2022 IEEE International Conference on Data Mining (ICDM) | 978-1-6654-5099-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICDM54844.2022.00144

Multi-view POI-level Cellular Trajectory Reconstruction for Digital Contact Tracing of Infectious Diseases

Dongmin Park[1], Junhyeok Kang[1], Hwanjun Song[2], Susik Yoon[3], Jae-Gil Lee[1]*
[1] KAIST, [2] Naver AI Lab, [3] University of Illinois at Urbana-Champaign

***Abstract*—Digital contact tracing is an effective solution to prevent such a pandemic, but the low adoption rate of a required mobile app hinders its effectiveness. A large collection of cellular trajectories from mobile subscribers can be an out-of-the-box solution that is free from the low adoption issue, but has been overlooked due to its low spatial resolution. In this paper, to increase the resolution of this cellular trajectory, we present a new problem that estimates the user's visited places *at the point-of-interest (POI) level*, which we call *POI-level cellular trajectory reconstruction*. We propose a novel algorithm, *Pincette*, that accomplishes more accurate POI reconstruction by leveraging various external data such as road networks and POI contexts. Specifically, *Pincette* comprises *multi-view feature extraction* and *GCN-LSTM-based POI estimation*. In the multi-view feature extraction, *Pincette* extracts three complementary features from three views: efficiency, periodicity, and popularity. In the GCN-LSTM-based POI estimation, these three views are seamlessly integrated, where spatio-temporal periodic patterns are captured by graph convolutional networks (GCNs) and an LSTM. With extensive experiments on two real data collections of two cities, we show that *Pincette* outperforms four POI estimation baselines by up to 21.20%. We believe that our work sheds light on the use of cellular trajectories for digital contact tracing. We release the source code at https://github.com/kaist-dmlab/Pincette.**

I. Introduction

A. Motivation

Digital contact tracing, the process of identifying persons who may have contact with a confirmed case through an automated tracking system [1], [2], has been actively developed to assist contact tracers (*i.e.*, public health authorities). As opposed to *manual* contact tracing, it facilitates very timely and accurate self-quarantine so that expected to lead to the sustained epidemic suppression [3]. A typical approach for digital contact tracing is to download an app on the user's mobile device which records the locations where the user visited or other people whom they spent time with, using location-finding technologies such as Bluetooth and Global Positioning System (GPS). Most notably, Google and Apple jointly created the Exposure Notifications System in May 2020 [4]. To stop the epidemic, this approach requires more than 60% uptake in the population [5], but the real adoption rate has been reported to be much lower, *e.g.*, less than 10% in the U.S., mainly due to privacy concerns. This situation highlights the importance of a complementary tracing system that can cover a wide range of users.

In this regard, *cellular network data*, which is generated by interactions between mobile devices and cell towers, has become a good alternative resource for digital contact tracing, considering that it collects the location of a large population *without* any additional efforts such as app installation. Moreover, thanks to the advent of 4G and 5G networks, its temporal resolution (*i.e.*, the amount of time needed for location updates) has been greatly improved, reaching only 90 seconds [6]. Nevertheless, the use of cellular network data is still hindered by its *low spatial resolution*, resulting from the localization errors of up to a few hundred meters in urban areas. The true location of the user is simply abstracted to that of a nearby cell tower in cellular network data.

To cope with this challenge, *cellular trajectory reconstruction* has been widely studied to infer a user's *true* locations, thereby improving the understanding of human mobility. The representative approach is to reconstruct the user's actual moving path by performing a map matching algorithm on the road network [6], [7]. However, such moving path reconstruction only supports coarse-grained contact tracing; finding two similar moving paths does *not* necessarily guarantee a contact between them. For example, if a person follows the same route that another person went through a few minutes ago, there is *no* actual contact between them, though their moving similarity is very high. To pursue *fine-grained* contact tracing of the infectious disease, the reconstruction should configure the point-of-interest (POI) where *users stay together with others* because the infectious disease spreads primarily by indoor transmission.

B. Research Problem and Pincette

In this paper, we first present a new problem, named **POI-level cellular trajectory reconstruction**, that aims to estimate truly visited POIs from a cellular trajectory. There are two main challenges for this problem: **(i)** each user has a *complex* POI visiting pattern; and **(ii)** the *coarse-grained* positioning ability of a cellular trajectory makes dozens of POI candidates. We propose a novel framework *Pincette* that leverages various external data including road networks, public transport routes, and POI contexts. To precisely capture the complex POI visiting pattern and conduct fine-grained POI estimation, as illustrated in Figure 1, *Pincette* leverages the complementary information between the *three* multi-view features:

∗ Jae-Gil Lee is the corresponding author.

978-1-6654-5099-7/22/$31.00 ©2022 IEEE
DOI 10.1109/ICDM54844.2022.00144

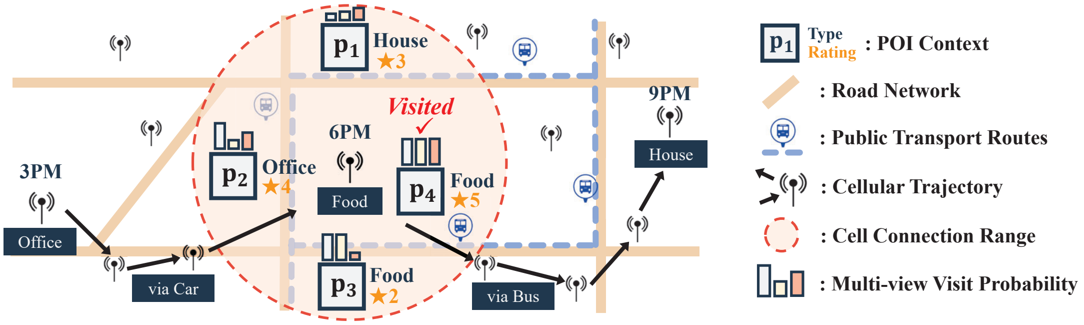

Fig. 1: Main intuition of *Pincette*. Suppose that a user visits a food area to eat dinner after leaving work before going back home. **(1)** In the efficiency-view, the user may not visit p_1 since his/her cellular trajectory is not efficient to get to p_1. **(2)** In the periodicity-view, the user may not visit p_2 since it is an office and thus out of periodicity. **(3)** In the popularity-view, since p_4 is more popular than p_3, the user probably visits p_4.

1. **Efficiency-view** features help POI reconstruction by a widely known observation that people tend to take an efficient path on the road network when they move in or out to a POI. With the road network and even public transport routes, it is possible to estimate how much a cellular trajectory is efficient to move in or out to a POI.
2. **Periodicity-view** features aid in POI reconstruction based on the spatio-temporal periodic pattern in people's movement. By estimating the periodic POI-type transition pattern of people with POI contexts, it is possible to infer which POI type the user of a cellular trajectory visited in a specific time interval.
3. **Popularity-view** features enhance POI reconstruction under the assumption that people tend to visit popular POIs more often. With the useful popularity features in POI contexts such as 5-star user ratings, it is possible to estimate how much a POI is visitable by users.

To fully take advantage of the multi-view features essential for our problem, *Pincette* comprises two steps.

- **Multi-view Feature Extraction**: Discriminative features are extracted for each view. **(1)** For the efficiency-view, the various distances of a trajectory to a POI based on its estimated transportation type are extracted. If a distance is high, the trajectory is not an efficient path to the POI. **(2)** For the periodicity-view, the local industry type and user's hourly flow of each cell tower are extracted. Accordingly, periodic POI type transition patterns can be estimated. **(3)** For the popularity-view, the properties that well explain a POI's popularity including its rating, number of comments, and size are extracted.
- **GCN-LSTM-based POI Estimation**: To effectively reconstruct truly visited POIs from the extracted multi-view features, we propose a deep neural network (DNN)-based POI estimation model. Specifically, a graph convolutional network (GCN) learns spatial industry types of each cell, and then periodic type transition patterns are learned by a long short term memory (LSTM). Finally, the model estimates the POI visit score of a trajectory from the integrated representation of the three views.

II. Related Work

A. Digital Contact Tracing

A recent study [3] shows that epidemic control with digital contact tracing apps using mobile devices plays an essential role in mitigating the spread of highly contagious diseases such as COVID-19. Many public health authorities and big tech companies have developed digital contact tracing apps using wireless technologies such as Bluetooth Low Energy (BLE) [4] and GPS [1]. Notably, Google and Apple jointly announced a new exposure notification system based on decentralized privacy-preserving proximity tracing [4]. The Germany government developed *Corona-Warn-App* [8] using BLE to exchange random codes between devices. In Bulgaria, *ViruSafe* generates heat maps of potentially infected people via a location tracker based on GPS coordinates. However, a low user adoption rate of BLE- or GPS-based contact tracing apps, which is mainly due to privacy concerns, limits the potential benefits expected on epidemic control [9].

B. Cellular Trajectory Reconstruction

Conventional studies use call detail record (CDR) data, where the location of the cell tower to which each user is being connected is recorded only when a call is made. The Voronoi diagram is commonly employed to infer the coverage of a cell tower. Algizawy et al. [10] utilized map-matching to find exit road segments at Voronoi cell boundaries. Chen et al. [11] factorized cellular trajectory tensors according to time contexts such as weeks, days, and time. The main focus of these approaches is to reconstruct temporally *missing* locations caused by the low sampling rate of the CDR. Recently, with a widespread of 4G and 5G, cellular trajectories are being collected more frequently, thereby leading to more precise trajectory reconstruction. Huang et al. [7] and Shen et al. [6] proposed hidden-Markov model (HMM)-based map-matching models to infer the road segments on which each moving trajectory passed; their methods can be used to leverage useful patterns in *moving* trajectories, *e.g.*, for similar trajectory retrieval. However, for digital contact tracing, reconstructing a location where a user *stays together with others* is more important, which is largely neglected by the existing literature.

TABLE I: Summary of the notation.

Notation	Description
$\mathcal{T}$	the set of all cellular trajectories
τ	a cellular trajectory $\tau = \{c_1, \cdots, c_t\}$
c	a cell composed of $\langle$id, longitude, latitude$\rangle$
$\mathcal{P}_c$	the set of POIs belonging to a cell c
p	a POI under consideration
$Y_{\langle \tau,p,c \rangle}$	whether p in c is visited by τ (1: visited, 0: unvisited)
$\mathcal{G}$	a road network of a node set $\mathcal{N}$ and an edge set $\mathcal{E}$
$\mathcal{B}$	a set of public transport routes (*e.g.*, bus)

III. Problem Setup

A. Input Datasets and Notation

We here denote the notation regarding four input datasets– a cellular trajectory from a mobile carrier and three external datasets including POI contexts, road networks, and public transport routes. Table I summarizes the notation.

Cellular Trajectory: A user's cellular trajectory $\tau = \{c_1, c_2, \cdots, c_t\}$ is a sequence of connected cell towers in a certain period of time. Each connected cell c_t at time t consists of $\langle$id, longitude, latitude$\rangle$. The time interval between two consecutive cells may be irregular in a range of few minutes.

POI Context: $\mathcal{P}_c$ is the set of all POIs belonging to a cell c, which are potential candidates to visit when the user's trajectory τ is connected to the cell c. Each POI $p \in \mathcal{P}_c$ can have various attributes such as the industry type (*e.g.*, residential or commercial), ratings by users, and building size.

Road Network: A road network $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ consists of a node set $\mathcal{N}$ and an edge set $\mathcal{E}$, where the element of $\mathcal{N}$ is the start or end point of a road segment (*e.g.*, intersection) and that of $\mathcal{E}$ is the road segment between two nodes. The road network greatly helps infer visited POIs since people usually go along the road before and after visiting a POI.

Public Transport Route: A public transport route is a sequence of stations. The set $\mathcal{B}$ of all routes contains many transport types such as bus and subway. Since, in urban areas, people often use public transport before and after visiting a POI, this dataset serves a complementary information source.

B. POI-level Trajectory Reconstruction

Given a user's cellular trajectory, a stay cell in Definition 1 is regarded as the target cell to find a visited POI in Definition 2. Accordingly, the problem of *POI-level cellular trajectory reconstruction* is described in Definition 3.

Definition 1. (Stay Cell) A cell $c \in \tau$ is a *stay cell* if a user stays more than ϵ minutes, where τ is the user's cell trajectory. We denote the set of stay cells in the trajectory as $\tau^* \subseteq \tau$. □

Definition 2. (Visited POI) A POI p is a *visited POI* if the user visited a POI p belonging to a stay cell c, where $p \in \mathcal{P}_c$ and $c \in \tau$. For each $p \in \mathcal{P}_c$ and $c \in \tau^* \subseteq \tau$, $Y_{\langle \tau,c,p \rangle} = 1$ if p is a visited POI, and $Y_{\langle \tau,c,p \rangle} = 0$ otherwise. □

Definition 3. (POI-level Reconstruction) The *POI-level cellular trajectory reconstruction* is, given a set $\mathcal{T}$ of all users' cellular trajectories, to identify all visited POIs from every $\tau \in \mathcal{T}$. □

TABLE II: Summary of the extracted multi-view features.

View	Feature	Description
X^E	f_1, f_6	Transportation mode of τ (private or public)
	f_2, f_7	Dynamic time warping (DTW) distance of τ to the shortest road path to or from a POI p
	f_3, f_8	Euclidean distance of a POI p to the preceding or following cell in τ
	f_4, f_9	Euclidean distance of a POI p to the nearest station on the transport route taken by τ
	f_5, f_{10}	Euclidean distance of a POI p to the nearest station *not* on the transport route taken by τ
X^I	f_{11}	Industry-type frequency vector of a cell c (*e.g.*, (food: 100, office: 50, $\cdots$))
	f_{12}	Number of people in a cell c at time t
X^P	f_{13}	Industry type of a POI p (*e.g.*, food)
	f_{14}	Average rating of a POI p
	f_{15}	Number of comments on a POI p
	f_{16}	Size (*i.e.*, area) of a POI p

IV. Multi-view Feature Extraction

Given a cellular trajectory $\tau \in \mathcal{T}$, for each $p \in \mathcal{P}_c$ and $c \in \tau^* \subseteq \tau$, **(1)** the *efficiency-view* feature $X^E_{\langle \tau,p \rangle}$ is derived for each pair of the trajectory τ and a POI p, **(2)** the *periodicity-view* feature $X^I_{\langle c \rangle}$ is derived for each stay cell c, and **(3)** the *popularity-view* feature $X^P_{\langle p \rangle}$ is derived for each POI p. Table II summarizes the extracted multi-view features.

A. Efficiency-view Feature

People tend to take the most efficient path when they visit POIs [12]. Thus, the feature indicating how much a user's trajectory is efficient to arrive at the POI helps the visited POI estimation. In addition, the optimal path to the visited POI differs depending on the transportation mode. When the user moves via private transport, *e.g.*, a car, the visited POI is located near the shortest path on the road network, whereas when via public transport, *e.g.*, a bus and a subway, it is rather located near one of the efficient public transport routes.

A user's cellular trajectory τ can be split into two types of sub-trajectories for each stay cell: the *in-coming* sub-trajectory moving into the cell; and the *out-going* sub-trajectory moving out from the cell. Hence, to benefit from the aforementioned property, we generate the efficiency-view features by considering the transport mode of both sub-trajectories; f_1–f_5 and f_6–f_{10} for in-coming and out-going sub-trajectories, respectively. f_2, f_3, f_7, and f_8 are designed for *private* transport, and f_4, f_5, f_9, and f_{10} are for *public* transport. We extracted f_1 and f_6 by adopting a transport mode detection algorithm [13].

B. Periodicity-view Feature

People have spatio-temporal periodic patterns when they visit POIs [14]. The feature representing the preference change of a user's visited areas can help estimate visited POIs. A periodic pattern, *e.g.*, an office area → a dining area → a residential area, is very helpful to reduce the candidates for visited POIs. To discover *cell-level periodicity*, we build two kinds of periodicity-view features (f_{11} and f_{12}) for characterizing each cell c. f_{12} is derived differently for the visit time t. A sequence of these two features for the previous, current, and next stay cells is fed to the GCN-LSTM-based model.

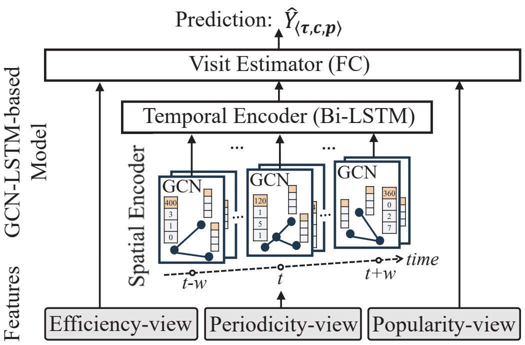

Fig. 2: High-level architecture of *Pincette*.

C. Popularity-view Feature

People tend to visit popular POIs more often than unpopular ones [15]. Thus, the feature indicating the popularity of a POI can enhance the quality of visited POI estimation. We extract four kinds of popularity-view features (f_{13}–f_{16}), such as industry type, POI rating, number of comments, and area size, for every POI p. Overall, the higher the values of f_{14}–f_{16} for a POI p, the higher probability of a user visiting it.

V. Methodology: *Pincette*

Figure 2 illustrates the overall architecture of *Pincette*. It receives the three multi-view features as inputs and performs the GCN-LSTM-based POI estimation to predict the score of a user visiting each POI in a stay cell.

A. Three Main Components

The GCN-LSTM-based model in *Pincette* consists of *spatial encoder*, *temporal encoder*, and *visit estimator*. The first two modules capture complex spatio-temporal periodic patterns from the periodicity-view feature; the third module aggregates all multi-view features and makes the final prediction.

1) Spatial Encoder (SE): The spatial encoder is a GCN-based encoder [16], which aggregates the *spatial movement* patterns of users across cells. In the GCN, a node refers to a cell, and an edge indicates the adjacency between two cells, *e.g.*, Voronoi boundary. The SE receives two inputs: (1) all periodicity-view features $X^I_{\langle\cdot\rangle}$ as its input node feature matrix and (2) the adjacency matrix A between the cells as its input binary edge feature. Then, the SE is formulated as

$$\begin{aligned} \mathrm{SE} = \mathrm{GCN}(X^I_{\langle\cdot\rangle}, A; W) = \prod_{l=1}^{L} H^{(l)} \\ \text{s.t.}\quad H^{(l+1)} = \sigma(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{\frac{1}{2}}H^{(l)}W^{(l)}), \end{aligned} \tag{1}$$

where $H^{(l)}$ is the matrix of the l-th layer activations, *i.e.*, $H^{(0)}{=}X^I_{\langle\cdot\rangle}$, $W^{(l)}{\in}\mathbb{R}^{m\times k}$ is the weight matrix of the l-th layer with input size m and output size k, $\tilde{A}{=}A{+}I$ is the adjacency matrix with self-connection, $\tilde{D}_{ii}{=}\sum_j \tilde{A}_{ij}$ is a diagonal matrix for normalization, and $\sigma(\cdot)$ is the ReLU activation.

2) Temporal Encoder (TE): The temporal encoder is a Bi-LSTM-based encoder [17], which captures the *periodic POI visit* patterns of users. Since spatial patterns are helpful for identifying people's temporal movement patterns, the output node features extracted from the SE are used as an input sequence to the TE. Specifically, for a trajectory $\tau^* = \{c_{t-w}, \cdots, c_t, \cdots, c_{t+w}\}$ centered at the current cell c_t, the TE is formulated as

$$\begin{aligned} \mathrm{TE}_{\langle\tau^*,c_t\rangle} = \text{Bi-LSTM}(Seq_{\langle\tau^*,c_t\rangle}) \\ \text{s.t.}\quad Seq_{\langle\tau^*,c_t\rangle} {=} \big[\mathrm{SE}_{\langle c_{t-w}\rangle},\ \cdots,\ \mathrm{SE}_{\langle c_{t+w}\rangle}\big], \end{aligned} \tag{2}$$

where $\mathrm{SE}_{\langle c\rangle}$ represents the output node feature of the cell c, which is equivalent to the corresponding row vector of the output matrix SE in Eq. (1). $Seq_{\langle\tau^*,c_t\rangle}$ is a sequence of all output node features in τ^*.

3) Visit Estimator (VE): The visit estimator is a fully connected (FC) layer, which predicts the visit probability of each POI in a stay cell. To use all three multi-view features, the output of the TE is concatenated with efficiency-view feature $X^E_{\langle\tau,p\rangle}$ and popularity-view feature $X^P_{\langle p\rangle}$. The combined features are fed to the FC layer as its input. The visit probability for a POI p in a stay cell $c \in \tau^*$ is estimated by

$$\hat{Y}_{\langle\tau,c,p\rangle} {=} \mathrm{Sigmoid}\Big(\mathrm{FC}\big(\mathrm{Concat}(X^E_{\langle\tau,p\rangle}, \mathrm{TE}_{\langle\tau^*,c\rangle}, X^P_{\langle p\rangle})\big)\Big), \tag{3}$$

where $\mathrm{TE}_{\langle\tau^*,c\rangle}$ represents the output periodic patterns involved with $c \in \tau^*$ in Eq. (2). Accordingly, $0 \le \hat{Y}_{\langle\tau,c,p\rangle} \le 1$, and it is used to judge whether p is a visited POI.

B. Loss Function for Training

Because the target variable $Y_{\langle\tau,c,p\rangle}$ in Definition 2 is binary, we adopt the binary cross entropy (BCE) loss, which is computed for each triple $\langle\tau, c, p\rangle$, *i.e.*, $\hat{Y}_{\langle\tau,c,p\rangle}$ in Eq. (3), by

$$\ell(\tau, c, p) = \frac{1}{|\mathcal{P}_c|}\sum_{p\in\mathcal{P}_c} \mathrm{BCE}(Y_{\langle\tau,c,p\rangle}, \hat{Y}_{\langle\tau,c,p\rangle}), \tag{4}$$

where $\mathrm{BCE}(Y, \hat{Y}) = Y\log\hat{Y} + (1-Y)\log(1-\hat{Y})$. The loss in Eq. (4) is aggregated for all stay cells in a user's trajectory τ^* and then for all user trajectories in the mini-batch $\mathcal{M}$ to obtain the final loss $\mathcal{L}$,

$$\mathcal{L}(\mathcal{M}) = \frac{1}{|\mathcal{M}|}\sum_{\tau^*\in\mathcal{M}} \frac{1}{|\tau^*|}\sum_{c\in\tau^*} \ell(\tau, c, p). \tag{5}$$

VI. Experiments

A. Data Collections

Table III summarizes the statistics of the two data collections. Each data collection contains human mobility (*i.e.*, trajectories), cell-tower information, POI information, and road and public transport networks.

1) Data Collection for Beijing:

- *GeoLife* [18] is a GPS trajectory dataset provided from the GeoLife project by Microsoft Research Asia. It contains over 20 million GPS data points of 17,621 trajectories of 182 users collected in Beijing from 2007 to 2012.
- *OpenCellID* [19] is a open source project that collects GPS coordinates and location IDs of cell towers in the world. We used the GPS coordinates of 30,254 cell towers in Beijing.
- *OpenStreetMap* is an open-source project that provides various types of geographic data. We obtained the road

TABLE III: Summary statistics of the data collections for Beijing and Chania. $\mathbb{E}[|\mathbb{P}_c|]$ (or $\mathbb{E}[\sum Y_{\mathbb{P}_c}]$) is the number of POIs (or true visited POIs) in a cell on average.

Data	# Points	# Users	# Stays	$\mathbb{E}[\lvert\mathbb{P}_c\rvert]$	$\mathbb{E}[\sum Y_{\mathbb{P}_c}]$
Beijing	20M	182	11,111	15.14	1.34
Chania	3M	10	505	42.71	1.22

networks, bus and subway routes, and locations and sizes of the buildings in Beijing through the Overpass API.

- *BaiduMaps* is a web-mapping application that provides rich information on POIs. We used the locations, industry types, ratings, and comment counts of the POIs in Beijing.

2) Data Collection for Chania (in Greece):

- *MySignals* [20] is a cellular trajectory dataset *with* mapped GPS coordinates. It contains over 3 million cell-GPS data points of 10 users collected in Chania from 2012 to 2013.
- *OpenStreetMap* [19] was used again, and the same kinds of information were obtained for Chania.
- *GoogleMaps* is a web-mapping application provided by Google. We used the locations, industry types, ratings, and comment counts of the POIs in Chania.

3) Preprocessing for Beijing: We carefully converted the GPS trajectories in GeoLife into cellular trajectories and true visited POI labels, by using the cell tower information in OpenCellID and the POI information in BaiduMaps. We considered only the center area of Beijing ($40km \times 40km$) covering around 75% of data in GeoLife.

- *Cellular Trajectory*: A sequence of $\langle$longitude, latitude$\rangle$ in a GPS trajectory was transformed to a sequence of cell tower IDs, using the Voronoi diagram as in the relevant work [10], [11]. A Voronoi cell represents a connection range of a cell tower, so that the GPS points contained in the Voronoi cell are converted to the corresponding cell tower ID.
- *True Visited POI*: Given a stay cell, a POI is marked as the true visited POI if it is closest to the GPS point. For each of $11,111$ stay cells, 1.34 POIs were marked as ”visited” out of 15.14 POIs on average.

4) Preprocessing for Chania (in Greece): Contrary to the data preprocessing procedure for Beijing, we only extracted true visited POI labels using the MySignals GPS-cellular trajectory dataset, because MySignals is a *genuine* cellular trajectory dataset with mapped GPS coordinates. Given a stay cell, a POI is marked as the true visited POI if it is closest to the GPS point. For each of 505 stay cells, 1.22 POIs were marked as “visited” out of 42.71 POIs on average.

B. Experiment Setting

1) Baselines: Since our work is the first attempt to reconstruct the cellular trajectories at POI-level, there exists no previous algorithm that can be naturally employed in our problem. For thorough comparison with *Pincette*, we evaluated two heuristic rule-based POI estimation algorithms, denoted $RULE^E$ and $RULE^P$, a Markov transition model based algorithm, denoted MTM^I, and a slight modification of a HMM-based *road-level* cellular trajectory reconstruction algorithm [6], denoted $CellSim^+$. $RULE^E$ (or $RULE^P$) focuses on only the efficiency view (or the popularity view).

- $RULE^E$ uses the inverse of the sum of the features f_2 to f_{10} in the efficiency view in Table II as the POI visit probability, assuming that people tend to follow efficient ways when visiting POIs.
- $RULE^P$ uses the multiplication of the features f_{14} to f_{16} in the popularity view in Table II as the POI visit probability, assuming that people tend to visit popular POIs.
- MTM^I builds a Markov transition probability matrix from the feature f_{11} in the periodicity view in Table II, assuming that people have periodic patterns of *POI type* transitions. Given the previous and next cells, MTM^I uses the type prediction probability from the Markov transition model as the POI visit probability. Since the POIs of the same type get the same visit probability, we randomly assign the ranking among the POIs of the same type to break a tie.
- $CellSim^+$ [6] was originally designed to infer multiple road segments of moving trajectories by map matching with a rule-based HMM. We reconstruct multiple road segments from the cellular trajectories following the original paper, and then use the inverse of the vertical distance from the estimated road segment as the POI visit probability.

2) Evaluation Metrics: We used four commonly-used ranking accuracy metrics [21]: Precision@k, Recall@k, F1-score@k, and normalized discounted cumulative gain (NDCG)@k. For all metrics with varying k, higher scores indicate more accurate prediction results.

3) Training Configuration: The entire data in each collection is divided into the training set and the test set with the ratio of 80:20 in the chronological order. We set the stay threshold ϵ to be 30 minutes. *Pincette* was implemented using PyTorch 1.2.0 and executed on a single NVIDIA Titan Volta GPU. The Adam optimizer with a learning rate of 0.01 and a batch size of 32 are used to train *Pincette*. Only w, the length of a window, and L, the number of hidden layers in the GCN of the spatial encoder, need to be tuned for *Pincette*, and we simply set both hyperparameters to be 1 because the duration of each cellular trajectory is typically not very long. For MTM^I and $CellSim^+$, we used the default or best hyperparameter values suggested in the original papers. For reliable evaluation, we reported the average of *five* repetitions in each test.

C. Overall Accuracy Comparison

Table IV shows the ranking accuracy results of the five algorithms including *Pincette* on the two data collections. Overall, *Pincette* achieved the highest ranking accuracy in all metrics on both data collections, thanks to the sophisticated incorporation of the multi-view features. Meanwhile, $CellSim^+$ achieved the second best ranking accuracy. For each data collection, *Pincette* improved the accuracy of the state-of-the-art algorithm, $CellSim^+$, by up to 21.20% and 14.62%, respectively. Also, the overall performance on the Beijing collection is higher than that of the Chania collection, because the former has a larger number of samples ($\#Stays$

TABLE IV: Overall accuracy comparison on two data collections (the best results are in bold, the second best results are underlined, and the % improvements over the second best are in italic).

Metrics		precision			recall			f1-score			NDCG		
Data	Method	@1	@3	@5	@1	@3	@5	@1	@3	@5	@1	@3	@5
Beijing	$RULE^E$	0.212	0.309	0.210	0.189	0.531	0.631	0.200	0.391	0.315	0.241	0.405	0.501
	$RULE^P$	0.201	0.296	0.201	0.184	0.515	0.628	0.192	0.376	0.304	0.219	0.391	0.522
	MTM^I	0.187	0.280	0.190	0.143	0.487	0.600	0.162	0.356	0.289	0.173	0.355	0.486
	$CellSim^+$	$\underline{0.224}$	$\underline{0.334}$	$\underline{0.245}$	$\underline{0.199}$	$\underline{0.558}$	$\underline{0.650}$	$\underline{0.211}$	$\underline{0.418}$	$\underline{0.355}$	$\underline{0.255}$	$\underline{0.431}$	$\underline{0.551}$
	Pincette	**0.251**	**0.350**	**0.257**	**0.211**	**0.574**	**0.663**	**0.229**	**0.435**	**0.370**	**0.309**	**0.456**	**0.573**
	%improve	*12.02*	*4.69*	*5.04*	*5.76*	*3.01*	*2.04*	*8.62*	*4.06*	*4.20*	*21.20*	*5.95*	*4.00*
Chania (in Greece)	$RULE^E$	0.125	0.064	0.035	0.121	0.183	0.183	0.123	0.095	0.059	0.129	0.159	0.159
	$RULE^P$	0.062	0.041	0.022	0.078	0.104	0.104	0.069	0.059	0.036	0.068	0.068	0.087
	MTM^I	0.048	0.048	0.026	0.048	0.145	0.145	0.048	0.073	0.045	0.048	0.097	0.097
	$CellSim^+$	$\underline{0.135}$	$\underline{0.102}$	$\underline{0.062}$	$\underline{0.135}$	$\underline{0.185}$	$\underline{0.185}$	$\underline{0.135}$	$\underline{0.131}$	$\underline{0.093}$	$\underline{0.135}$	$\underline{0.164}$	$\underline{0.164}$
	Pincette	**0.154**	**0.113**	**0.069**	**0.155**	**0.191**	**0.191**	**0.154**	**0.142**	**0.101**	**0.154**	**0.187**	**0.187**
	%improve	*13.82*	*11.09*	*10.91*	*14.62*	*2.90*	*2.90*	*14.22*	*8.56*	*9.16*	*13.74*	*13.91*	*13.91*

TABLE V: F1-score@k, Precision@k, recall@k, and NDCG@k results of the combinations of three multi-view features in *Pincette* on the Beijing collection (the best results are in bold and the second best are underlined).

Multi-view Features			precision			recall			f1-score			NDCG		
Efficiency	Periodicity	Popularity	@1	@3	@5	@1	@3	@5	@1	@3	@5	@1	@3	@5
○	×	×	0.226	0.330	0.225	0.195	0.553	0.635	0.210	0.413	0.332	0.269	0.428	0.522
×	○	×	0.224	0.336	0.230	0.195	0.555	0.615	0.208	0.419	0.334	0.264	0.433	0.526
×	×	○	0.205	0.304	0.206	0.188	0.529	0.606	0.196	0.386	0.307	0.196	0.386	0.307
○	○	×	$\underline{0.246}$	$\underline{0.347}$	$\underline{0.254}$	$\underline{0.211}$	$\underline{0.568}$	$\underline{0.658}$	$\underline{0.227}$	$\underline{0.431}$	$\underline{0.367}$	$\underline{0.294}$	$\underline{0.453}$	$\underline{0.564}$
×	○	○	0.235	0.330	0.225	0.204	0.550	0.645	0.219	0.423	0.334	0.277	0.433	0.526
○	×	○	0.240	0.327	0.242	0.210	0.559	0.649	0.224	0.423	0.353	0.224	0.413	0.353
○	○	○	**0.251**	**0.350**	**0.257**	**0.211**	**0.574**	**0.663**	**0.229**	**0.435**	**0.370**	**0.309**	**0.456**	**0.573**

in Table III) and a smaller number of POI candidates in a cell on average ($\mathbb{E}[|\mathbb{P}_c|]$ in Table III) than the latter.

D. *Efficacy of Multi-view Features*

We analyzed the efficacy of the three view features used in *Pincette* through an ablation study on the Beijing collection, which is larger among the two collections. Table V shows the precision@k, recall@k, f1-score@k, and NDCG@k results of the variants of *Pincette*, each of which is a possible combination of the three view features. Every individual feature is shown to be effective. While putting all three view features together is the most effective regardless of k, combining the efficiency-view and periodicity-view features is the second most effective, which indicates that those two views play an important role in predicting POI visit patterns.

VII. Conclusion

In this paper, we presented the *POI-level cellular trajectory reconstruction* problem that can significantly benefit digital contact tracing for preventing the spread of infectious diseases. A novel algorithm, *Pincette*, is proposed to incorporate the efficiency, periodicity, and popularity aspects in a sophisticated way to predict the visited POIs from a coarse-grained cellular trajectory. Overall, we believe that *Pincette* will promote the usability of cellular network data in digital contact tracing for ever-emerging infectious diseases.

Acknowledgement

This work was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2022-0-00157, Robust, Fair, Extensible Data-Centric Continual Learning).

References

[1] T. Martin *et al.*, "Demystifying COVID-19 digital contact tracing: A survey on frameworks and mobile apps," *WCMC*, 2020.

[2] M. Kim *et al.*, "Hi-COVIDNet: Deep learning approach to predict inbound COVID-19 patients and case study in South Korea," in *SIGKDD*, 2020.

[3] L. Ferretti *et al.*, "Quantifying SARS-CoV-2 transmission suggests epidemic control with digital contact tracing," *Science*, 2020.

[4] J. Li and X. Guo, "Global deployment mappings and challenges of contact-tracing apps for COVID-19," 2020, available at SSRN 3609516.

[5] R. Hinch *et al.*, "Effective configurations of a digital contact tracing app: A report to NHSX," University of Oxford, Tech. Rep., 2020.

[6] Z. Shen *et al.*, "Retrieving similar trajectories from cellular data at city scale," 2019, arXiv preprint arXiv:1907.12371.

[7] X. Huang *et al.*, "CTS: A cellular-based trajectory tracking system with GPS-level accuracy," *IMWUT*, 2018.

[8] J. H. Reelfs *et al.*, "Corona-Warn-App: Tracing the start of the official COVID-19 exposure notification app for Germany," in *SIGCOMM*, 2020.

[9] E. Seto *et al.*, "Adoption of COVID-19 contact tracing apps: A balance between privacy and effectiveness," *JMIR*, 2021.

[10] E. Algizawy *et al.*, "Real-time large-scale map matching using mobile phone data," *TKDD*, 2017.

[11] G. Chen *et al.*, "Complete trajectory reconstruction from sparse mobile phone data," *EPJ Data Science*, 2019.

[12] S. Zhu *et al.*, "Do people use the shortest path? An empirical test of Wardrop's first principle," *PloS One*, 2015.

[13] Y. Qu *et al.*, "Transportation mode split with mobile phone data," in *ICITS*, 2015.

[14] C. Song *et al.*, "Limits of predictability in human mobility," *Science*, 2010.

[15] Y. Liu *et al.*, "An experimental evaluation of point-of-interest recommendation in location-based social networks," *PVLDB*, 2017.

[16] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, arXiv preprint arXiv:1609.02907.

[17] Z. Huang *et al.*, "Bidirectional LSTM-CRF models for sequence tagging," 2015, arXiv preprint arXiv:1508.01991.

[18] Y. Zheng *et al.*, "Geolife: A collaborative social networking service among user, location and trajectory," *Data Engineering Bulletin*, 2010.

[19] T. Landspurg, "OpenCellid," https://opencellid.org/, 2021.

[20] E. Alimpertis and A. Bletsas, "CRAWDAD dataset tuc/mysignals," https://crawdad.org/tuc/mysignals/20191030, 2019.

[21] D. Park *et al.*, "TRAP: Two-level regularized autoencoder-based embedding for power-law distributed data," in *WebConf*, 2020.