UNCERTAINTY-AWARE HUMAN MOBILITY MODELING AND ANOMALY DETECTION

Anonymous authors

Paper under double-blind review

ABSTRACT

Given the GPS coordinates of a large collection of human agents over time, how can we model their mobility behavior toward effective anomaly detection (e.g. for badactor or malicious behavior detection) without any labeled data? Human mobility and trajectory modeling have been studied extensively with varying capacity to handle complex input, and performance-efficiency trade-offs. With the arrival of more expressive models in machine learning, we attempt to model GPS data as a sequence of stay-point events, each with a set of characterizing spatiotemporal features, and leverage modern sequence models such as Transformers for un/self-supervised training and inference. Notably, driven by the inherent stochasticity of certain individuals' behavior, we equip our model with aleatoric/data uncertainty estimation. In addition, to handle data sparsity of a large variety of behaviors, we incorporate epistemic/model uncertainty into our model. Together, aleatoric and epistemic uncertainty enable a robust loss and training dynamics, as well as uncertaintyaware decision making in anomaly scoring. Experiments on large expert-simulated datasets with tens of thousands of agents demonstrate the effectiveness of our model against both forecasting and anomaly detection baselines. All code is available at https://anonymous.4open.science/r/mobility-ad.

026 027

025

003

006 007 008

009 010 011

012

013

014

015

016

017

018

019

021

023

028 1 INTRODUCTION

Anomaly detection of human mobility has become crucial for various applications, ranging from
 security and surveillance to health monitoring. Accurately detecting anomalous human behavior
 from GPS data can reveal critical insights, such as abnormal patterns that indicate security threats or
 spread of infectious diseases Meloni et al. (2011); Barbosa et al. (2018); Stanford et al. (2024).

However, the complexity and inherent uncertainty of human behavior make this task particularly challenging: (1) The first difficulty lies in capturing the complex *spatiotemporal dependencies* both within individual activities and between multiple activities across different times and locations. The former aims to capture the correlations between different spatiotemporal features (a.k.a. markers) of a single activity; such as the time-of-day (e.g. 4am) being indicative of POI location (e.g. home). The latter associates with the dependency patterns between different activities over time (e.g. restaurant at weekday lunchtime followed by office building). (2) Furthermore, human activity data is abound with *uncertainty*—arising from the unpredictable behavior of inherently stochastic individuals as well as data sparsity—making accurate anomaly detection even more difficult.

Prior work on behavior modeling often focused on temporal event forecasting, either ignoring spatial information Minor et al. (2015); Manzoor and Akoglu (2017) or based only on a few markers Du et al. (2016); Zhou et al. (2022). With the emergence of the Transformer architecture Vaswani et al. (2023), more recent work modeled complex spatiotemporal events more expressively Xue et al. (2022); Corrias et al. (2023). However, to our knowledge, no existing work considered uncertainty-aware anomaly detection of human mobility.

To address both of the above challenges, we introduce UIFORMER for <u>uncertainty-incorporated</u>
human behavior modeling and anomaly detection, which takes advantage of a "dual" Transformer
architecture Truong Jr and Bepler (2023) as well as aleatoric and epistemic uncertainty into account.
Transformer has emerged as the de facto model for modern AI problems, due to its superior ability to
model long-range dependencies. Through a dual architecture equipped with both feature- and event-level attention, we address the first challenge in capturing the intra- and inter-event dependencies.

054 For the second challenge, we first account for data (a.k.a. aleatoric) uncertainty, which stems from the inherent stochasticity in certain human behavior. For example, it is typical of some populations 056 (e.g. retired individuals) to be less predictable relative to some others (e.g. shift workers with a strict schedule). In addition to data uncertainty, we incorporate model (a.k.a. epistemic) uncertainty, which 058 typically stems from incomplete or sparse observations from a potentially complex underlying data generating distribution. Explicitly modeling both aleatoric (data) and epistemic (model) uncertainty benefits UIFORMER in two ways: First, it factors into the learning objective, enabling a more robust 060 training. Second, it allows the model to reason effectively about the confidence of its predictions, 061 thus enabling more reliable inference. Specifically, while it is typical for forecasting-based detection 062 methods to flag anomalies based on the deviation between the predicted and observed values alone, 063 our uncertainty-aware UIFORMER considers both the predicted values as well as total estimated 064 uncertainties, thus providing a more nuanced and accurate approach to anomaly detection. 065

066 The following summarizes the main contributions of this work.

- **Problem formulation:** We cast the anomaly detection problem in human mobility onto sequence prediction, where we translate the regular GPS readings (of the form $\langle x, y, t \rangle$) into irregular staypoint events over time with spatiotemporal features. This step resembles a form of "tokenization" of the raw input data to a higher-level representation for sequence modeling.
- Uncertainty-aware dependency modeling: We design UIFORMER, a *dual* Transformer architecture that is *uncertainty-aware*; which is equipped with (i) both feature-level and event-level attention—for capturing dependencies among features as well as across events, respectively; and (*ii*) both data and model uncertainty estimation—toward accounting for inherent stochasticity in human behavior as well as data scarcity, respectively.
 - Uncertainty-aware anomaly scoring: Beyond prediction, we extend our uncertainty modeling to anomaly detection, which allows the model to flag abnormal behavior with greater accuracy by factoring in both prediction value and uncertainty estimates.
 - Effectiveness: Through extensive experiments on expert-simulated benchmark datasets, we show that UIFORMER significantly outperforms both forecasting as well as anomaly detection baselines. Ablation analyses demonstrate UIFORMER's key advantages: accurate uncertainty estimates and effective uncertainty-aware anomaly scores.

2 PROBLEM AND PRELIMINARIES

076

077

078

079

081

082

084

091

104 105 We introduce related concepts and formulate the anomaly detection problem on human mobility.

Stay-point Event. As shown in Figure 1, a stay-point event is extracted from raw GPS data that is used to describe an individual's daily activity. Let e_i be an event with spatiotemporal features (or markers) x_i :

$$\boldsymbol{x}_{i} = \left(x_{i}^{\mathrm{st}}, x_{i}^{\mathrm{sd}}, x_{i}^{\mathrm{x}}, x_{i}^{\mathrm{y}}, x_{i}^{\mathrm{poi}}, x_{i}^{\mathrm{dow}}\right), \qquad (1)$$

where $\mathcal{F}_n = \{x_i^{\text{st}}, x_i^{\text{sd}}, x_i^{\text{x}}, x_i^{\text{y}}\}$ is the numerical feature set: x_i^{st} and x_i^{sd} are the start time and stay duration of the event, $x_i^{\text{x}}, x_i^{\text{y}}$ are the two-dimensional coordinates depicting the latitude and longitude of the event's location. $\mathcal{F}_c = \{x_i^{\text{poi}}, x_i^{\text{dow}}\}$ is the categorical feature set, depicting the Point-of-Interest (POI) such as office building, store, etc. and Day-of-Week (DOW) of the event, respectively.



Figure 1: Spatiotemporal event sequence with various features (a.k.a. markers) per event.

100 Spatiotemporal Event Sequence. Let $\mathcal{E}^u = [e_1^u, e_2^u, \dots, e_{N_u}^u]$ be the event sequence that records 101 all historical events of individual u in the dataset, where N_u is u's total number of historical events. 102 Furthermore, let $e_i^{u,d}$ be the *i*-th event of individual u at day d, and $N_{u,d}$ be the number of events of 103 u on day d. The event sequence with w-day time window can be given as

$$\mathcal{E}_{w}^{u} = [e_{1}^{u,d-w}, e_{2}^{u,d-w}, \dots, e_{1}^{u,d-w+1}, \dots, e_{1}^{u,d}, \dots, e_{N_{u,d}}^{u,d}] .$$
⁽²⁾

106 Unsupervised Human Mobility Anomaly Detection. Given a individual u's event sequence in a 107 w-day window \mathcal{E}_u^w and a target event $e \in \mathcal{E}_u^w$, we aim to learn an anomaly score function to identify whether e is anomalous, i.e. not aligned with u's typical behavior pattern, without any labeled data.



Figure 2: Proposed model architecture for human mobility behavior modeling. Raw GPS data is represented as a (ordered) sequence of stay-point events each with a (unordered) sequence of spatiotemporal markers. Dual-Transformer architecture models each individual's sequence-of-sequences via both feature- and event-level attention, as well as uncertainty estimation that simultaneously enables robust training and inform anomaly scoring at inference.

3 PROPOSED MODEL

135 136

134

Figure 2 illustrates the overall architecture of UIFORMER, which learns human mobility patterns in 137 a self-supervised manner through pre-training, with the goal of accurately reconstructing masked 138 events as well as evaluating the prediction uncertainty. In a nutshell, the feature tokenizer first projects 139 features into high-dimensional space by representing each event's feature as an individual token (§3.1). 140 Then, the Dual-Transformer encoder further encodes the input by both feature-level and event-level 141 self-attention (§3.2). Lastly, unlike previous work, we introduce the uncertainty-aware decoder to 142 recover all features of the masked event along with the aleatoric and epistemic uncertainty associated 143 with each feature (§3.3). These estimates are used to compare the predicted versus observed events at 144 inference time for uncertainty-incorporated anomaly scoring (§3.4). 145

3.1 FEATURE TOKENIZER

The feature tokenizer transforms all features $x \in \mathbb{R}^F$ of an event into embeddings $e \in \mathbb{R}^D$, where Fis the number of input features and D is the embedding dimension. Specifically, a numerical feature $x_j^{(num)}$ is projected by a linear transformation with weight $W_j^{(num)} \in \mathbb{R}^D$ and bias $b_j \in \mathbb{R}^D$, and the embedding of a categorical feature is implemented as the embedding lookup table $W_j^{(cat)} \in \mathbb{R}^{C_j \times D}$, where C_j is the total number of categories for feature $x_j^{(cat)}$. Overall:

154

146

147

$$egin{aligned} oldsymbol{e}_j^{(num)} &= oldsymbol{b}_j^{(num)} + x_j^{(num)} \cdot oldsymbol{W}_j^{(num)} &\in \mathbb{R}^D \;, \ oldsymbol{e}_j^{(cat)} &= x_j^{(cat)} oldsymbol{W}_j^{(cat)} &\in \mathbb{R}^D \;, \ oldsymbol{e} &= ext{stack} \left[oldsymbol{e}_1^{(num)}, \, \dots, \, oldsymbol{e}_{k^{(num)}}^{(num)}, \, oldsymbol{e}_1^{(cat)}, \, \dots, \, oldsymbol{e}_{k^{(cat)}}^{(cat)}
ight] \in \mathbb{R}^{F imes D} \;. \end{aligned}$$

where $x_j^{(cat)}$ is represented as a one-hot vector. The feature embedding $e \in \mathbb{R}^{F \times D}$ of one event is the concatenation of all numerical embeddings and categorical embeddings. Denoting *B* as the batch size and *L* the max length of \mathcal{E}_w^u in a batch, the output of the feature tokenizer is $E_0 \in \mathbb{R}^{B \times L \times F \times D}$.

162 3.2 DUAL TRANSFORMER ENCODER 163

164 Inspired by (Truong Jr and Bepler, 2023), we design a Dual Transformer encoder that treats the data 165 as a sequence-of-sequences, which takes the sequence of events each with a sequence of feature embeddings as input, and models both feature-level and event-level interactions with two types of 166 Transformer-based (Vaswani et al., 2023) components. A description of the Transformer block is 167 included in Appendix B. 168

169 Feature-level Transformer. By considering each feature as an input token to the Transformer, the 170 Feature-level Transformer represents a feature in a way that explicitly incorporates other features' 171 information. In the implementation, we convert the feature tokenizer's output $E_0 \in \mathbb{R}^{B \times L \times F \times D}$ into $E_1 \in \mathbb{R}^{(B \times L) \times F \times D}$, which is then fed into M_1 Transformer blocks to get the updated feature embeddings $E_2 \in \mathbb{R}^{(B \times L) \times F \times D}$. Note that we do not include any positional embedding as input for 172 173 this module as there is no sequential relationship between the features of an event. 174

175 **Event-level Transformer.** Based on E_2 , we calculate 176 an event's embedding by averaging all its correspond-177 ing feature embeddings, which results in the input of the 178 Event-level Transformer $E_3 \in \mathbb{R}^{B \times L \times D}$. The Event-level 179 Transformer then captures the dependencies (e.g., sequential patterns) between different events by considering each 181 event as a token. Moreover, considering that the sequential information in the event sequence is important to reflect 182

| Within-day Position Encoding | | | | | | | | | |
|------------------------------|-----------------|-----------------|-----------------|-------------|--|--|--|--|--|
| Day d_1 : | $e_{1}^{d_{1}}$ | $e_2^{d_1}$ | $e_{3}^{d_{1}}$ | $e_4^{d_1}$ | | | | | |
| $Dayd_2\!\!:$ | $e_1^{d_2}$ | $e_{2}^{d_{2}}$ | $e_{3}^{d_{2}}$ | | | | | | |
| | 0 | 1 | 2 | 3 | | | | | |

Figure 3: Within-day position encoding.

human mobility behavior, we design two types of posi-183 tional encoding for each event: (i) Sequence positional encoding; used to describe the order of an 184 event in the input sequence, and (ii) Within-day positional encoding (as shown in Figure 3); used to 185 describe the order of the event in its specific day, explicitly demarcating day boundaries. After M_2 updates of the Event-level Transformer blocks, we get the final output of the encoder $\bar{E} \in \mathbb{R}^{B \times L \times D}$ 187

188 189

190

3.3 UNCERTAINTY-AWARE DECODER

Motivation. Uncertainty modeling has shown great promise in various fields, such as computer 191 vision (Kendall and Gal, 2017; Kendall et al., 2016) and natural language processing (Xiao and Wang, 192 2019; Gal and Ghahramani, 2016b). In particular, human mobility sequences are inherently associated 193 with uncertainty. For instance, certain individuals (e.g., shift workers) may exhibit highly predictable 194 patterns (i.e., low uncertainty), while others (e.g., retirees) can be much more unpredictable in their 195 activities over time. This type of uncertainty, known as aleatoric uncertainty (or data uncertainty) 196 (Kendall and Gal, 2017), arises naturally from the data itself and cannot be reduced by simply adding 197 more training data. In addition to aleatoric uncertainty, our setting also involves epistemic uncertainty, which refers to the uncertainty in the model itself due to limited knowledge. This type of uncertainty 199 captures the model's confidence in its own predictions and can be reduced by acquiring more data or improving the model specification or architecture. In human event sequence modeling, epistemic 200 uncertainty is crucial for identifying situations where the model lacks sufficient information to make 201 confident predictions, especially in underrepresented or complex scenarios. Furthermore, given that 202 different features can have different predictability even for the same individual/event, we design a 203 feature-level uncertainty learning mechanism. In the following part, we elaborate on how we model 204 the epistemic and aleatoric uncertainty for both numerical and categorical features. 205

206 207

3.3.1 **EPISTEMIC UNCERTAINTY MODELING**

208 Inspired by (Kendall, 2018), we model the epistemic uncertainty by placing a prior distribution over 209 the model's parameters, and then try to capture how much these parameters vary given observed data. 210

211

To ease the presentation, we use s to denote the input instance \mathcal{E}_w^u , and θ to denote all parameters of the model \mathcal{M} . We start by assuming θ follows a prior distribution, which reflects our initial belief 212 about the possible values of these parameters. The goal is to update this belief based on the available 213 data s by calculating the posterior distribution of θ , that is $p(\theta|s)$. This posterior captures how the 214 parameters might vary, thereby reflecting the model's uncertainty about its predictions. Formally, 215 Bayes' theorem (Bayes, 1763) gives us the posterior: $p(\theta|s) = \frac{p(s|\theta)p(\theta)}{n(s)}$.

However, computing the exact posterior distribution in deep learning models is often intractable due to the high dimensionality of the parameter space and the complexity of the model. As a practical alternative, we apply Monte Carlo (MC) Dropout (Gal and Ghahramani, 2016a) to approximate the posterior. In practice, we introduce dropout layers after the embedding layer of each feature with a dropout ratio of 0.05, which randomly drops a subset of neurons during both training and inference. MC Dropout works by sampling different subsets of model parameters θ during each forward pass, thereby creating an ensemble of models. The variance in the predictions across these passes provides an estimate of the epistemic uncertainty.

224 Epistemic uncertainty for numerical features: The decoding of numerical feature $f \in \mathcal{F}_n$ is modeled 225 as a regression task. In this case, the variance of the predictions across T (equals 50 in our model) 226 steps of stochastic forward passes quantifies the epistemic uncertainty α_f^{num} for f, formulated as

235 236 237

240

250

251

264 265

$$\alpha_f^{num} : \operatorname{Var}(y) = \frac{1}{T} \sum_{t=1}^T \left(\mathcal{M}_{\theta_t}(s) - \bar{y} \right)^2 , \quad \text{where } \bar{y} = \frac{1}{T} \sum_{t=1}^T \mathcal{M}_{\theta_t}(s) , \tag{3}$$

230 Epistemic uncertainty for categorical features: The decoding of categorical feature $f \in \mathcal{F}_c$ is 231 modeled as a classification task. We apply the softmax function to the logits produced by the 232 model, giving a probability distribution over the classes: $p(y|\mathcal{M}_{\theta}(s)) = \operatorname{softmax}(\mathcal{M}_{\theta}(s))$. Here, the 233 epistemic uncertainty is measured by the entropy of the predicted probability distributions averaged 234 by multiple forward processes, formulated as

$$\alpha_f^{cls}: H(p) = -\sum_c p_c \log(p_c) , \quad \text{where } p(y=c|s) = \frac{1}{T} \sum_{t=1}^T \operatorname{softmax}(\mathcal{M}_{\theta_t}(s)) .$$
(4)

Overall, we approximate the epistemic uncertainty for both numerical and categorical features by MC Dropout, providing a practical and scalable way to capture the model's uncertainty in its predictions.

241 3.3.2 ALEATORIC UNCERTAINTY MODELING

We model aleatoric uncertainty by placing a distribution over the model's output. For example, in this paper, we model the regression output as a Gaussian distribution with random noise, i.e., $p(y|\mathcal{M}_{\theta}(s)) = \mathcal{N}(\mathcal{M}_{\theta}(s), \sigma^2)$, where aleatoric uncertainty aims to learn the variance of noise as a function of the input data.

Aleatoric uncertainty for numerical features: Let σ_f denote the noise level of a feature $f \in \mathcal{F}_n$, which captures how much noise we have in the output of the feature. Moreover, we assume that σ_f is data-dependent and can vary with the input. Then, σ_f can be learned as a function of the data:

$$\mathcal{L}_{f}^{reg} = \frac{1}{2\sigma_{f}^{2}} \left\| y_{f} - \widehat{y}_{f} \right\|^{2} + \frac{1}{2} \log \sigma_{f}^{2} , \qquad (5)$$

where y_f and \hat{y}_f are the true and predicted value of f. σ_f is also a model's output that serves as a 253 learned loss attenuation. It gives less penalty to the model when the input data is associated with high 254 aleatoric uncertainty, thus making the loss more robust to noisy data. The second term in Eq. (5) is a regularization term that prevents the model from learning a high uncertainty score for all instances, 255 trivially driving the first term to zero. In practice, we further adopt a more numerically stable loss 256 given as: $\mathcal{L}_{f}^{reg} = \frac{1}{2} \exp(-r_{f}) \|y_{f} - \hat{y}_{f}\|^{2} + \frac{1}{2}r_{f}$, where $r_{f} = \log \sigma_{f}^{2}$. Instead of predicting the variance directly, predicting the log variance followed by an exponential function can ensure the 257 258 positive value of the variance and make the training more numerically stable. In that case, the aleatoric 259 uncertainty for numerical feature f is denoted as $\beta_f^{num} := \sigma_f^2$. 260

Aleatoric uncertainty for categorical features: Let C_f denote the number of unique categories (i.e. classes) of feature $f \in \mathcal{F}_c$. For the classification task, the model assumes that the prediction logits at sample time t for each category follows a Gaussian distribution:

$$\widehat{\boldsymbol{m}}_{f,t} = \boldsymbol{u}_f + \boldsymbol{\sigma}_f \boldsymbol{\epsilon}_t, \quad \boldsymbol{\epsilon}_t \sim \mathcal{N}(0, I) , \qquad (6)$$

where the predicted mean logits $u_f \in \mathbb{R}^{C_f}$ and uncertainty term $\sigma_f \in \mathbb{R}^{C_f}$ are the model outputs, as a function of its parameters at the *t*-th sample time. $\epsilon_t \in \mathbb{R}^{C_f}$ represents a random vector drawn from a unit normal distribution $\mathcal{N}(0, I)$. Here σ_f accounts for the inherent uncertainty in the feature when conducting the prediction. The aleatoric uncertainty of categorical feature *f* is then the average of variance across all classes, given as $\beta_f^{cls} := \frac{1}{C_t} \sum_{c'} \sigma_{c'}^2$. To learn such uncertainty, the loss function \mathcal{L}_{f}^{cls} calculates the cross entropy based on the average predicted logits, normalized over all possible categorical outcomes:

$$\mathcal{L}_{f}^{cls} = -\log\frac{1}{T}\sum_{t}\exp\left(\widehat{m}_{f,t,c} - \log\sum_{c'}\exp\widehat{m}_{f,t,c'}\right).$$
(7)

The above loss function achieves a similar effect to the numerical case, which can also be considered as learning the loss attenuation (Kendall and Gal, 2017). When the model assigns a high logit value m_c to the observed class c, and the noise value σ_c is low, the loss approaches zero — which is the desired outcome. Overall, the total training loss is the sum of the regression and classification losses across all features, given as:

$$\mathcal{L} = \sum_{f \in \mathcal{F}_n} \mathcal{L}_f^{reg} + \lambda \sum_{f \in \mathcal{F}_c} \mathcal{L}_f^{cls} , \qquad (8)$$

where λ is a hyper-parameter designated to balance the scale of regression and classification losses. In our model, we set $\lambda = 1$ for simplicity.

3.4 UNCERTAINTY-INCORPORATED ANOMALY SCORING

286 We compute event-level anomaly scores by aggregating deviations between the predicted and the 287 observed values of all features, incorporating both epistemic and aleatoric uncertainties into the 288 computation. For each test event on a given day, we first create a sequence of events spanning 289 a 3-day window, including the previous and next day as context. We mask the event of interest 290 and use the pre-trained UIFORMER to predict the masked features. For numerical features, we 291 calculate the deviation as: $\Delta_f = |y_f - \hat{y}_f|$ with y_f and \hat{y}_f depicting the observed and predicted values, respectively. For categorical features, the deviation is computed as: $\Delta_f = 1 - \hat{p}_f$ where \hat{p}_f is 292 the predicted probability of the observed class. UIFORMER learns the inherent patterns of human 293 behaviors using only normal data during training. Intuitively, higher deviations from these patterns suggest that an event is less consistent with learned behaviors and thus, more anomalous. Then, we 295 can compute an anomaly score (AS) for a target event e from the deviations alone as: 296

$$AS(e) = agg\left(pth\left(\Delta_{f}\right) \mid f \in \mathcal{F}_{n} \cup \mathcal{F}_{c}\right) , \qquad (9)$$

298 where $agg(\cdot)$ denotes the aggregation function (e.g., mean or max) applied over Δ_f 's across all 299 features \mathcal{F} . The percentile function pth(·) is used to standardize deviations across features of different 300 scales, effectively capturing the relative ranking of the deviation for each feature. This helps ensure 301 that features with smaller absolute deviations are not undervalued compared to those with larger 302 deviations. However, using deviations alone to detect anomalies is insufficient due to the diverse 303 nature of human mobility. Some agents may deviate from the general population's behavior but are 304 not necessarily anomalous. For instance, a (stochastic) agent with a tendency to explore different locations frequently (i.e., with high data uncertainty) may have higher deviations naturally, which does 305 not imply an anomaly. To account for such cases, we propose to incorporate uncertainty estimates 306 into the anomaly score. By scaling down deviations by the uncertainties, we can prevent inflating the 307 anomaly scores of agents whose behaviors, though atypical, are still consistent with natural variability. 308 As such, we compute an uncertainty-incorporated anomaly score (UI-AS) for a target event e as: 309

310 311

312

313 314

315 316

273

274

280 281

284

285

297

$$\text{UI-AS}(e) = \arg\left(\text{pth}\left(\frac{\Delta_f}{1 + \alpha_f + \beta_f}\right) \mid f \in \mathcal{F}_n \cup \mathcal{F}_c\right) , \qquad (10)$$

where $\alpha_f + \beta_f$ denotes the total of the aleatoric and epistemic uncertainties for feature f.

4 EXPERIMENTS

We conduct extensive experiments to investigate the following research questions (RQs):
RQ1: What is the performance of UIFORMER in the masked prediction task?
RQ2: Can our model outperform previous approaches in the anomaly detection task?
RQ3: Does UIFORMER have the ability to accurately capture aleatoric and epistemic uncertainty?
RQ4: How does each uncertainty component contribute to UIFORMER?
Datasets. We utilize two human activity datasets, namely SimLA and NUMOSIM. SimLA is an expert-

simulated dataset that contains human mobility data in greater Los Angeles area. NUMOSIM (Stanford et al., 2024) is a publicly-available synthetic human mobility dataset, designed to benchmark anomaly

detection techniques for mobility data. Both datasets are obtained by training deep learning models
 on real mobility data and simulating realistic human mobility patterns, incorporating both normal and
 anomalous behaviors. The statistics of the two datasets are shown in Appendix Table 5.

We split the data into train/validation/test temporally by date, respectively spanning 3/1/2 weeks. During training, we randomly mask events in each sequence using a specified mask ratio. For validation and test data, we mask only one event per sequence, aligning with how we use the pre-trained model for anomaly detection.

331 332

333

4.1 RQ1: MASKED PREDICTION RESULTS

334 Settings. For baselines, we include classical deep models MLP (Murtagh, 1991), LSTM (Hochreiter 335 and Schmidhuber, 1997), the original Transformer (Vaswani et al., 2023), as well as Dual-Tr; a variant of UIFORMER that removes the uncertainty estimation mechanism and only keeps the dual Trans-336 former architecture (w/ cross-feature and cross-event attention). In addition, since UIFORMER can 337 estimate the uncertainty of its prediction, we also report its conformal prediction performance where 338 the top-5% most uncertain test samples are not evaluated, denoted by UIFORMER (5%). We report 339 MAE and MAPE to evaluate the prediction performance of numerical features and use accuracy 340 (ACC) for the categorical features. Higher ACC and lower MAE and MAPE indicate better perfor-341 mance. The models are trained on 1 GPU of NVIDIA RTX A6000. We search the hyperparameters 342 in a given model space for all deep models, and each model's best performance is reported in the 343 experiments. Detailed experiment settings and model configurations can be found in Appendix C. 344

Table 1: Masked prediction results. Uncertainty-aware UIFORMER outperforms the baselines, while allowing conformal prediction. Abbr: ST; start time (min), SD; stay duration (min), x, y; lat/lon (km).

| | 1 | | | | SimL | A. | | | | | | | | NUMO | SIM | | | |
|---------------|------|-------|--------|--------------|----------|--------|--------|--------|--------------|-------------|----------|--------|--------|--------|--------------|--------|--------|-------|
| Method | MAE | | | | MAPE (%) | | ACC(%) | | MAE | | MAPE (%) | | | ACC(%) | | | | |
| | х | У | ST | SD | х | У | ST | SD | POI | х | У | ST | SD | х | У | ST | SD | POI |
| MLP | 9.73 | 10.16 | 208.07 | 420.92 | 188.13 | 500.59 | 43.28 | 569.01 | 45.61 | 15.97 | 15.74 | 207.64 | 396.58 | 416.55 | 361.58 | 339.15 | 572.69 | 56.08 |
| LSTM | 4.73 | 4.63 | 19.85 | 322.62 | 204.73 | 195.76 | 18.25 | 420.76 | 63.32 | 6.92 | 6.40 | 34.41 | 343.82 | 265.52 | 139.71 | 212.30 | 547.02 | 68.29 |
| Transformer | 2.67 | 2.65 | 26.85 | 77.12 | 132.33 | 115.35 | 17.95 | 66.82 | 73.49 | 4.08 | 3.45 | 21.57 | 71.33 | 143.81 | <u>72.15</u> | 15.19 | 68.51 | 75.90 |
| Dual-Tr | 2.71 | 2.63 | 20.62 | 76.53 | 131.72 | 117.29 | 16.41 | 51.95 | <u>73.74</u> | <u>3.63</u> | 3.48 | 21.69 | 80.08 | 134.32 | 71.34 | 17.36 | 60.17 | 76.12 |
| UIFORMER | 2.44 | 2.29 | 11.42 | <u>35.77</u> | 109.44 | 102.47 | 15.86 | 20.93 | 72.97 | 3.68 | 3.60 | 26.18 | 45.25 | 121.33 | 84.01 | 236.47 | 28.87 | 74.33 |
| UIFORMER (5%) | 2.05 | 1.91 | 8.65 | 21.98 | 99.39 | 89.45 | 1.18 | 20.66 | 75.58 | 3.31 | 3.21 | 8.30 | 33.61 | 109.94 | 74.45 | 1.23 | 28.86 | 76.95 |

353 354 355

356

357

358

359

360

361

362

363

345

> **Results.** Table 1 shows the masked prediction results (here and after, we bold the **best** performance and underline the <u>second</u> best). UIFORMER delivers the best performance across both datasets and for all features compared with the baselines. In SimLA, UIFORMER achieves the lowest MAE for start time (ST) and stay duration (SD) with values of 11.42 and 35.77 minutes, respectively, outperforming the powerful sequence model, Transformer, which yields 26.85 for ST and 77.12 for SD. Compared to Dual-Tr, building uncertainty into UIFORMER leads to significant improvements in prediction performance; in SimLA, UIFORMER reduces MAE for ST from 20.62 to 11.42 minutes, and MAPE for SD from 51.95% to 21.93%, underscoring the significant impact of modeling both epistemic and aleatoric uncertainty in achieving more accurate and robust predictions. Lastly, UIFORMER (5%) further improves over UIFORMER, which demonstrates UIFORMER's ability to identify highly uncertain samples and that it can be quite accurate on its certain predictions.

364 365 366

4.2 RQ2: ANOMALY DETECTION RESULTS

367 Settings. There are only a few related public achievements for human mobility anomaly detection, 368 thus it is difficult to find state-of-the-art models for direct comparison. For a comprehensive evaluation, 369 we implement some baselines as well as extract core components from related literature in trajectory 370 anomaly detection, including Spatial-Temporal Outlier Detector (STOD) (Cruz and Barbosa, 2020), 371 RioBus (Bessa et al., 2016), a self-supervised (SS) MLP, LSTM, and Transformer. STOD and RioBus 372 are designed for bus trajectories to predict the Bus ID from the sequence of GPS coordinates which we 373 repurpose for human trajectories. SS-MLP is trained to distinguish real events from fake ones that are 374 synthetically-constructed via random alterations. Further details on baselines are given in Appendix 375 D. We report two common metrics for evaluation: AUROC (Area Under ROC curve) and AUPR (Area Under Precision-Recall curve). These are frequently used for anomaly detection as quantities 376 of ranking quality. As the percentage of anomalous events is very small on the original datasets, we 377 construct two additional one with around 5% anomalies by including agents with anomalous events

378 and randomly sampling normal agents without any anomalous events. Dataset statistics for anomaly 379 detection can be found in Appendix Table 6. 380

| 382 | | | | | | | | · · · · · · · · · · · · · · · · · · · | | | 0 | ¢υ | | |
|-----|-------------|--------------|--------------|--------------|-----------|---------|--------------|---------------------------------------|--------------|---------|-----------|--------------|--------------|--|
| 282 | Mathod | | Even | t-level An | omaly De | tection | | Agent-level Anomaly Detection | | | | | | |
| 303 | Wiethou | SimLA (| (small) | NUMOSI | M (small) | NUMOSI | M (20K) | SimLA (| small) | NUMOSIN | A (small) | NUMOSI | M (20K) | |
| 384 | | AUROC | AUPR | AUROC | AUPR | AUROC | AUPR | AUROC | AUPR | AUROC | AUPR | AUROC | AUPR | |
| 385 | STOD | 0.577 | 0.073 | 0.571 | 0.071 | 0.584 | 0.003 | 0.506 | 0.655 | 0.511 | 0.468 | 0.519 | 0.020 | |
| 386 | RioBus | 0.503 | 0.044 | 0.530 | 0.047 | 0.532 | 0.002 | 0.488 | 0.632 | 0.510 | 0.453 | 0.508 | 0.019 | |
| 387 | SS-MLP | 0.596 | 0.172 | 0.630 | 0.144 | 0.634 | 0.008 | 0.653 | 0.777 | 0.697 | 0.638 | 0.703 | 0.053 | |
| 001 | LSTM | <u>0.755</u> | 0.182 | <u>0.637</u> | 0.122 | 0.642 | 0.020 | 0.631 | 0.706 | 0.599 | 0.526 | 0.658 | 0.062 | |
| 388 | Transformer | 0.723 | <u>0.232</u> | 0.635 | 0.132 | 0.636 | <u>0.033</u> | <u>0.679</u> | <u>0.785</u> | 0.625 | 0.525 | <u>0.717</u> | <u>0.095</u> | |
| 389 | UIFORMER | 0.795 | 0.329 | 0.694 | 0.146 | 0.670 | 0.034 | 0.746 | 0.841 | 0.655 | 0.572 | 0.727 | 0.114 | |
| 390 | | | | | | | | | | | | | | |

Table 2: Anomaly detection results at the event-level (left) and the agent-level (right).

391 **Results.** Table 2 presents the anomaly detection results for both event-level and agent-level anomalies. UIFORMER consistently outperforms the baselines across all datasets in detecting anomalous events 392 and agents. At the event level, particularly on SimLA, UIFORMER achieves a significantly large margin 393 against the baselines, with the highest AUROC of 0.795 and AUPR of 0.329 ($8 \times$ lift over the 0.041 394 base rate). This highlights the effectiveness of incorporating both aleatoric and epistemic uncertainty, 395 which substantially enhances event-level detection. UIFORMER also performs well for agent-level 396 anomaly detection (w/ maximum anomaly scores across all events per agent). On the NUMOSIM 397 (20K agents) dataset, it achieves an AUROC of 0.727 and an AUPR of 0.114 (57× lift over the 398 0.002 base rate), demonstrating its ability to effectively identify rare anomalous agents within a large 399 population. By incorporating uncertainty into the anomaly score computation, UIFORMER prevents 400 inflated scores for agents with exploratory but not necessarily anomalous behavior, thereby reducing 401 false positives. As a result, it more effectively detects true anomalies that deviate from otherwise predictable behavior without incorrectly flagging exploratory behavior. 402

403 404

405

381

RQ3: ANALYSIS OF UNCERTAINTY ESTIMATION 4.3

406 Next we scrutinize the quality of the estimated uncertainty scores. To this end, we analyze the 407 relationship between prediction accuracy as a function of total uncertainty, as well as how the 408 estimated aleatoric and epistemic uncertainty scores vary by the amount of training data using SimLA.

409 Relationship between uncertainty and MAE/accuracy. We first sort the test samples by the 410 total AU+EU uncertainty score (i.e., $\sum_{f \in \mathcal{F}} \alpha_f + \beta_f$), and then gradually remove test samples with 411 uncertainty larger than a certain percentile threshold while recording the prediction performance of 412 UIFORMER on the rest. As shown in Figure 4, UIFORMER's performance gradually improves as 413 we exclude test samples with large total uncertainty. This means that the uncertainty estimates are 414 well aligned with prediction performance, as all MAE (Accuracy) curves follow a monotonically 415 increasing (decreasing) trend for numerical (categorical) features. Notably, for stay duration (SD) prediction (middle), MAE increases sharply from around 20 minutes to 35 minutes due to the top 416 5% most uncertain samples. As for the POI prediction (right), around 50% of the most certain 417 samples can yield a quite promising performance that achieves near 100% accuracy, which gradually 418 drops with increasing uncertainty. Interestingly, AU-only UIFORMER (in blue) appears to be more 419 reflective of these expected trends for x and SD, while it is the EU-only variant (in purple) for POI, 420 demonstrating the complementary benefit of estimating these different uncertainty types. 421

422 423



Figure 4: Relationship btwn. MAE (accuracy) & uncertainty follow an increasing (decreasing) trend.

| Train | Test | 2 | ĸ | У | 1 | Start | Time | Stay D | uration | POI | |
|--------|--------|-----------------------------|---------------------|------------------------------|--------------------------|---------------------|------------------------------|---------------------|-------------------------|-----------------------------|------|
| | | $\mathrm{AU} 	imes 10^{-3}$ | $EU \times 10^{-5}$ | $\mathrm{AU}\times\!10^{-3}$ | ${\rm EU}\times 10^{-5}$ | AU $\times 10^{-3}$ | $\mathrm{EU}\times\!10^{-5}$ | $AU \times 10^{-3}$ | $ $ EU $\times 10^{-5}$ | $\mathrm{AU} 	imes 10^{-3}$ | EU |
| 1-week | 2-week | 1.30 | 2.37 | 2.40 | 4.00 | 0.85 | 15.49 | 0.10 | 0.48 | 17.11 | 1.01 |
| 2-week | 2-week | 1.44 | 2.05 | 2.45 | 3.12 | 0.61 | 7.29 | 0.13 | 0.25 | 14.34 | 0.98 |
| 3-week | 2-week | 1.36 | 1.23 | 2.51 | 2.00 | 0.59 | 4.08 | 0.11 | 0.14 | 8.25 | 0.94 |

432 Table 3: Aleatoric (AU) and epistemic uncertainties (EU) under different training data size settings.

438 Relationship between uncertainty and amount of training data. Next we investigate how aleatoric 439 and epistemic uncertainty estimates change with varying amount of training data, under three different 440 configurations with 1/2/3 weeks of training data and 2 weeks of test data on SimLA. Table 3 reports the average aleatoric uncertainty (AU) and epistemic uncertainty (EU) across test samples for each 442 feature. We observe that (i) AU remains almost constant as training data size increases, which 443 empirically shows that AU accounts for the inherent randomness/stochasticity in the data that is not a matter of the amount of training data, and that (*ii*) EU decreases noticably with increasing data size, 444 suggesting that it can be reduced with sufficient training data. 445

446 Relationship between EU and accuracy of each 447 **POI class**. From the results in Figure 5, we have 448 the following observations: (i) As before, there is 449 an inverse trend where accuracy decreases as epistemic uncertainty (EU) increases. This suggests that 450 as the model becomes less certain (higher EU), its 451 performance in predicting the correct POI class dete-452 riorates. (ii) A higher frequency of POI classes (such 453 as home and office buildings) generally leads to lower 454 EU. Since EU aligns with amount of training data as 455 showcased earlier, the more frequent POI classes, for 456 which the model is trained on more observations, as-457 sociate with lower EU and higher prediction accuracy. 458 (iii) A lower frequency does not always mean high 459 EU however. For example, POI:education in Figure 5 460 does not have a frequency as high as POI:home and POI:store with similar accuracies, but it has a rela-461 tively low estimated EU possibly because people's 462 visits to such POI are highly patterned. 463



Figure 5: The relationship between epistemic uncertainty (EU) and accuracy of each POI class. Size of each point/POI is proportional to its frequency in the dataset. We only annotate a few POI labels for better illustration.

464 465

466

441

4.4 RQ4: ABLATION STUDY

As uncertainty is a key component of our anomaly detection model, we conduct ablation studies on 467 its role from both the training and inference perspectives. 468

469 Uncertainty from the training perspective. To study the role of aleatoric versus epistemic un-470 certainty estimation, we consider three variants of UIFORMER: (i) UIFORMER-AU, which only implements the aleatoric uncertainty while keeping the model backbone the same as UIFORMER, (*ii*) 471 UIFORMER-EU, which only models epistemic uncertainty; and (*iii*) UIFORMER-none, which only 472 has the bare Dual Transformer and does not model epistemic or aleatoric uncertainty. 473

474 For comparison, we plot the relationship between uncertainty and performance in Figure 4. Firstly, 475 the inclusion of uncertainty modeling can significantly enhance the predictive accuracy of numerical 476 features. Moreover, UIFORMER-AU model significantly outperforms UIFORMER-EU in regression 477 tasks. This suggests that aleatoric uncertainty captures essential variance within the dataset that epistemic uncertainty cannot address effectively. However, when considering classification tasks, 478 such as POI and DOW, modeling aleatorical uncertainty does not provide a significant performance 479 boost. UIFORMER slightly outperforms UIFORMER-AU, which suggests that modeling both aleatoric 480 uncertainty and epistemic uncertainty can further improve the performance. 481

482 **Uncertainty from the inference/anomaly scoring perspective.** In this section, we investigate the impact of different anomaly scoring functions on detection performance. 483

484 Table 4 presents the results of UIFORMER's event-level anomaly detection using three different 485 scores per feature: Δ_f , pth (Δ_f) , and pth $\left(\frac{\Delta_f}{1+\alpha_f+\beta_f}\right)$. Here, Δ_f represents the feature delta error,

486 $pth(\cdot)$ denotes the percentile function, and α_f and β_f represent aleatoric and epistemic uncertainties, 487 respectively. The feature anomaly scores were aggregated by taking the maximum across features. 488

From the comparison between Δ_f and 489 $pth(\Delta_f)$, we observe a significant im-490 provement in performance when using per-491 centiles. This shows that the percentile 492 function and thus using the relative rank-493 ing of errors is beneficial when features 494 are variable in scale, ensuring that features 495 with smaller error scales are not overshad-

Table 4: Comparison of different anomaly scores

| Function | SimLA (| small) | NUMOSI | M (small) | NUMOSI | M (20K) |
|--|--------------|--------------|--------|--------------|--------------|--------------|
| Tunction | AUROC | AUPR | AUROC | AUPR | AUROC | AUPR |
| Δ_f | 0.666 | 0.200 | 0.553 | 0.062 | 0.556 | 0.003 |
| $pth(\Delta_f)$ | <u>0.789</u> | <u>0.295</u> | 0.694 | <u>0.143</u> | <u>0.699</u> | <u>0.031</u> |
| $\operatorname{pth}\left(\frac{\Delta_f}{1+\alpha_f+\beta_f}\right)$ | 0.795 | <u>0.329</u> | 0.694 | 0.147 | 0.700 | 0.034 |

496 owed by those with larger scales, leading to more balanced and effective anomaly detection. Furthermore, comparing $pth(\Delta_f)$ to $pth\left(\frac{\Delta_f}{1+\alpha_f+\beta_f}\right)$, we see that incorporating uncertainties into the 497 498 anomaly scoring further enhances detection performance. This shows that accounting for total 499 uncertainty improves the model's ability to identify the true anomalies more effectively. 500

5 **RELATED WORK** 502

501

Human Mobility Modeling. Methods in this field can be broadly categorized into traditional 504 statistical models and the more recent deep learning approaches. Traditional statistical approaches 505 typically rely on specific functional forms such as Poisson or Hawkes processes to predict event arrival times (Hawkes, 1971; Daley and Vere-Jones, 2008; Ogata, 1998), which struggle to capture 506 the intricate spatiotemporal patterns present in human mobility data. Other statistical methods 507 like Markov Chains, use transition matrices to predict future locations (Chen et al., 2014; Gambs 508 et al., 2012; Cheng et al., 2013; Monreale et al., 2009) yet they fall short in modeling long-term 509 dependencies. With the rise of deep learning, models such as RNNs and LSTMs have been used to 510 learn the complex transition patterns (Gao et al., 2017; Song et al., 2016; Du et al., 2016), leading 511 to better performance in the next location and trajectory prediction. Most recently, there has been 512 a growing trend of using Transformer-based models for human mobility and trajectory data (Wan 513 et al., 2021; Feng et al., 2018; Xue et al., 2021; Abideen et al., 2021; Wu et al., 2020; Wang and 514 Osaragi, 2024), mainly inspired by its immense success in modeling inherently sequential data in NLP. 515 However, most Transformer-based models overlook the underlying uncertainty in the data, especially as it pertains to human behavior over time. Specifically, most prior work focus on pretraining solely 516 based on the masked reconstruction loss, without uncertainty estimation of the prediction. This 517 motivates us to develop a new uncertainty-aware model to fill the gap, toward more effectively 518 capturing human behavior that is often dependent on time and context. 519

520 Uncertainty Estimation in Deep Neural Networks. Uncertainty estimation has been studied in 521 the field of NLP (Gal and Ghahramani, 2016b; Xiao and Wang, 2019) and CV (Kendall et al., 2016; Huang et al., 2018), which shows great promise in reflecting the underlying uncertainty in outcomes 522 and thus improving task performance. One common approach is the Monte Carlo (MC) Dropout (Gal 523 and Ghahramani, 2016a), which estimates uncertainty by applying dropout during inference, thereby 524 improving model robustness. Others employ deep ensembles for model uncertainty estimation 525 Lakshminarayanan et al. (2017). Kendall and Gal (2017) propose to learn aleatoric uncertainty 526 through loss attenuation while modeling epistemic uncertainty using MC Dropout in both regression 527 and classification tasks for CV. Mobiny et al. (2021) drop individual weights of the network during 528 training and inference to measure uncertainty robustly. Motivated by the success of uncertainty 529 estimation in these domains, we explore the potential benefit of uncertainty estimation in modeling 530 human mobility, which forms the basis of our proposed model.

531 532

533

CONCLUSION 6

534 We introduced UIFORMER for human mobility modeling and anomalous behavior detection. UIFORMER is equipped with both data and model uncertainty estimation to account for the stochas-536 ticity inherent to human nature as well as the data scarcity in sufficiently observing complex human behavior. While its dual-attention mechanism captures dependencies among features and events, uncertainty estimation lends itself to more robust training and more nuanced anomaly scoring. Exper-538 iments on two benchmark datasets showed that UIFORMER outperforms existing baselines, while ablation analyses underscored the benefit of uncertainty-aware forecasting and anomaly detection.

540 REFERENCES

558

559

561

562

566

567

568 569

570

571

572

573

574

580

581

582

583

584

590

Zain Ul Abideen, Heli Sun, Zhou Yang, Rana Zeeshan Ahmad, Adnan Iftekhar, and Amir Ali. 2021.
 Deep Wide Spatial-Temporal Based Transformer Networks Modeling for the Next Destination
 According to the Taxi Driver Behavior Prediction. *Applied Sciences* 11, 1 (2021). https:
 //doi.org/10.3390/app11010017

- Hugo Barbosa, Marc Barthelemy, Gourab Ghoshal, Charlotte R James, Maxime Lenormand, Thomas Louail, Ronaldo Menezes, José J Ramasco, Filippo Simini, and Marcello Tomasini. 2018. Human mobility: Models and applications. *Physics Reports* 734 (2018), 1–74.
- Thomas Bayes. 1763. An Essay towards Solving a Problem in the Doctrine of Chances. *Philosophical Transactions of the Royal Society of London* 53 (1763), 370–418. https://doi.org/10. 1098/rstl.1763.0053 Communicated by Mr. Price, in a letter to John Canton, A. M. F. R.
 S..
- Aline Bessa, Fernando de Mesentier Silva, Rodrigo Frassetto Nogueira, Enrico Bertini, and Juliana Freire. 2016. RioBusData: Outlier Detection in Bus Routes of Rio de Janeiro. arXiv:1601.06128 [cs.HC] https://arxiv.org/abs/1601.06128
 - Meng Chen, Yang Liu, and Xiaohui Yu. 2014. NLPMM: A Next Location Predictor with Markov Modeling. In Advances in Knowledge Discovery and Data Mining, Vincent S. Tseng, Tu Bao Ho, Zhi-Hua Zhou, Arbee L. P. Chen, and Hung-Yu Kao (Eds.). Springer International Publishing, Cham, 186–197.
- Chen Cheng, Haiqin Yang, Michael R. Lyu, and Irwin King. 2013. Where you like to go next:
 successive point-of-interest recommendation. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence* (Beijing, China) (*IJCAI '13*). AAAI Press, 2605–2611.
 - Riccardo Corrias, Martin Gjoreski, and Marc Langheinrich. 2023. Exploring transformer and graph convolutional networks for human mobility modeling. *Sensors* 23, 10 (2023), 4803.
 - Michael Cruz and Luciano Barbosa. 2020. Learning GPS Point Representations to Detect Anomalous Bus Trajectories. *IEEE Access* 8 (2020), 229006–229017. https://doi.org/10.1109/ ACCESS.2020.3046912
 - D. J. Daley and David Vere-Jones. 2008. An Introduction to the Theory of Point Processes: Volume II (2nd ed.). Springer.
- 575
 576
 576
 576
 577
 578
 578
 579
 579
 574
 575
 575
 576
 576
 577
 578
 579
 578
 579
 579
 574
 575
 575
 576
 576
 576
 577
 578
 579
 578
 579
 578
 579
 579
 578
 579
 578
 579
 579
 578
 579
 578
 579
 579
 579
 579
 570
 570
 571
 572
 573
 574
 574
 574
 575
 574
 575
 576
 576
 576
 576
 576
 576
 576
 578
 579
 579
 579
 570
 570
 571
 571
 572
 574
 574
 574
 575
 574
 575
 574
 575
 576
 576
 576
 576
 576
 576
 576
 576
 576
 576
 576
 576
 576
 576
 576
 576
 576
 576
 576
 576
 576
 576
 576
 576
 576
 576
 576
 576
 576
 576
 576
 576
 576
 576
 576
 576
 576
 576
 576
 576
 576
 576
 576
 576
 576
 576
 576
 576
 576
 576
 576
 576
 576
 576
 576
 576
 576
 576
 576
 576
 576
 576
 576
 576
 576
 576
 576
 576
 576
 576
 576
 576
 576
 576
 576
 576
 - Jie Feng, Yong Li, Chao Zhang, Funing Sun, Fanchao Meng, Ang Guo, and Depeng Jin. 2018. DeepMove: Predicting Human Mobility with Attentional Recurrent Networks. In *Proceedings of the 2018 World Wide Web Conference* (Lyon, France) (*WWW '18*). International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 1459–1468. https: //doi.org/10.1145/3178876.3186058
- Yarin Gal and Zoubin Ghahramani. 2016a. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In *Proceedings of The 33rd International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 48)*, Maria Florina Balcan and Kilian Q. Weinberger (Eds.). PMLR, New York, New York, USA, 1050–1059. https: //proceedings.mlr.press/v48/gal16.html
- Yarin Gal and Zoubin Ghahramani. 2016b. A theoretically grounded application of dropout in recur rent neural networks. In *Proceedings of the 30th International Conference on Neural Information Processing Systems* (Barcelona, Spain) (*NIPS'16*). Curran Associates Inc., Red Hook, NY, USA, 1027–1035.

| 594 595 596 597 | Sébastien Gambs, Marc-Olivier Killijian, and Miguel Núñez del Prado Cortez. 2012. Next place prediction using mobility Markov chains. In <i>Proceedings of the First Workshop on Measurement</i> , <i>Privacy, and Mobility</i> (Bern, Switzerland) (<i>MPM '12</i>). Association for Computing Machinery, New York, NY, USA, Article 3, 6 pages. https://doi.org/10.1145/2181196.2181199 |
|---------------------------------|--|
| 598 599 600 601 | Qiang Gao, Fan Zhou, Kunpeng Zhang, Goce Trajcevski, Xucheng Luo, and Fengli Zhang. 2017. Iden- tifying human mobility via trajectory embeddings. In <i>Proceedings of the 26th International Joint</i> <i>Conference on Artificial Intelligence</i> (Melbourne, Australia) (<i>IJCAI'17</i>). AAAI Press, 1689–1695. |
| 602 603 604 | Alan G. Hawkes. 1971. Spectra of some self-exciting and mutually exciting point processes. Biometrika 58 (1971), 83–90. https://api.semanticscholar.org/CorpusID: 14122089 |
| 606 607 | Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. Neural Comput. 9, 8 (Nov. 1997), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735 |
| 608 609 610 611 612 | Po-Yu Huang, Wan-Ting Hsu, Chun-Yueh Chiu, Ting-Fan Wu, and Min Sun. 2018. Efficient Uncertainty Estimation for Semantic Segmentation in Videos. In Computer Vision – ECCV 2018: 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part I (Munich, Germany). Springer-Verlag, Berlin, Heidelberg, 536–552. https://doi.org/10.1007/ 978-3-030-01246-5_32 |
| 613 614 615 | Alex Kendall. 2018. <i>Geometry and Uncertainty in Deep Learning for Computer Vision</i> . Ph.D. Dissertation. University of Cambridge. |
| 616 617 618 | Alex Kendall, Vijay Badrinarayanan, and Roberto Cipolla. 2016. Bayesian SegNet: Model Uncertainty in Deep Convolutional Encoder-Decoder Architectures for Scene Understanding. arXiv:1511.02680 [cs.CV] https://arxiv.org/abs/1511.02680 |
| 619 620 621 622 623 | Alex Kendall and Yarin Gal. 2017. What uncertainties do we need in Bayesian deep learning for computer vision?. In <i>Proceedings of the 31st International Conference on Neural Information Processing Systems</i> (Long Beach, California, USA) (<i>NIPS'17</i>). Curran Associates Inc., Red Hook, NY, USA, 5580–5590. |
| 624 625 626 | Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. 2017. Simple and scalable predictive uncertainty estimation using deep ensembles. <i>Advances in neural information processing systems</i> 30 (2017). |
| 627 628 629 630 | Emaad Manzoor and Leman Akoglu. 2017. Rush! targeted time-limited coupons via purchase forecasts. In <i>Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining</i> . 1923–1931. |
| 631 632 633 | Sandro Meloni, Nicola Perra, Alex Arenas, Sergio Gómez, Yamir Moreno, and Alessandro Vespignani. 2011. Modeling human mobility responses to the large-scale spreading of infectious diseases. <i>Scientific reports</i> 1, 1 (2011), 62. |
| 634 635 636 637 | Bryan Minor, Janardhan Rao Doppa, and Diane J Cook. 2015. Data-driven activity prediction: Algorithms, evaluation methodology, and applications. In <i>Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining</i> . 805–814. |
| 638 639 640 | Amir Mobiny, Pengyu Yuan, Soumyabrata K. Moulik, S. M. Iftekhar Uddin Abedin, and Hung T. Nguyen. 2021. DropConnect is effective in modeling uncertainty of Bayesian deep networks. Scientific Reports 11 (2021), 5458. https://doi.org/10.1038/s41598-021-84854-x |
| 641 642 643 644 645 | Anna Monreale, Fabio Pinelli, Roberto Trasarti, and Fosca Giannotti. 2009. WhereNext: a location predictor on trajectory pattern mining. In <i>Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining</i> (Paris, France) (<i>KDD '09</i>). Association for Computing Machinery, New York, NY, USA, 637–646. https://doi.org/10.1145/1557019.1557091 |
| 040 | Eight Mustock 1001 Multilator reportions for close footion and represeive Neuroperation 2.5 |

Fionn Murtagh. 1991. Multilayer perceptrons for classification and regression. *Neurocomputing* 2, 5 (1991), 183–197. https://doi.org/10.1016/0925-2312(91)90023-5

- Yosihiko Ogata. 1998. Space-Time Point-Process Models for Earthquake Occurrences. Annals of the Institute of Statistical Mathematics 50 (1998), 379–402. https://doi.org/10.1023/A: 1003403601725
- Kuan Song, Hiroshi Kanasugi, and Ryosuke Shibasaki. 2016. Deeptransport: prediction and simulation of human mobility and transportation mode at a citywide level. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence* (New York, New York, USA) (*IJCAI'16*). AAAI Press, 2618–2624.
- Chris Stanford, Suman Adari, Xishun Liao, Yueshuai He, Qinhua Jiang, Chenchen Kuai, Jiaqi
 Ma, Emmanuel Tung, Yinlong Qian, Lingyi Zhao, Zihao Zhou, Zeeshan Rasheed, and Khurram
 Shafique. 2024. NUMOSIM: A Synthetic Mobility Dataset with Anomaly Detection Benchmarks.
 arXiv:2409.03024 [cs.LG] https://arxiv.org/abs/2409.03024
- Timothy Truong Jr and Tristan Bepler. 2023. Poet: A generative model of protein families as sequences-of-sequences. *Advances in Neural Information Processing Systems* 36 (2023), 77379–77415.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2023. Attention Is All You Need. arXiv:1706.03762 [cs.CL] https://arxiv.org/abs/1706.03762
- Huaiyu Wan, Yan Lin, Shengnan Guo, and Youfang Lin. 2021. Pre-training time-aware location
 embeddings from spatial-temporal trajectories. *IEEE Transactions on Knowledge and Data Engineering* 34, 11 (2021), 5510–5523.
- Weiying Wang and Toshihiro Osaragi. 2024. Learning Daily Human Mobility with a Transformer-Based Model. *ISPRS International Journal of Geo-Information* 13, 2 (2024). https://doi.org/10.3390/ijgi13020035
- Kian Wu, Chao Huang, Chuxu Zhang, and Nitesh V. Chawla. 2020. Hierarchically Structured Transformer Networks for Fine-Grained Spatial Event Forecasting. In *Proceedings of The Web Conference 2020* (Taipei, Taiwan) (*WWW '20*). Association for Computing Machinery, New York, NY, USA, 2320–2330. https://doi.org/10.1145/3366423.3380296
- Yijun Xiao and William Yang Wang. 2019. Quantifying uncertainties in natural language processing tasks. In Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence (Honolulu, Hawaii, USA) (AAAI'19/IAAI'19/EAAI'19). AAAI Press, Article 899, 8 pages. https://doi.org/10. 1609/aaai.v33i01.33017322
- Hao Xue, Flora Salim, Yongli Ren, and Nuria Oliver. 2021. MobTCast: Leveraging Auxiliary Trajectory Forecasting for Human Mobility Prediction. In *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (Eds.), Vol. 34.
 Curran Associates, Inc., 30380–30391. https://proceedings.neurips.cc/paper_files/paper/2021/file/fecf2c550171d3195c879d115440ae45-Paper.pdf
- Hao Xue, Bhanu Prakash Voutharoja, and Flora D Salim. 2022. Leveraging language foundation
 models for human mobility forecasting. In *Proceedings of the 30th International Conference on Advances in Geographic Information Systems*. 1–9.
- Zihao Zhou, Xingyi Yang, Ryan Rossi, Handong Zhao, and Rose Yu. 2022. Neural point process
 for learning spatiotemporal event dynamics. In *Learning for Dynamics and Control Conference*.
 PMLR, 777–789.
- 696

684

651

670

- 697
- 030

699 700

100

DATA STATISTICS А

| Table 5: Datase | et statistics | • |
|----------------------------|---------------|-----------|
| | SimLA | NUMOSIM |
| #Agents | 20,000 | 20,000 |
| #Anomalous Agents | 162 | 381 |
| #Events | 3,111,712 | 3,428,714 |
| #Anomalous Events | 892 | 3,468 |
| x range (km) | [-57, 57] | [-59, 59] |
| y range (km) | [-40, 40] | [-60, 60] |
| Avg of stay duration (min) | 509 | 455 |
| Avg of start time (min) | 850 | 776 |

714 715 716

717 718

719

720

721

722

723 724 725

702

В **DETAILS OF TRANSFORMER BLOCK**

Transformer Block, which contains two key components: a Multi-Head Self-Attention (MSA) layer and a Feed-Forward Network (FFN) layer. The MHA layer facilitates message passing between input tokens, while the FFN applies non-linear transformations to enhance feature extraction across different dimensions of the input vectors. To capture more complex interactions between tokens, multi-head attention is employed, where the attention mechanism is defined in Equation (11).

Attention
$$(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \operatorname{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{d}}\right)\boldsymbol{V},$$
 (11)

(12)

where $Q \in \mathbb{R}^{N \times D}$, $K \in \mathbb{R}^{N \times D}$, and $V \in \mathbb{R}^{N \times D}$ represent the query, key, and value matrices, 726 respectively, all projected from the same input matrix E (which have different forms in Feature-level 727 Transformer and Event-level Transformer) with different learnable weight matrix. The softmax 728 function transforms the scaled dot product into attention weights for V, and d is the dimensionality of 729 K used for scaling the inner product. Besides, more high-order mutual information can be captured by 730 stacking multiple Transformer blocks. Denote the embedding outputted by block $m \in \{1, \dots, M\}$ 731 as e_i^m , its updating process can be formulated as follows 732

 $\hat{e}_i = e_i^{m-1} + \mathrm{MSA}_i^m \left(\mathrm{head}_1^{m-1}, \dots, \mathrm{head}_{n_{\mathrm{head}}}^{m-1} \right) \\ e_i^m = \hat{e}_i + \mathrm{FFN}(\hat{e}_i).$ where n_{head} is the number of heads.

736 738

733

734

735

С DETAILED SETTINGS FOR MASKED PREDICTION

739 Baselines. The following methods are chosen as baselines: (1) MLP (Murtagh, 1991), which applies 740 multiple Linear layers to encode the event sequence, and decodes the masked features according to the embedding from the last layer. (2) LSTM (Hochreiter and Schmidhuber, 1997), a recurrent 741 neural network commonly used to model sequential data, it reads the event sequence and decodes 742 the masked event based on its hidden vector. (3) Transformer(Vaswani et al., 2023), one of the most 743 powerful sequence models nowadays, which models the correlation of different events by multi-head 744 self-attention. (4) Dual-Transformer, a variant of our proposed model, which removes the uncertainty 745 learning mechanism and only keeps the dual Transformer architecture. 746

Metrics. We consider the prediction of numerical features and categorical features as regression and 747 classification tasks, respectively. To this end, we apply MAE and MAPE to evaluate the performance 748 of regression, and Accuracy (ACC) is applied to evaluate the classification performance. The unit for 749 x, y are kilometers while the unit for start time and stay duration are minutes. 750

751 Settings. The models are trained on 1 GPU of NVIDIA RTX A6000. For all deep models, we search 752 the parameters in a given parameter space, and each model's best performance is reported in the 753 experiment. Here we report the parameter search space of our model: the mask ratio for pre-training is searched from [0.05, 0.3]. The embedding size for the transformer is searched from [32, 64, 128], 754 and the layers for the transformer are searched from [2, 3, 5]. The batch size of each epoch is searched 755 from 128 and the learning rate of Adam optimizer starts from 1e-3 with a weight decay 1e-05.

| 757 | | - | | |
|-----|---------------------------|---------------|-----------------|---------------|
| 750 | | SimLA (small) | NUMOSIM (small) | NUMOSIM (20K) |
| 700 | | | | |
| 759 | #Agents | 254 | 851 | 20,000 |
| 760 | #Anomalous Agents | 162 | 381 | 381 |
| 761 | Ratio of Anomalous Agents | 0.638 | 0.448 | 0.019 |
| 762 | #Events | 18,963 | 73,129 | 1,706,468 |
| 763 | #Anomalous Events | 892 | 3,468 | 3,468 |
| 764 | Ratio of Anomalous Events | 0.041 | 0.047 | 0.002 |
| | | | | |

Table 6: Dataset statistics for anomaly detection. Ratio of anomalies depict the base rate.

765 766

767

756

D DETAILED SETTINGS FOR ANOMALY DETECTION

768 Baselines. We selected the following baseline methods for comparison: RioBus Bessa et al. (2016), 769 Spatial-Temporal Outlier Detector (STOD) Cruz and Barbosa (2020), a self-supervised MLP (SS-770 MLP), LSTM, and Transformer. RioBus Bessa et al. (2016) is initially applied to bus trajectories and 771 employs a Convolutional Neural Network (CNN) to predict the ID of the bus based on the sequence 772 of GPS coordinates. For agent-level anomaly scores, it uses the negative probability of the correct 773 prediction of the agent (bus) ID. For the stay-point (or event) level, scores are derived from the 774 anomaly score of the 20-coordinate sub-trajectory ending at the given stay point. STOD Cruz and Barbosa (2020) is also originally applied to bus trajectories and is GRU-based to predict the bus 775 ID. Its anomaly score computation approaches are similar to RioBus. SS-MLP takes event features 776 including GPS coordinates and time as input. These features are first converted into embeddings, 777 which are then concatenated to form a unified input vector. This vector is processed by an 8-layer 778 MLP that outputs a probability indicating whether an event is real (1) or fake (0). During training, 779 fake events are generated by randomly altering real events, and the model learns to classify between real and fake using binary cross-entropy loss. This approach allows the model to distinguish between 781 normal and anomalous data. Finally, the negative of the output probability is used as the anomaly 782 score, with a higher score indicating greater anomaly likelihood.

Settings. Since the percentage of anomalous events in the entire dataset is extremely small, we construct two additional test datasets with approximately 0.05% anomalous events, named SimLA (small) and NUMOSIM (small). These datasets are constructed by including all agents with anomalous events and randomly sampling from other agents without any anomalous events. The statistics of the test datasets are in Table 6. This approach provides three test datasets with varying levels of anomaly scarcity, allowing for diverse evaluation settings.

789 790 791

E CASE STUDY

792 Here, we present two examples: one agent with high total uncertainty and another with low uncertainty. 793 Their daily trajectories in the first week of training data are shown in Figure 6. Markers of different 794 shapes and colors represent various POI types. The numerical labels next to the markers indicate the 795 order of events occurring each day. The high-uncertainty agent visits a wide variety of POI types 796 daily, without any fixed pattern in the locations visited or the order of visiting those locations. In 797 contrast, the low-uncertainty agent exhibits regular patterns on weekdays, such as traveling from 798 home to a possible workplace, and remains at home all day on Sunday during the weekend. These examples demonstrate that our model's uncertainty measure effectively captures the variability in 799 human mobility patterns. 800

801 802

803

F BROADER IMPACT

Our work proposes an unsupervised approach to modeling uncertainty in sequence data, with a specific focus on human mobility. By leveraging both aleatoric and epistemic uncertainties, our model captures the variability inherent in complex behaviors and provides a robust way to detect anomalies without relying on labeled data. This approach is highly generalizable and can be applied across various domains, such as finance (e.g., detecting credit card fraud or money laundering), online shopping (e.g., analyzing user purchasing patterns and identifying bot activity), and app user behavior analysis (e.g., detecting shifts or unusual engagement). By modeling uncertainty, the proposed



method can provide more informative anomaly scores, distinguishing between natural variability and significant deviations that require attention. This makes the model particularly useful in situations where uncertainty is expected, and traditional methods might lead to false positives.