Capturing Uncertainty in Unsupervised GPS Trajectory Segmentation Using Bayesian Deep Learning

Christos Markos,[1,2] **James J.Q. Yu,**[1] **Richard Yi Da Xu**[2]
[1]Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen, China
[2]Faculty of Engineering and Information Technology, University of Technology Sydney, Australia
11760007@mail.sustech.edu.cn, yujq3@sustech.edu.cn, yida.xu@uts.edu.au

Abstract

Intelligent transportation management requires not only statistical information on users' mobility patterns, but also knowledge of their corresponding transportation modes. While GPS trajectories can be readily obtained from GPS sensors found in modern smartphones and vehicles, these massive geospatial data are neither automatically annotated nor segmented by transportation mode, subsequently complicating transportation mode identification. In addition, predictive uncertainty caused by the learned model parameters or variable noise in GPS sensor readings typically remains unaccounted for. To jointly address the above issues, we propose a Bayesian deep learning framework for unsupervised GPS trajectory segmentation. After unlabeled GPS trajectories are preprocessed into sequences of motion features, they are used in unsupervised training of a channel-calibrated temporal convolutional neural network for timestep-level transportation mode identification. At test time, we approximate variational inference via Monte Carlo dropout sampling, leveraging the mean and variance of the predicted distributions to classify each input timestep and estimate its predictive uncertainty, respectively. The proposed approach outperforms both its non-Bayesian variant and established GPS trajectory segmentation baselines on Microsoft's Geolife dataset without using any labels.

Introduction

Citywide knowledge of users' mobility patterns and associated transportation modes is crucial for intelligent transportation management and infrastructure design. How to infer transportation modes from *segments* of Global Positioning System (GPS) trajectories has been studied extensively in recent years (Zheng et al. 2008b; Stenneth et al. 2011; Xiao, Juan, and Zhang 2015; Zhang et al. 2019; Dabiri et al. 2020; Yu 2020). Yet the real-world applicability of most such methods is limited by their simplifying assumption that single-transportation-mode segments would somehow be available to begin with. The challenging problem of *trajectory segmentation* (Fig. 1), or how to split GPS trajectories into segments involving exactly one transportation mode, is often avoided by leveraging the transportation mode labels available during data preprocessing. In reality, however, the tremendous amounts of GPS data that are generated daily are not automatically segmented or even annotated by transportation mode.

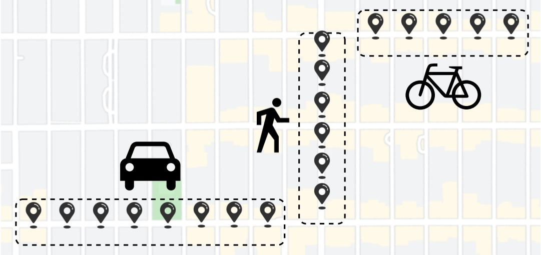


Figure 1: GPS trajectory segmentation aims to extract single-transportation-mode segments from sequences of GPS points.

Following a naive approach to trajectory segmentation, one could simply extract consecutive nonoverlapping trajectory segments of uniform length (Dabiri and Heaslip 2018). In doing so, however, there could be multiple transportation modes in many of the obtained segments. Even by mapping these segments to their dominant transportation mode, one would be introducing unnecessary noise to the data, potentially impacting the performance of transportation mode classifiers. In this direction, the literature has sought to detect transportation mode change points within GPS trajectories by leveraging sliding window approaches (Stenneth et al. 2011; Bolbol et al. 2012), mobility-based heuristics (Zheng et al. 2008b; Xiao, Juan, and Zhang 2015), and more recently, optimization-based algorithms (Dabiri et al. 2020). However, such approaches often exhibit poor scalability, require extensive feature engineering or transportation domain knowledge, and assume independent and identically distributed samples, respectively.

In this work, we argue that a truly comprehensive approach to GPS trajectory segmentation would need to address the following issues tied to the transportation domain:

1. **Semantic trajectory segmentation.** Given the limitations of existing methods, as mentioned above, we consider deep semantic segmentation as a promising alternative. Commonly encountered in computer vision applications (Long, Shelhamer, and Darrell 2015; Chen et

al. 2018; Fu et al. 2019), semantic segmentation methods train deep Convolutional Neural Networks (CNNs) to detect refined object boundaries within images or videos by performing pixel- rather than image-level classification.

2. **Learning from unlabeled GPS data.** Due to privacy concerns or simply lack of motivation, users cannot be expected to label their voluminous GPS data consistently and accurately. Given that traditional clustering algorithms often produce suboptimal results when applied to high-dimensional data (Li, Qiao, and Zhang 2018), we turn to the emerging research area of deep clustering, which trains deep neural networks to cluster unlabeled data via unsupervised learning (Xie, Girshick, and Farhadi 2016; Caron et al. 2018; Ji, Henriques, and Vedaldi 2019).
3. **Capturing predictive uncertainty.** While standard deep learning classifiers produce discrete probability distributions via the softmax function, this does not sufficiently capture model uncertainty (Kendall and Gal 2017). In GPS-based applications, we posit that uncertainty could also be tied to inconsistent sensor sampling rates among different observations, as well as erroneous or missing measurements due to signal loss. Bayesian deep learning can be viewed as a probabilistic extension to standard deep learning that enables quantification of both model and input uncertainty (Ribeiro et al. 2019).

Drawing inspiration from recent developments in semantic segmentation, deep clustering, and Bayesian deep learning, this work proposes to reframe GPS trajectory segmentation as timestep-level transportation mode identification. As such, a channel-calibrated Bayesian Temporal Convolutional Network (BTCN) is introduced for unsupervised, uncertainty-aware GPS trajectory segmentation. BTCN extends standard TCNs, recently proposed as a sequence modeling alternative to recurrent neural networks (Bai, Kolter, and Koltun 2018), with (1) Squeeze-and-Excitation (SE) blocks (Hu, Shen, and Sun 2018) to encourage learning interdependencies between channels, and (2) Monte Carlo (MC) dropout sampling as an approximation of variational inference (Kendall and Gal 2017) to not only capture predictive uncertainty but also use it to refine predictions. In our experiments on Microsoft's Geolife dataset (Zheng et al. 2008b; 2008a), BTCN achieved 65.8% timestep-level accuracy without using any labels, outperforming established baselines as well as its non-Bayesian variant.

Related Work

Semantic Segmentation In this study, we tackle trajectory segmentation from a novel perspective, drawing inspiration from recent deep semantic segmentation methods successfully applied to computer vision applications.

Semantic image segmentation classifies each pixel in an input image, allowing for fine-grained detection of object boundaries. Earlier work (Chen et al. 2014) repurposes contracting image-level CNN classifiers for semantic segmentation via bilinear upsampling and fully connected conditional random fields, while (Long, Shelhamer, and Darrell 2015) converts them to Fully Convolutional Networks (FCNs) by replacing their fully connected layers with convolutions, introducing in-network upsampling, and adding skip connections from shallow to deep layers.

Since then, FCNs have served as the foundation for several semantic segmentation frameworks. To improve the recovery of downsampled spatial information, architectural adaptations include adding a symmetric decoder (Badrinarayanan, Kendall, and Cipolla 2017) with encoder-decoder skip connections (Ronneberger, Fischer, and Brox 2015) or dilated convolutions at the encoder side (Chen et al. 2018). Some works reduce the amount of downsampling operations in FCNs by developing context aggregation modules, often including dilated convolutions in cascades (Yu and Koltun 2016) or in parallel (Chen et al. 2017; Fu et al. 2019). Instead, (Sun et al. 2019) avoids the need to recover downsampled spatial information by maintaining high-resolution latent representations throughout the encoding process.

Deep Clustering Combining unsupervised deep learning with traditional or custom clustering methods, deep clustering is viewed as a promising technique towards labeled data independence for the proposed trajectory segmentation framework.

Some approaches optimize image-to-image distance in a lower-dimensional space typically learned via unsupervised autoencoder pretraining, by applying K-means (Yang, Parikh, and Batra 2016) or attaching custom clustering layers (Xie, Girshick, and Farhadi 2016; Li, Qiao, and Zhang 2018) and minimizing a clustering loss. (Caron et al. 2018) instead leverages CNNs pretrained on different datasets, iteratively clustering their output using K-means and setting the resulting cluster assignments as pseudo-ground-truth labels to update the model parameters. It is also possible to leverage denoising (Ghasedi Dizaji et al. 2017) or variational (Jiang et al. 2017) autoencoders, as well as GANs (Mukherjee et al. 2019). However, while only optimizing a clustering loss offers simplicity of implementation, it also risks irreversibly corrupting the learned representations (Ji, Henriques, and Vedaldi 2019).

Another body of research instead clusters data based on mutual information between pairs of samples typically generated via data augmentation, or between samples and their latent representations. For instance, (Hu et al. 2017) trains neural networks to produce similar outputs for input samples and their augmented versions, while also maximizing the mutual information between the original samples and their learned representations. (Hjelm et al. 2018) instead maximizes the information between latent representations and local regions of the input samples, while also attempting to match these representations to a prior distribution. Finally, (Ji, Henriques, and Vedaldi 2019) maximizes the mutual information between a neural network's outputs for pairs of samples, additionally improving performance by simultaneously training an auxiliary overclustering component.

Preliminaries

This section clarifies our working definitions of key transportation-related terms, before formalizing the problem to be addressed in this work and finally introducing Bayesian deep learning.

Segmenting GPS Trajectories by Transportation Mode

Definition 1 (GPS Point) A GPS point is denoted by $p_j = \langle t_j, \mathrm{lat}_j, \mathrm{lon}_j \rangle \in \mathbb{R}^3$, where t_j measures the number of days since a reference date, and $-90 \leq \mathrm{lat}_j \leq 90$, $-180 \leq \mathrm{lon}_j \leq 180$ are latitude and longitude coordinates given in decimal degrees.

Definition 2 (GPS Trajectory) A GPS trajectory T_i comprises a sequence of GPS points. Since the backbone of this work is a deep convolutional neural network operating on fixed-size inputs, we assume T_i to be a sequence $T_i = \{p_j\}_1^M$ truncated to fixed length M.

Definition 3 (Motion Feature Sequence) Directly training a deep learning model on raw GPS data could be problematic. Intuitively, it might generalize poorly whenever users visit locations inadequately represented in the training set. Therefore, following established transportation mode identification literature (Zheng et al. 2008b; Dabiri et al. 2020; Yu 2020), we preprocess each T_i into a sequence of motion features $F_i = \{f_j\}_1^M$ by transforming each $p_j \in T_i$ into a vector $f_j \in \mathbb{R}^N$ of N motion features.

Problem (Trajectory Segmentation) Given a GPS trajectory $T_i \in \mathbb{R}^{M\times 3}$ preprocessed into a sequence of motion features $F_i \in \mathbb{R}^{M\times N}$ and K target transportation mode classes, predict the transportation modes $Y_i \in \{0, \ldots, K-1\}^M$ for each timestep m, where $0 < m \leq M$.

Bayesian Deep Learning

Let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$ be a dataset of observations X and targets Y. For a standard fully connected neural network with L stacked hidden layers and parameters $\theta = \{(W^l, b^l)\}_{l=0}^{L+1}$, the ReLU-activated output of the l-th hidden layer can be written as:

$$h^l = \mathrm{ReLU}(W^l \cdot h^{l-1} + b^l). \tag{1}$$

For K-class classification, the softmax activation function is typically applied to the neural network's output logits. The model likelihood is then given by:

$$p(y = j|x, \theta) = \frac{\exp(W_j^{L+1} \cdot h^L + b^{L+1})}{\sum_{k\in K} \exp(W_k^{L+1} \cdot h^L + b^{L+1})}. \tag{2}$$

In contrast, Bayesian neural networks place a prior distribution $p(\theta)$ on their weights and biases, resulting in the posterior distribution:

$$p(\theta|\mathcal{D}) = \frac{p(\mathcal{D}|\theta)p(\theta)}{p(\mathcal{D})} = \frac{\prod_{i=1}^n p(y_i|x_i, \theta)p(\theta)}{p(\mathcal{D})}, \tag{3}$$

and the following predictive distribution for new inputs x', y':

$$p(y'|x', \mathcal{D}) = \int_\Theta p(y'|x', \theta)p(\theta, \mathcal{D})\, d\theta. \tag{4}$$

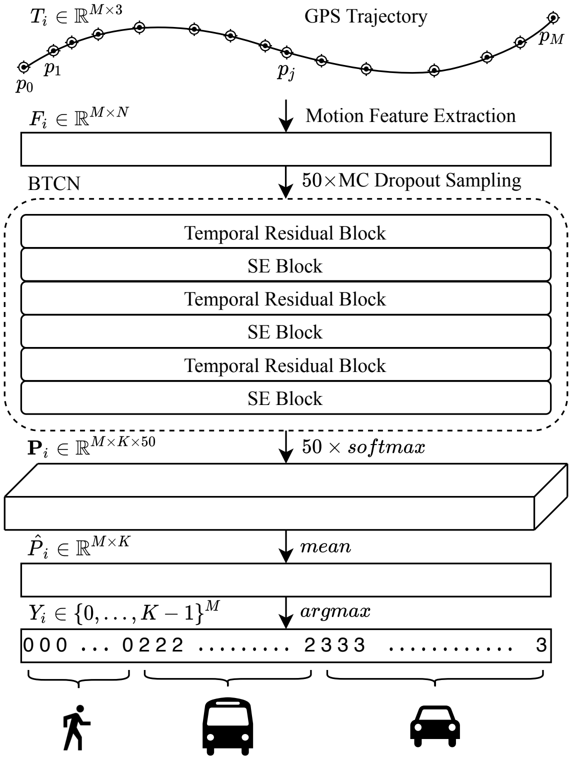


Figure 2: Overview of the proposed trajectory segmentation framework. At test time, GPS trajectories are preprocessed into sequences of motion features and repeatedly fed to the proposed BTCN while dropout remains activated. The mean of these aggregated softmax probabilities is taken as the final predictions, while their variance is used to quantify predictive uncertainty.

However, analytical estimation of the posterior is typically intractable, as it requires integration with respect to Θ, i.e., the space of all parameters, for which a closed form often does not exist. One way to approximate the posterior is through variational inference, which minimizes the Kullback-Leibler divergence between the posterior $p(\theta|\mathcal{D})$ and a variational distribution $q_\omega(\theta)$ with parameters ω (Kwon et al. 2020). Another approach is to approximate variational inference itself by applying stochastic regularization techniques like dropout to non-Bayesian neural networks. As will be detailed in the following section, in this work we approximate variational inference via MC dropout sampling (Kendall and Gal 2017).

Proposed Framework

This section first introduces the Bayesian temporal convolutional network proposed for GPS trajectory segmentation. Next, it explains how we approximate variational inference and capture uncertainty to refine model predictions. Finally, it presents the objective function that is optimized to learn the unsupervised segmentation task.

A high-level architectural view of BTCN is given in Fig. 2. BTCN fuses TCNs with SE blocks (Hu, Shen, and Sun

2018) to enrich the learned temporal trajectory representations with knowledge of feature map interdependencies. At inference time, a sequence of motion features is extracted from each GPS trajectory and sampled from BTCN multiple times while dropout is activated. Variational inference is approximated by the mean of these aggregated softmax probabilities constituting the final predictions, while their variance is used to estimate aleatoric and epistemic uncertainties.

Bayesian Temporal Convolutional Network

Given GPS trajectories $\mathcal{T}$ converted into sequences of motion features $\mathcal{F}$, one could approach trajectory segmentation via standard sequence modeling practice, i.e., using recurrent neural networks. However, recurrent architectures can be cumbersome to train, owing to reduced parallelism, vanishing or exploding gradients, as well as high memory requirements (Bai, Kolter, and Koltun 2018). On the other hand, recent work has shown that TCNs can effectively address the above issues and even outperform recurrent architectures in established sequence modeling tasks (Bai, Kolter, and Koltun 2018).

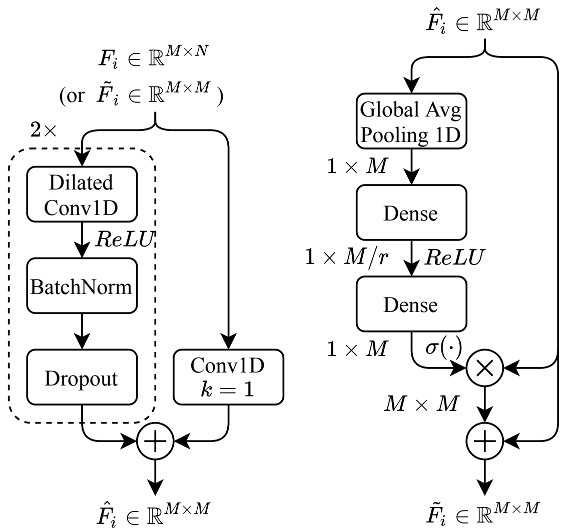
Architecture Similarly to TCNs, the basic component of BTCN is the temporal residual block. As shown in Fig. 3(a), the residual path first performs an optional 1D convolution with unit kernel size to ensure that the number of input features matches the desired number of output channels. Crucially, the latter is set equal to the length of the input sequence, allowing TCNs to produce sequences of the same length as their input. Next, the block applies two consecutive dilated 1D convolutions, each followed by a ReLU activation, a batch normalization layer, and finally a dropout layer. Contrary to standard convolutions, dilated ones apply filters over a larger area than their own by skipping input values with step d, otherwise known as dilation rate. As a result, dilation allows convolution layers to expand their receptive fields as the network depth grows, preserving input resolution without need for downsampling.

While stacking temporal residual blocks allows TCNs to uncover temporal dependencies among timesteps, we posit that it may not sufficiently account for dependencies within the channels themselves. Consequently, we follow recent work in this direction (Hu, Shen, and Sun 2018) and incorporate SE blocks after each temporal residual block. As shown in Fig. 3(b), the SE block first embeds channel information in $\hat{F}_i \in \mathbb{R}^{M\times N}$ across the temporal dimension using global average pooling. A vector $z \in \mathbb{R}^M$ is built for channels $c_i \in \mathbb{R}^M$, where:

$$z_i = \frac{1}{M}\sum_{m=1}^{M} u_i(m). \tag{5}$$

Then, the SE block scales down the resulting feature maps by a factor of $r \in \mathbb{Z}^+$, before rescaling them and applying a self-gating mechanism. This is achieved using two fully-connected (or dense) layers with weights $W_1 \in \mathbb{R}^{M/r}$ and $W_2 \in \mathbb{R}^M$; the operations can be formally described as follows:

$$\hat{z} = \sigma(W_1(\mathrm{ReLU}(W_2 z))), \tag{6}$$


$F_i \in \mathbb{R}^{M\times N}$
(or $\tilde{F}_i \in \mathbb{R}^{M\times M}$)
2×
Dilated Conv1D
ReLU
BatchNorm
Dropout
Conv1D $k=1$
$\hat{F}_i \in \mathbb{R}^{M\times M}$
$\hat{F}_i \in \mathbb{R}^{M\times M}$
Global Avg Pooling 1D
$1\times M$
Dense
$1\times M/r$
ReLU
Dense
$1\times M$
$\sigma(\cdot)$
$M\times M$
$\tilde{F}_i \in \mathbb{R}^{M\times M}$

(a) Temporal Residual Block (b) SE Block

Figure 3: The main components of the proposed BTCN. Temporal residual blocks leverage dilated 1D convolutions to capture high-resolution temporal information without need for downsampling. Always-on dropout layers are inserted to approximate variational inference via MC dropout sampling. The feature maps produced by each temporal residual block are subsequently recalibrated via an SE block.

where $\sigma(\cdot)$ denotes the sigmoid activation function. The gated output $\hat{z}$ is finally multiplied with the block's input to construct the final output $\tilde{F}_i \in \mathbb{R}^{M\times N}$, effectively promoting the most useful channels.

Dropout Variational Inference As mentioned earlier, BTCN approximates variational inference via dropout. This corresponds to injecting each deterministic $W_l \in \theta$ with noise following a Bernoulli distribution to create its stochastic counterpart W_l'. In practice, dropout is applied after each convolution in the network with probability of dropping connections p_{drop}, except for the output layer. Importantly, dropout is enabled not only during training but also at inference time; we perform S stochastic forward passes to obtain class probabilities $\mathbf{P}_i \in \mathbb{R}^{M\times K\times S}$, which are averaged to produce the posterior distribution of class probabilities $\hat{P}_i \in \mathbb{R}^{M\times K}$. This process is known as MC dropout sampling (Kendall and Gal 2017).

Predictive Uncertainty Quantification Given S MC dropout samples with class probabilities $\mathbf{P}_i \in \mathbb{R}^{M\times K\times S}$ reshaped into $\mathbf{P}_i \in \mathbb{R}^{S\times M\times K}$, the total predictive uncertainty can be approximated by the variance of $\mathbf{P}_i$, defined as follows (Ribeiro et al. 2019):

$$\begin{aligned}\sigma[\mathbf{P}_i]^2 &\approx \mathrm{tr}(\underbrace{\mathbb{E}[\mathrm{diag}(\mathbf{P}_i) - \mathbf{P}_i\mathbf{P}_i^{\mathsf{T}}]}_{\text{aleatoric}} + \underbrace{\mathbb{E}[\mathbf{P}_i^2] - \mathbb{E}[\mathbf{P}_i]^2}_{\text{epistemic}}) \\ &= \mathrm{tr}(\sigma_a[\mathbf{P}_i]^2) + \mathrm{tr}(\sigma_e[\mathbf{P}_i]^2),\end{aligned} \tag{7}$$

where matrices $\sigma_a[\mathbf{P}_i]^2$ and $\sigma_e[\mathbf{P}_i]^2$ correspond to aleatoric and epistemic uncertainties, respectively. Their traces are then taken to produce a single aleatoric and epistemic uncertainty value per timestep.

Segmentation Objective Function

For unsupervised trajectory segmentation, we leverage the mutual-information-based deep clustering method in (Ji, Henriques, and Vedaldi 2019) for its demonstrated effectiveness in semantic image segmentation over centroid-based clustering approaches. It takes the K-dimensional softmax predictions for each timestep in the input sequence and attempts to maximize the mutual information between neighboring pairs of timestep patches.

We note that the original image-based work obtained pairs of neighboring patches not only within each image in the dataset, but also from synthetic images generated via multiple data augmentations; however, time series data augmentation is non-trivial, especially for the problem at hand. Intuitively, it would be hard to verify whether perturbed motion features would still realistically correspond to the same transportation mode. Therefore, considering only adjacent pairs of timestep patches centered at u and $u+t$, where $t \in T \subset \mathbb{Z}$ is a small displacement, the original clustering objective can be written as:

$$\max_{\Phi} \frac{1}{|T|} \sum_{t \in T} I(P_t), \tag{8a}$$

$$P_t = \frac{1}{n|\Omega|} \sum_{i=1}^{n} \sum_{u \in \Omega} \Phi_u(F_i) \cdot [\Phi(F_i)]_{u+t}^T, \tag{8b}$$

where Φ is a neural network, in this case BTCN, and $\Phi_u(F_i)$ is its output for patch $u \in \Omega = \{1, \ldots, M\}$ centered at timestep u within the input motion feature sequence F_i.

Experiments

Simulation Setup

All experiments were conducted on a server with an Intel Xeon Silver 4210 CPU clocked at 2.20GHz and nVidia GeForce RTX 2080Ti GPUs with 11 GB of GDDR6 memory. Deep learning models were developed using TensorFlow 2.3.0. Each baseline was implemented as per the description in its cited study. The reported values for all evaluation metrics were averaged over five executions.

Dataset This study is evaluated on the Geolife dataset by Microsoft Asia Research (Zheng et al. 2008b; 2008a), which has been widely used for GPS-based trajectory research. It contains 18,670 GPS trajectories obtained from 182 users over five years, totalling nearly 25 million GPS points and 1.3 million kilometers. Most trajectories were collected in Beijing, China at a sampling rate of $1-5$ seconds. Only 69 users have labeled parts of their trajectories by transportation mode; these include walking, bike, bus, car, taxi, train, subway and airplane. Following the dataset authors' recommendations, cars and taxis are treated as a single class, “car”, and trains and subways as “train”. As typically done in the literature (Zheng et al. 2008b; 2008a; Dabiri et al. 2020), we only retain the five classes for which there are sufficient samples, namely “walk”, “bike”, “bus”, “car”, and “train”. For evaluation purposes, we only use labeled trajectories; no ground-truth labels are involved in model training. All trajectories are split into chunks of length M and divided into training and test sets using a 90/10 ratio.

Preprocessing For each trajectory, we first discard any GPS point whose timestamp is greater than that of the next point, or whose geographical coordinates does not fall within valid ranges. Next, we extract motion features by first computing the relative geodesic distance Δx_{p_i} and elapsed time Δt_{p_i} between each GPS point p_i and its successor p_{i+1}. These two features enable the computation of higher-order distance derivatives, including velocity V_{p_i}, acceleration A_{p_i}, jerk J_{p_i}, and turn U_{p_i}:

$$\Delta x_{p_i} = \mathrm{Geodesic}(\mathrm{lat}_i, \mathrm{lon}_i, \mathrm{lat}_{i+1}, \mathrm{lon}_{i+1}), \tag{9a}$$

$$\Delta t_{p_i} = p_{i+1}[t] - p_i[t], \tag{9b}$$

$$V_{p_i} = \Delta x_{p_i} / \Delta t_{p_i}, \tag{9c}$$

$$A_{p_i} = (V_{p_{i+1}} - V_{p_i}) / \Delta t_{p_i}, \tag{9d}$$

$$J_{p_i} = (A_{p_{i+1}} - A_{p_i}) / \Delta t_{p_i}, \tag{9e}$$

$$\begin{aligned} U_{p_i} = {} & \tan^{-1} \frac{\mathrm{Geodesic}(\mathrm{lat}_i, \mathrm{lon}_i, \mathrm{lat}_{i-1}, \mathrm{lon}_i)}{\mathrm{Geodesic}(\mathrm{lat}_i, \mathrm{lon}_i, \mathrm{lat}_i, \mathrm{lon}_{i-1})} \\ & - \tan^{-1} \frac{\mathrm{Geodesic}(\mathrm{lat}_i, \mathrm{lon}_i, \mathrm{lat}_{i+1}, \mathrm{lon}_i)}{\mathrm{Geodesic}(\mathrm{lat}_i, \mathrm{lon}_i, \mathrm{lat}_i, \mathrm{lon}_{i+1})}. \end{aligned} \tag{9f}$$

Following previous transportation literature (Zheng et al. 2008b; Dabiri et al. 2020; Yu 2020), we empirically select velocity, acceleration, jerk, and turn as the time series features to train our trajectory segmentation model. The features are standardized by removing the mean and scaling to unit variance. We empirically set the sample length M to 128, such that $F_i \in \mathbb{R}^{128 \times 4}$.

Model Configuration BTCN uses 3 consecutive pairs of temporal residual and SE blocks. Within the i-th temporal residual block, both convolutions combine a larger kernel size of 8 with exponentially increasing dilation rates $d = 2^i$, $i \in \{0, 1, 2\}$. The dropout probability p_{drop} is set to 0.2 for all dropout layers, as in (Ribeiro et al. 2019); it is maintained at test time as required for MC dropout sampling, from which $S = 50$ samples are obtained. For all SE blocks, the reduction ratio r is set to 8. BTCN is trained for 200 epochs using the Adam optimizer with a learning rate of 10^{-5} and default hyperparameters $\beta_1 = 0.9$, $\beta_2 = 0.999$.

Baselines BTCN is evaluated against the three most prominent families of trajectory segmentation methods among the relatively few found in the GPS-based transportation mode identification literature:

- **Uniform segmentation.** A naive approach to trajectory segmentation is to extract fixed-size, nonoverlapping chunks of timesteps from each trajectory. (Dabiri and Heaslip 2018) uses a window of 200 points, which they found to be the median length of the labeled GPS trajectories in Geolife.

Table 1: GPS trajectory segmentation evaluation results. Where applicable, accuracies are reported for each class, with ACC denoting global accuracy. Higher ACC, lower MAE, and $\mathrm{PR} \approx 1$ is better. Training time is in minutes.

Segmentation Method	*Walk*	*Bike*	*Bus*	*Car*	*Train*	ACC	MAE	PR	Train (min)
(Dabiri and Heaslip 2018)			N/A			N/A	728.4	3.12	N/A
(Zheng et al. 2008b)			N/A			N/A	234.5	1.91	N/A
(Xiao, Juan, and Zhang 2015)			N/A			N/A	287.7	2.34	N/A
(Dabiri et al. 2020)			N/A			N/A	526.1	**1.23**	0.02
TCN (non-Bayesian, no SE)	0.821	0.755	0.416	0.376	0.726	0.619	31.5	3.51	33.89
TCN (non-Bayesian, $r = 8$)	0.799	0.782	0.441	0.367	**0.812**	0.640	26.1	2.92	45.12
BTCN (no SE)	0.792	0.752	0.414	0.391	0.775	0.625	29.7	2.84	33.67
BTCN ($r = 2$)	**0.827**	0.727	0.349	0.478	0.753	0.627	25.8	2.04	45.14
BTCN ($r = 4$)	0.809	0.773	0.353	**0.485**	0.744	0.632	28.3	2.43	45.21
BTCN (r = 8)	0.820	**0.811**	0.396	0.475	0.789	**0.658**	**24.7**	1.97	45.15
BTCN ($r = 16$)	0.773	0.799	**0.454**	0.416	0.751	0.638	28.1	2.44	44.93

- **Heuristics-based change point detection.** Following the intuition that walking must precede any change of transportation mode, the authors in (Zheng et al. 2008b) propose a four-step method that divides trajectories into alternating "walk" and "non-walk" segments, based on distance, velocity, and acceleration thresholds. Similarly, (Xiao, Juan, and Zhang 2015) identifies not only "walk" segments, but also gaps caused by GPS signal interruption, extracting change points at the boundaries of "walk" and gap segments.
- **Optimization-based change point detection.** (Dabiri et al. 2020) first applies uniform segmentation to each trajectory, then converts the obtained segments into multivariate time series of velocity and acceleration features before feeding them to an optimization-based model. The latter attempts to find subsegments such that the homogeneity within each subsegment is maximized, while the number of change points is penalized by a hyperparameter-controlled linear function.

Evaluation Metrics Following standard clustering evaluation practice (Xie, Girshick, and Farhadi 2016; Ji, Henriques, and Vedaldi 2019), all classes predicted by BTCN and its variants are first mapped to the ground-truth classes using linear assignment (Kuhn 1955). Segmentation performance is then measured by timestep-level accuracy (ACC).

While BTCN labels each timestep by transportation mode, the above baselines are label-agnostic: they merely detect transportation mode change points, i.e., discrete timesteps where the transportation mode changes, without knowledge of the specific mode corresponding to each implicitly defined segment. To allow for direct comparison with BTCN, we extract change points by simply scanning its output from left to right. We then evaluate BTCN against the above baselines using the following evaluation metrics:

- **Mean Absolute Error (MAE)**. As the number of predicted and ground-truth change points may differ, we first identify for each predicted change point $\hat{y}_i \in \hat{Y}$ its nearest ground-truth $y_i \in Y$. Given that most trajectories in Geolife have been sampled at a rate of 1–5 seconds, we preserve generality by measuring the distance between two change points by the number of discrete timesteps separating them. Then, given n predicted change points $\hat{Y}$ and their nearest ground-truth change points Y', the MAE is calculated as $\mathrm{MAE} = \frac{1}{n}\sum_{1}^{n} |y'_i - \hat{y}_i|$. We select this metric under the assumption that the number of timesteps between y'_i and $\hat{y}_i$ matters more than whether $\hat{y}_i$ precedes or comes after y'_i.
- **Prediction Ratio (PR)**. Let n_{pred}, n_{true} be the total numbers of predicted change points $\hat{Y}$ and true change points Y, respectively. The PR is computed as $\mathrm{PR} = n_{\text{pred}}/n_{\text{true}}$. When $\mathrm{PR} < 1$ or $\mathrm{PR} > 1$, the segmentation model is said to under- or over-predict change points. Either case could complicate transportation mode identification, even if the MAE is low: too few change points could mean noisy segments with more than one transportation mode, while too many would result in short, harder to classify segments.

Experimental Results

Performance Evaluation Our experimental results are summarized in Table 1. BTCN consistently and significantly outperformed all evaluated methods, with its lowest MAE of 24.7 constituting a nearly 10× improvement over the best-performing baseline (Zheng et al. 2008b). Recall that our definition of MAE measures the mean number of discrete timesteps between true and predicted change points; for unit GPS sampling rate, our result means BTCN would only take approximately 25 seconds on average to identify a change in transportation mode, while the above baseline would require nearly 4 minutes. Although the proposed method did demonstrate change point over-prediction on par with the baselines, it is important to note that BTCN is designed for timestep-level classification rather than change point detection. In fact, we formalized trajectory segmentation based on mutual information maximization alone, without explicitly penalizing over-segmentation. This is an open problem, especially in the absence of labels, and is left for future work.

Although each of the selected baselines has its own merit, they are not without limitations. Uniform segmentation (Dabiri and Heaslip 2018) offers simplicity at the expense of constructing noisy segments likely to contain multiple transportation modes. Indeed, uniform segmentation attained both the highest MAE and PR among all meth-

Table 2: Performance and uncertainty metrics for BTCN over number of Monte Carlo samples.

S	ACC	MAE	PR	Aleatoric	Epistemic
1	0.6763	32.6	4.03	1.047E-02	0.0
2	0.6769	31.0	3.23	1.050E-02	3.902E-01
5	0.6775	29.3	2.76	1.043E-02	6.401E-01
10	0.6776	28.3	2.41	1.042E-02	7.162E-01
20	0.6777	27.2	2.22	1.048E-02	7.516E-01
50	0.6778	26.5	2.08	1.044E-02	7.746E-01
100	0.6778	26.1	2.02	1.047E-02	7.818E-01

ods. Guided by domain-specific knowledge, the heuristics-based segmentation baselines (Zheng et al. 2008b; Xiao, Juan, and Zhang 2015) both produced much lower MAEs, although they still predicted approximately twice the number of change points compared to the ground truth. This might be due to heuristics not always being able to account for unexpected traffic conditions (Zheng et al. 2008b).

Scoring the second highest MAE of 526.1, despite having the best PR, the sub-par performance of optimization-based segmentation (Dabiri et al. 2020) is not surprising: this method is individually applied to truncated segments, thereby failing to consider the entire data distribution. In other words, it assumes that the samples are independent and identically distributed. Such an assumption is unlikely to hold for trajectory data collected from numerous users at varying times and traffic or weather conditions, as is the case with Geolife.

Ablation Study To quantify the contribution of dropout variational inference and SE blocks to the performance of BTCN, we evaluate the latter when both components are disabled, as well as by removing either component while maintaining the other. As shown in Table 1, the non-Bayesian variant without SE blocks achieved the lowest global and per-class accuracy. Adding MC dropout pushed the global accuracy from 61.9% to 62.5%, while enabling SE blocks with $r = 8$ instead boosted it to 64.0%, and combining both components resulted in the highest global accuracy at 65.8%. In fact, the best accuracy for four out of five classes was attained by the full BTCN, albeit with different reduction ratios r, while removing either of the Bayesian or SE components resulted in nearly twice the over-prediction of change points.

Finally, Table 2 shows the sensitivity of evaluation and uncertainty metrics to the number of MC samples S. For this experiment, we used the best-performing BTCN out of 5 executions. Although global accuracy did not seem to drastically improve as S increased, the MAE dropped by about 19% and the PR was nearly halved. No noticeable improvement to accuracy was observed when $S > 50$, which is the value we used throughout our experiments.

Uncertainty-filtered Segmentation For safety-critical intelligent transportation applications involving GPS trajectory segmentation, the ability of a model to selectively make timestep-level predictions based on some measure of confidence could be beneficial. Fig. 4 shows how global and per-sample accuracy varies under different confidence percentiles for the best-performing BTCN out of 5 executions. In this context, confidence was simply defined as the negative aleatoric or epistemic uncertainty.

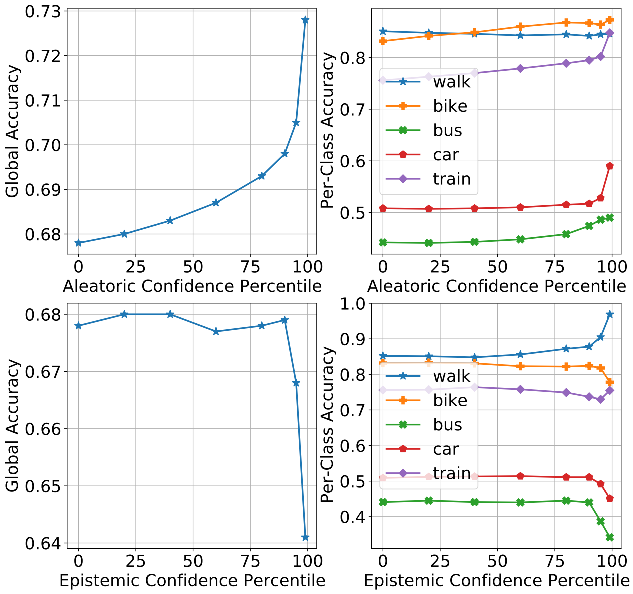


Figure 4: Global and per-class accuracy of BTCN when only classifying timesteps that exceed confidence thresholds.

It appears that selectively classifying timesteps with higher aleatoric confidence consistently achieved higher global and per-class accuracy. On the contrary, we noted minimal global accuracy improvement in the case of epistemic confidence; there were almost no variations until the 90th percentile, followed by a sharp drop for the 95th and 99th percentiles. It is hypothesized that the poor performance of epistemic confidence as a measure of accuracy in our work is tied to the unsupervised, mutual-information-based objective function that BTCN was trained with.

Conclusion and Future Work

In this paper, we introduced a novel GPS trajectory segmentation approach to address the shortcomings of related GPS-based work in learning from unlabeled data and capturing predictive uncertainty. Viewing trajectory segmentation through the scope of semantic image segmentation, the proposed BTCN reached 65.8% timestep-level accuracy on Microsoft's Geolife dataset, significantly outperforming established GPS-based baselines. We also conducted an ablation study to empirically validate the necessity of its components, and showed that BTCN effectively captured uncertainty by producing higher accuracy for input timesteps with lower aleatoric uncertainty.

In future work, we will investigate graph neural networks to capture not only temporal but also spatial information from non-Euclidean road network representations. We will also explore how to effectively penalize over-segmentation in the absence of labels.

Acknowledgement

This work was supported by the General Program of Guangdong Basic and Applied Basic Research Foundation No. 2019A1515011032 and by the Guangdong Provincial Key Laboratory of Brain-inspired Intelligent Computation No. 2020B121201001. James J.Q. Yu is the corresponding author.

References

Badrinarayanan, V.; Kendall, A.; and Cipolla, R. 2017. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39(12):2481–2495.

Bai, S.; Kolter, J. Z.; and Koltun, V. 2018. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*.

Bolbol, A.; Cheng, T.; Tsapakis, I.; and Haworth, J. 2012. Inferring hybrid transportation modes from sparse GPS data using a moving window SVM classification. *Computers, Environment and Urban Systems* 36(6):526–537.

Caron, M.; Bojanowski, P.; Joulin, A.; and Douze, M. 2018. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European Conference on Computer Vision*, 132–149.

Chen, L.-C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; and Yuille, A. L. 2014. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv preprint arXiv:1412.7062*.

Chen, L.-C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; and Yuille, A. L. 2017. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40(4):834–848.

Chen, L.-C.; Zhu, Y.; Papandreou, G.; Schroff, F.; and Adam, H. 2018. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European Conference on Computer Vision*, 801–818.

Dabiri, S., and Heaslip, K. 2018. Inferring transportation modes from GPS trajectories using a convolutional neural network. *Transportation Research Part C: Emerging Technologies* 86:360–371.

Dabiri, S.; Lu, C.-T.; Heaslip, K.; and Reddy, C. K. 2020. Semi-supervised deep learning approach for transportation mode identification using GPS trajectory data. *IEEE Transactions on Knowledge and Data Engineering* 32(5).

Fu, J.; Liu, J.; Tian, H.; Li, Y.; Bao, Y.; Fang, Z.; and Lu, H. 2019. Dual attention network for scene segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3146–3154.

Ghasedi Dizaji, K.; Herandi, A.; Deng, C.; Cai, W.; and Huang, H. 2017. Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization. In *Proceedings of the IEEE International Conference on Computer Vision*, 5736–5745.

Hjelm, R. D.; Fedorov, A.; Lavoie-Marchildon, S.; Grewal, K.; Bachman, P.; Trischler, A.; and Bengio, Y. 2018. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*.

Hu, W.; Miyato, T.; Tokui, S.; Matsumoto, E.; and Sugiyama, M. 2017. Learning discrete representations via information maximizing self-augmented training. *arXiv preprint arXiv:1702.08720*.

Hu, J.; Shen, L.; and Sun, G. 2018. Squeeze-and-excitation networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 7132–7141.

Ji, X.; Henriques, J. F.; and Vedaldi, A. 2019. Invariant information clustering for unsupervised image classification and segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, 9865–9874.

Jiang, Z.; Zheng, Y.; Tan, H.; Tang, B.; and Zhou, H. 2017. Variational deep embedding: An unsupervised and generative approach to clustering. In *International Joint Conference on Artificial Intelligence*.

Kendall, A., and Gal, Y. 2017. What uncertainties do we need in bayesian deep learning for computer vision? In *Advances in Neural Information Processing Systems*, 5574–5584.

Kuhn, H. W. 1955. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly* 2(1-2):83–97.

Kwon, Y.; Won, J.-H.; Kim, B. J.; and Paik, M. C. 2020. Uncertainty quantification using bayesian neural networks in classification: Application to biomedical image segmentation. *Computational Statistics & Data Analysis* 142:106816.

Li, F.; Qiao, H.; and Zhang, B. 2018. Discriminatively boosted image clustering with fully convolutional autoencoders. *Pattern Recognition* 83:161–173.

Long, J.; Shelhamer, E.; and Darrell, T. 2015. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3431–3440.

Mukherjee, S.; Asnani, H.; Lin, E.; and Kannan, S. 2019. Clustergan: Latent space clustering in generative adversarial networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 4610–4617.

Ribeiro, F. D. S.; Calivá, F.; Swainson, M.; Gudmundsson, K.; Leontidis, G.; and Kollias, S. 2019. Deep bayesian self-training. *Neural Computing and Applications* 1–17.

Ronneberger, O.; Fischer, P.; and Brox, T. 2015. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-assisted Intervention*, 234–241.

Stenneth, L.; Wolfson, O.; Yu, P. S.; and Xu, B. 2011. Transportation mode detection using mobile phones and GIS information. In *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 54–63.

Sun, K.; Xiao, B.; Liu, D.; and Wang, J. 2019. Deep high-resolution representation learning for human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5693–5703.

Xiao, G.; Juan, Z.; and Zhang, C. 2015. Travel mode detection based on GPS track data and Bayesian networks. *Computers, Environment and Urban Systems* 54:14–22.

Xie, J.; Girshick, R.; and Farhadi, A. 2016. Unsupervised deep embedding for clustering analysis. In *Proceedings of the 33rd International Conference on Machine Learning*, 478–487.

Yang, J.; Parikh, D.; and Batra, D. 2016. Joint unsupervised learning of deep representations and image clusters. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5147–5156.

Yu, F., and Koltun, V. 2016. Multi-scale context aggregation by dilated convolutions. In *International Conference on Learning Representations*.

Yu, J. J. Q. 2020. Travel mode identification with gps trajectories using wavelet transform and deep learning. *IEEE Transactions on Intelligent Transportation Systems*.

Zhang, R.; Xie, P.; Wang, C.; Liu, G.; and Wan, S. 2019. Classifying transportation mode and speed from trajectory data via deep multi-scale learning. *Computer Networks* 162:106861.

Zheng, Y.; Li, Q.; Chen, Y.; Xie, X.; and Ma, W.-Y. 2008a. Understanding mobility based on GPS data. In *Proceedings of the 10th International Conference on Ubiquitous Computing*, 312–321.

Zheng, Y.; Liu, L.; Wang, L.; and Xie, X. 2008b. Learning transportation mode from raw GPS data for geographic applications on the web. In *Proceedings of the 17th International Conference on World Wide Web*, 247–256.