 453 (7196), 779–782.

Hanson, S., Huff, J.O., 1988. Repetition and day-to-day variability in individual travel patterns: Implications for classification. Behav. Model. Geogr. Plan. Croom Helm Lond. 368–398.

Harris, C.R., Millman, K.J., van der Walt, S.J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N.J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M.H., Brett, M., Haldane, A., del Río, J.F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., Oliphant, T.E., 2020. Array programming with NumPy. Nature 585 (7825), 357–362.

Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural Comput. 9 (8), 1735–1780.

Hunter, J.D., 2007. Matplotlib: A 2D graphics environment. Comput. Sci. Eng. 9 (3), 90–95.

Jiang, L., Chen, C.-X., Chen, C., 2023. L2mm: learning to map matching with deep models for low-quality gps trajectory data. ACM Trans. Knowl. Discov. Data 17 (3), 1–25.

Kim, E.-K., MacEachren, A., 2014. An index for characterizing spatial bursts of movements: A case study with geo-located Twitter data. In: GIScience 2014 Workshop on Analysis of Movement Data. Citeseer.

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.
Kingma, D.P., Ba, J., 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.
Kitamura, R., Van Der Hoorn, T., 1987. Regularity and irreversibility of weekly travel behavior. Transportation 14, 227–251.
Lapidus, L., Pinder, G.F., 1999. Numerical Solution of Partial Differential Equations in Science and Engineering. John Wiley & Sons.
Lasserre, J.A., Bishop, C.M., Minka, T.P., 2006. Principled hybrids of generative and discriminative models. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR'06, 1, IEEE, pp. 87–94.
Lee, M.S., McNally, M.G., 2003. On the structure of weekly activity/travel patterns. Transp. Res. A 37 (10), 823–839.
Lee, M., McNally, M.G., 2006. An empirical investigation on the dynamic processes of activity scheduling and trip chaining. Transportation 33, 553–565.
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 (8), 1777–1784.
Liu, X., Qian, S., Teo, H.-H., Ma, W., 2024. Estimating and mitigating the congestion effect of curbside pick-ups and drop-offs: a causal inference approach. Transportation Science 58 (2), 355–376.
McGuckin, N., Murakami, E., 1999. Examining trip-chaining behavior: Comparison of travel by men and women. Transp. Res. Rec. 1693 (1), 79–85, arXiv:https://doi.org/10.3141/1693-12.
Mohamed, A., Zhu, D., Vu, W., Elhoseiny, M., Claudel, C., 2022. Social-implicit: Rethinking trajectory prediction evaluation and the effectiveness of implicit maximum likelihood estimation. In: European Conference on Computer Vision. Springer, pp. 463–479.
Morris, D.H., Rossine, F.W., Plotkin, J.B., Levin, S.A., 2021. Optimal, near-optimal, and robust epidemic control. Commun. Phys. 4 (1), 78.
Nevin, J.W., Vaquero-Caballero, F., Ives, D.J., Savory, S.J., 2021. Physics-informed Gaussian process regression for optical fiber communication systems. J. Lightwave Technol. 39 (21), 6833–6844.
Nishii, K., Kondo, K., Kitamura, R., 1988. An empirical analysis of trip chaining behavior.
Ong, C., Williamson, R.C., Smola, A., 2002. Hyperkernels. Adv. Neural Inf. Process. Syst. 15.
Pappalardo, L., Simini, F., Barlacchi, G., Pellungrini, R., 2019. Scikit-mobility: A python library for the analysis, generation and risk assessment of mobility data. arXiv preprint arXiv:1907.07062.
Pendyala, R.M., Kitamura, R., Chen, C., Pas, E.I., 1997. An activity-based microsimulation analysis of transportation control measures. Transp. Policy 4 (3), 183–192, URL https://www.sciencedirect.com/science/article/pii/S0967070X9700005X.
Raissi, M., Perdikaris, P., Karniadakis, G.E., 2019. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J. Comput. Phys. 378, 686–707.
Rasmussen, C., Ghahramani, Z., 2000. Occam' s Razor. In: Advances in Neural Information Processing Systems. Vol. 13, MIT Press, URL https://papers.nips.cc/paper/2000/hash/0950ca92a4dcf426067cfd2246bb5ff3-Abstract.html.
Rasmussen, C.E., Williams, C.K.I., 2006. Gaussian Processes for Machine Learning. Adaptive computation and machine learning, MIT Press, Cambridge, Mass, OCLC: ocm61285753.
van Rossum, G., 1995. Python tutorial. Technical Report CS-R9526, Centrum voor Wiskunde en Informatica (CWI), Amsterdam.
Schölkopf, B., Smola, A.J., 2002. Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press.
Schwarz, G., 1978. Estimating the dimension of a model. Ann. Stat. 461–464.
Shin, Y., Darbon, J., Karniadakis, G.E., 2020. On the convergence of physics informed neural networks for linear second-order elliptic and parabolic type PDEs. arXiv preprint arXiv:2004.01806.
Snelson, E., Ghahramani, Z., 2005. Sparse Gaussian processes using pseudo-inputs. Adv. Neural Inf. Process. Syst. 18.
Song, C., Qu, Z., Blumm, N., Barabási, A.-L., 2010. Limits of predictability in human mobility. Science 327 (5968), 1018–1021.
Timmermans, H., Arentze, T.A., 2011. Transport models and urban planning practice: Experiences with albatross. Transp. Rev. 31 (2), 199–207, arXiv:https://doi.org/10.1080/01441647.2010.518292.
Tobler, W.R., 1970. A computer movie simulating urban growth in the Detroit region. Econ. Geogr. 46 (sup1), 234–240.
Ugurel, E., 2023. Time-varying transition matrices with multi-task Gaussian processes. arXiv preprint arXiv:2306.11772.
Ugurel, E., Guan, X., Wang, Y., Huang, S., Wang, Q.R., Chen, C., 2024. Correcting missingness in passively-generated mobile data with multi-task Gaussian processes. Transp. Res. C 161.
Wallace, B., Barnes, J., Rutherford, G.S., 2000. Evaluating the effects of traveler and trip characteristics on trip chaining, with implications for transportation demand management strategies. Transp. Res. Rec. 1718 (1), 97–106, arXiv:https://doi.org/10.3141/1718-13.
Wang, S., Wang, H., Perdikaris, P., 2021b. On the eigenvector bias of Fourier feature networks: From regression to solving multi-scale PDEs with physics-informed neural networks. Comput. Methods Appl. Mech. Engrg. 384, 113938.
Wang, Z., Xing, W., Kirby, R., Zhe, S., 2022b. Physics informed deep kernel learning. In: International Conference on Artificial Intelligence and Statistics. PMLR, pp. 1206–1218.
Wang, S., Yu, X., Perdikaris, P., 2022a. When and why PINNs fail to train: A neural tangent kernel perspective. J. Comput. Phys. 449, 110768.
Wang, Z., Zhang, S., James, J., 2020. Reconstruction of missing trajectory data: a deep learning approach. In: 2020 IEEE 23rd International Conference on Intelligent Transportation Systems. ITSC, IEEE, pp. 1–6.
Wes McKinney, 2010. Data structures for statistical computing in python. In: Stéfan van der Walt, Jarrod Millman (Eds.), Proceedings of the 9th Python in Science Conference. pp. 56–61.
Wilson, A.G., 2014. Covariance Kernels for Fast Automatic Pattern Discovery and Extrapolation with Gaussian Processes (Ph.D. thesis). University of Cambridge Cambridge, UK.
Wu, F., Cheng, Z., Chen, H., Qiu, Z., Sun, L., 2024. Traffic state estimation from vehicle trajectories with anisotropic Gaussian processes. Transp. Res. C 163, 104646.
Yang, X., Barajas-Solano, D., Tartakovsky, G., Tartakovsky, A.M., 2019. Physics-informed CoKriging: A Gaussian-process-regression-based multifidelity method for data-model convergence. J. Comput. Phys. 395, 410–431.
Yuan, Y., Raubal, M., 2012. Extracting dynamic urban mobility patterns from mobile phone data. In: Geographic Information Science: 7th International Conference, GIScience 2012, Columbus, OH, USA, September 18-21, 2012. Proceedings 7. Springer, pp. 354–367.
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. B 146, 88–110.
Zheng, Y., Li, Q., Chen, Y., Xie, X., Ma, W.-Y., 2008. Understanding mobility based on GPS data. In: Proceedings of the 10th International Conference on Ubiquitous Computing. pp. 312–321.
Zheng, Y., Xie, X., Ma, W.-Y., et al., 2010. GeoLife: A collaborative social networking service among user, location and trajectory. IEEE Data Eng. Bull. 33 (2), 32–39.
Zheng, Y., Zhang, L., Xie, X., Ma, W.-Y., 2009. Mining interesting locations and travel sequences from gps trajectories. In: Proceedings of the 18th International Conference on World Wide Web. pp. 791–800.
Zhu, Z., Xu, M., Di, Y., Yang, H., 2022. Fitting spatial-temporal data via a physics regularized multi-output grid Gaussian process: case studies of a bike-sharing system. IEEE Trans. Intell. Transp. Syst. 23 (11), 21090–21101.