Toward Discriminating and Synthesizing Motion Traces Using Deep Probabilistic Generative Models

Fan Zhou, *Member, IEEE*, Xin Liu, Kunpeng Zhang, and Goce Trajcevski, *Member, IEEE*

***Abstract*— Mining knowledge from human mobility, such as discriminating motion traces left by different anonymous users, also known as the trajectory-user linking (TUL) problem, is an important task in many applications requiring location-based services (LBSs). However, it inevitably raises an issue that may be aggravated by TUL, i.e., how to defend against location attacks (e.g., deanonymization and location recovery). In this work, we present a Semisupervised Trajectory- User Linking model with Interpretable representation and Gaussian mixture prior (STULIG)—a novel deep probabilistic framework for jointly learning disentangled representation of user trajectories in a semisupervised manner and tackling the location recovery problem. STULIG characterizes multiple latent aspects of human trajectories and their labels into separate latent variables, which can be then used to interpret user check-in styles and improve the performance of trace classification. It can also generate synthetic yet plausible trajectories, thus protecting users' actual locations while preserving the meaningful mobility information for various machine learning tasks. We analyze and evaluate STULIG's ability to learn disentangled representations, discriminating human traces and generating realistic motions on several real-world mobility data sets. As demonstrated by extensive experimental evaluations, in addition to outperforming the state-of-the-art methods, our method provides intuitive explanations of the classification and generation and sheds lights on the interpretable mobility mining.**

***Index Terms*— Disentangled representation, location privacy, trace discrimination, variational autoencoder (VAE).**

Nomenclature

$\mathcal{U}$	Set of users.
$c_{i,u}$	ith check-in by user u.
$\mathbf{T_u}$	Trajectory generated by user u.
$\widetilde{\mathbf{T}}$	Unlabeled trajectory.
$\mathcal{T}$	Set of unlabeled trajectories.
θ and ϕ	Parameters of decoder and encoder.
$\mathbf{z}$ and $p_\theta(\mathbf{z})$	Latent factor and its prior.
$\mathcal{L}(*)$	Evidence lower bound (ELBO).
$p(\mathbf{T}\|\mathbf{z})$	Generative networks (decoder).
$q(\mathbf{z}\|\mathbf{T})$	Inference networks (encoder).
$q(u\|\mathbf{T})$	Classifier needs to estimated.
$p(\mathbf{z}_1) \sim \mathcal{N}(0, \mathbf{I})$	Unimodal Gaussian.
$p(\mathbf{z}_2\|\mathbf{z}_1)$	Mixture of Gaussians.

Manuscript received October 22, 2019; revised March 14, 2020; accepted June 21, 2020. Date of publication August 12, 2020; date of current version June 2, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant 61602097 and Grant 61472064, in part by the NSF under Grant 1213038, and in part by the Division of Computer and Network Systems (CNS) under Grant 1646107. *(Corresponding author: Kunpeng Zhang.)*

Fan Zhou is with the School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu 610051, China (e-mail: fan.zhou@uestc.edu.cn).

Xin Liu is with the 360 AI Security Research Labs, University of Electronic Science and Technology of China, Chengdu 610051, China (e-mail: lxin0502@gmail.com).

Kunpeng Zhang is with the Department of Decision, Operations and Information Technologies, University of Maryland, College Park, MD 20742 USA (e-mail: kpzhang@umd.edu).

Goce Trajcevski is with the Department of Electrical and Computer Engineering, Iowa State University, Ames, IA 50011 USA (e-mail: gocet25@iastate.edu).

Color versions of one or more of the figures in this article are available online at https://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2020.3005325

I. Introduction

RECORDING large volumes of geotagged behavioral data enabled by location-based social networks (LBSNs), e.g., Foursquare, Yelp, and Instagram, has recently spurred research activities on uncovering user check-in preference and moving patterns, which are important for many downstream applications, such as venue recommendation [1], next location prediction [2], [3], social link/circle prediction [4], and human trace classification [5], [6]. Recently, an important human mobility task called trajectory-user linking (TUL) [5], [6] has received increased research attention. The objective of TUL is to discriminate and classify the unknown motion footprints to known users in LBSN. This is crucial for a variety of downstream practical tasks, ranging from identifying the suspects/criminals and recommending personalized items to spatial event detection and epidemic trend prediction.

At the core of human mobility pattern mining are various machine learning models depending on the data modeling and representation. For example, for data representing continuous check-ins, classical time series models [e.g., Markov chain, hidden Markov model (HMM), and recurrent neural networks (RNNs)] are widely used to mine the dependences among spatiotemporal data samples [7]. If the data are structured as a matrix and the target problem is recommendation, collaborative filtering techniques, such as matrix factorization and its many variants, are the *de facto* framework [8], [9]. Recently, deep learning-based models have readdressed many spatiotemporal prediction tasks providing improvements over the state-of-the-art models in many aspects. For example, the RNN-based models, such as LSTM [10] and GRU [11], have been recently adapted for modeling the human mobility [3], [12], largely due to their flexibility in capturing longer term dependences among locations. RNN-based models were also proposed for improving point-of-interest (POI)

recommendation performance [13], inferring the social community each user belongs to by exploiting their check-in behavior in trajectories [4]. Other deep learning techniques, such as attention mechanism and generative models, have also been exploited to improve the mobility learning performance [6], [14].

From a broad perspective, the existing methods usually capture the sequential check-in patterns with recurrent units combined with various attention mechanisms and make predictions based on the hidden states of recurrent layers. While outperforming traditional mobility modeling methods based on Markov chain (MC), they still have several limitations.

L1 (Interpretability): Lack of disentangled representation of the latent factors governing human mobility patterns.

L2 (Unlabeled Data): While generating a large amount of geotagged data, LBSNs usually do not allow the leakage of data labels, e.g., the user name and profile information, due to privacy issues, which renders a vast number of unlabeled data unusable.

L3 (Structural and Periodical Mobility): Existing models usually focus on learning transition regularities of users' trajectories, ignoring the fact that human movements may exhibit multilevel periodicity.

L4 (Efficiency): RNN models are often more delicate to tune and more brittle to train, accompanied by significant computational burden compared with standard feedforward architectures.

L5 (Privacy): Due to the sensitivity of location data, using real LBSN data raises privacy concerns (i.e., exposing sensitive personal information) which may be worsened by the TUL enabling user deanonymization and location recovery attacks.

In this work, we propose a methodology for tackling the TUL problem in a manner that enables overcoming the limitations discussed earlier. Specifically, we focus on interpreting the trajectory generating process by analyzing the disentangled latent space of the trajectory data. Toward addressing the critical limitations, we propose a Semisupervised Trajectory-User Linking model with Interpretable representation and Gaussian mixture prior (STULIG).

STULIG tackles L1 (Interpretability) by using a variational autoencoder (VAE) model with a mixture of Gaussians (MoG) as the prior of the latent variables. This allows us to encode distinct aspects of human trajectories into separate latent variables and infer latent distributions where a trajectory is generated. By incorporating the unlabeled trajectories into the supervised TUL task under such a semisupervised framework, STULIG is capable of addressing L2 through encoding trajectory labels (e.g., users) as one of the disentangled latent variables. This is also helpful in interpreting the user-generated trajectories from multiple angles—we may understand user identities from one latent variable while their movement patterns from other(s).

For L3 and L4, we propose to learn the human mobility with feedforward architectures, such as convolutional neural networks (CNNs) that are able to capture the structural and periodical patterns of human mobility and can significantly improve the learning efficiency. Empirically, we demonstrate that STULIG, combined with carefully tuned semisupervised Bayesian networks, exhibit comparable performance with the existing RNN-based models.

Lastly but more importantly, STULIG is a deep generative model that can generate synthetic but realistic trajectories for individual users, which could preserve the real traces and yet enable machine learning tasks using only generated data. This property of STULIG allows us to release fake samples for research/commercial usage without sacrificing location privacy (see L5)—enabling compliance with data-privacy regulations such as the European General Data Protection Regulation (GDPR).

In sum, the main contributions of this work are given in the following:

1) a novel semisupervised human mobility learning model which utilizes spatiotemporal features of POIs and rich unlabeled data to improve the TUL performance and training efficiency;
2) a disentangled latent factor learning model for capturing human mobility patterns that can interpret the trajectory generation in LBSNs and can be used for synthesizing plausible mobility patterns;
3) extensive experimental evaluations illustrating the improvements enabled by STULIG over existing models on model interpretation, trace discrimination, and realistic trajectory generation, using publicly available LBSN data sets.

The remainder of this article is organized as follows. We discuss the related work in Section II and introduce the problem and provide the necessary background in Section III. The details of the proposed method are presented in Section IV, followed by the report on comprehensive experimental observations against baselines in Section V. We conclude this article and point some future work remarks in Section VI.

II. Related Work

We now position STULIG with respect to the related literature, categorized in three main bodies of works.

A. Mobility Pattern Mining

Unveiling the governing properties of mobility behavior has been a trending research topic in AI [5], [12], [15]–[17], GIS [4], [18], and venue recommendation systems [1]. Traditionally, human trajectories are modeled via MCs, which can typically capture short-term dependence of user check-ins [19]. In recent years, RNNs become a popular paradigm for modeling sequential data and have achieved great performance in many NLP tasks. Not surprisingly, RNNs have been adopted and widely used to model (human) mobility due to their flexibility in capturing longer term dependence among check-in locations. Numerous works have been proposed to learn human mobility and check-in preferences with various RNN models, including prediction of next check-in location [2], [12], venue recommendation [13], and identifying the users of the trajectories [6]. However, these RNN-based supervised trajectory models suffer from the lack of interpretability of the latent factors governing motion, as well as the efficiency.

In other words, RNN is well known to be difficult to train and computationally expensive.

In this work, we tackle the TUL problem with a novel model that is capable of learning latent factors governing user trajectories and is more efficient than the existing approaches, by utilizing feedforward networks (and without sacrificing effectiveness in comparison with related works).

B. Deep Generative Models

Deep generative models, such as VAE [20], provides a general framework for learning representation of data by fitting a latent variable and allowing inference of the learned latent representation. The latent encoding can also serve as a compressed representation for various downstream tasks, such as image generation [20], [21], text classification [22], [23], and trajectory identification [6]. However, an individual dimension of the latent representation does not necessarily encode any particular semantically meaningful variation [as would the classical principal component analysis (PCA)] and, in general, is not directly amenable to human interpretation [21]. These issues motivated many recent articles to introduce mechanisms for encoding disentangled latent variables [21], [24] for alleviating the problem of poor reconstruction quality in VAEs [25], [26].

In this spirit, the most relevant recent work TULVAE [6]—which extends [5]—uses a generative model to capture latent factors. However, the model follows regular VAEs without encoding disentangled representation of trajectories and, consequently, is not sufficient for interpreting complex trajectories generation. In addition, both [5] and [6] suffer computational overheads due to their RNN-based encoder–decoder.

The STULIG model is inspired by recent advances in improving interpretability of VAEs [27], [28]. While MoG prior is also used in [27], the work focuses on unsupervised clustering of data with VAE. Complementary to this, although [28] is a semisupervised VAE-based model, it is limited to importance weight-based data reconstruction and thus makes the posterior inference intractable. Hence, STULIG differs from this body of earlier works because it tackles the disentangled representation learning and latent variable inference problem in human mobility trajectories, learns both sequential and periodical semantics of human check-in sequences, incorporates the unlabeled data for both discriminating individual mobility pattern and encoding supervised data for some subset of the variables, and introduces a new Evidence Lower BOund (ELBO), tailored for addressing the TUL problem.

C. Generating Plausible Traces

There is a growing interest in releasing data sets for research and commercial usage. Privacy policies of data holders, however, prevent them from sharing their sensitive data [29]. A possible way of tackling this paradox is to allow researchers to access the synthetic data records rather than the real data. Therefore, a major open problem is how to generate synthetic data with provable privacy and to achieve promising utility in various machine learning settings. In this aspect, a privacy-preserving generative model to synthesize location traces by generating consistent lifestyles, meaningful mobilities, and geographical similar traces was presented in [30]. However, their models require the full semantics of the traces (e.g., the categories of locations) as the seeds and suffer computational complexity for preserving high-order semantic features. Another recent work [31] synthesizes trajectory by directly leveraging Wasserstein GAN with gradient penalty [32], which is problematic due to the unstable training and bias loss when applying GANs to discrete data [33].

III. Preliminaries

We now introduce the basic notations and the problem definition, along with the background on VAE and semi-VAE.

Let $\mathbf{T_u} = \{c_{1,u}, \ldots, c_{n,u}\}$ denote a trajectory generated by the user u during a given time interval, where $c_{i,u}$ $(i \in [1, n])$ is a ith check-in for the user u, associated with a check-in time $c_{i,u} \cdot t = t_i$ and geolocation $c_{i,u} \cdot g = \langle c_{i,u} \cdot lo, c_{i,u} \cdot la \rangle$, where lo and la correspond to a longitude and latitude. A trajectory $\widetilde{\mathbf{T}}$ for which we do not know the user who generated it is called unlinked. The frequently used notations in this article are given in the Nomenclature.

A. Trajectory-User Linking (See [5])

Suppose that we have a number of unlinked trajectories $\mathcal{T} = \{\widetilde{\mathbf{T}}_1, \ldots, \widetilde{\mathbf{T}}_M\}$ produced by a set of users $\mathcal{U} = \{u_1, \ldots, u_N\}$ $(M \gg N)$. The TUL problem is to learn a classifying function that links unlinked trajectories to users: $\mathcal{T} \mapsto \mathcal{U}$.

B. Trajectory Generative Model

Given a data set consisting of pairs $(\mathbf{T}_{u_1}, u_1), \ldots, (\mathbf{T}_{u_m}, u_m)$, with the ith trajectory $\mathbf{T}_{u_i} \in \mathcal{T}$ and the corresponding user (label) $u_i \in \mathcal{U}$, we assume that the observed trajectory $\mathbf{T}_{u_i}$ is generated by a latent variable $\mathbf{z}_i$. We omit the index i whenever it is clear that we are referring to terms associated with a single data point, i.e., a trajectory.

We aim at maximizing the probability of each trajectory $\mathbf{T}$ in the training set under the generative model, according to $p_\theta(\mathbf{T}) = \int_{\mathbf{z}} p_\theta(\mathbf{T}|\mathbf{z}) p_\theta(\mathbf{z}) \mathrm{d}\mathbf{z}$, where $p_\theta(\mathbf{T}|\mathbf{z})$ refers to a generative model or decoder, $p_\theta(\mathbf{z})$ is the prior distribution of the random latent variable $\mathbf{z}$, e.g., an isotropic multivariate Gaussian: $p_\theta(\mathbf{z}) = \mathcal{N}(0, \mathbf{I})$ ($\mathbf{I}$ is the identity matrix), and θ is the generative parameters of the model.

C. Variational Autoencoders (See [20])

Typically, to estimate the generative parameters θ, the ELBO—denoted as $\mathcal{L}(\mathbf{T})$—on the marginal likelihood of a single trajectory is used as the objective.

$$\begin{aligned} \log p_\theta(\mathbf{T}) &\geq \log p_\theta(\mathbf{T}) - \mathbb{KL}[q_\phi(\mathbf{z}|\mathbf{T}) \parallel p_\theta(\mathbf{z}|\mathbf{T})] \triangleq \mathcal{L}(\mathbf{T}) \\ &= \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{T})}[\log p_\theta(\mathbf{T}|\mathbf{z})] - \mathbb{KL}[q_\phi(\mathbf{z}|\mathbf{T})||p_\theta(\mathbf{z})] \end{aligned} \tag{1}$$

where $q_\phi(\mathbf{z}|\mathbf{T})$ is an approximation to the true posterior $p_\theta(\mathbf{z}|\mathbf{T})$ (also known as recognition model or encoder) parameterized by ϕ. $\mathbb{KL}[q_\phi(\mathbf{z}|\mathbf{T})||p_\theta(\mathbf{z})]$ is the Kullback–Leibler

divergence between the learned latent posterior distribution $q(\mathbf{z}|\mathbf{T})$ and the prior $p(\mathbf{z})$ (for brevity, we will omit the parameters ϕ and θ in the subsequent formulas). Since the objective is to minimize the KL divergence between $q(\mathbf{z}|\mathbf{T})$ and the true distribution $p(\mathbf{z}|\mathbf{T})$, we can alternatively maximize ELBO $\mathcal{L}(\mathbf{T})$ of $\log p(\mathbf{T}, u)$ with respect to both θ and ϕ, which are jointly trained with separate neural networks such as multilayer perceptrons.

IV. Proposed STULIG Approach

In this section, we focus on the fundamental aspects of our proposed STULIG model.

From a high-level overview perspective, STULIG is a deep Bayesian generative model proposed for mining human mobility data. It consists of three main components: 1) contextual POI embedding, which learns the POI representation in a fully unsupervised manner; 2) latent factor learning model is a mobility generative model, which extends VAE with more interpretable multimodal Gaussian posterior approximation; and 3) semisupervised mobility classification, which aims at combining the learned latent factors to classify human trajectories in a semisupervised learning manner. In the rest of this section, we describe each of these components in greater detail and conclude with a discussion on implementation aspects.

A. Contextual POI Embedding

We use a simple yet effective embedding layer to incorporate contextual factors to represent the check-ins. Specifically, we embed each check-in c_i as a low-dimensional vector $\mathbf{v}_i^s \in \mathbb{R}^{d_i}$ using any distributed representation method, such as word2vec [34]. Here, we follow previous works [3], [12] using the CBOW architecture to embed the check-in ids, which is to predict this check-in given its contexts in a trajectory. We trained the check-in id embedding on all available trajectories. Since the check-in time $c_i \cdot t$ can be quantized into discrete time intervals, we denote each $c_i \cdot t$ as a 24-D one-hot vector $\mathbf{v}_i^t$. The geographical location of each check-in representation is obtained by transforming its dense representation (i.e., $\langle c_i \cdot lo, c_i \cdot la \rangle$) to a low-dimensional vector $\mathbf{v}_i^g$ through a simple fully connected network with nonlinear transformation. Finally, we concatenate ($\oplus$) the three vectors as $\mathbf{v}_i = [\mathbf{v}_i^s \oplus \mathbf{v}_i^t \oplus \mathbf{v}_i^g] \in \mathbb{R}^d$ to represent the contextual information associated with each check-in.

Typically, the trajectories of a given user u_i are generated across multiple days. Previous research suggests segmenting the trajectory data $\mathbf{T}$ into k consecutive subsequences $\mathbf{T}^1, \ldots, \mathbf{T}^k$, where k is the number of days on which a user u_i has check-in activities. In this work, we consider two additional contexts when splitting an individual trajectory in a daily frequency: 1) temporal—an individual trajectory $\mathbf{T}^j$ is split into $\mathbf{T}_1^j, \mathbf{T}_2^j, \ldots, \mathbf{T}_m^j$ for m distinct periods within a given day [following [5], in our experiments $m = 4$ (i.e., 6-h intervals)] and 2) spatial—we further split a subtrajectory $\mathbf{T}_l^j$ into $\mathbf{T}_{l,1}^j, \mathbf{T}_{l,2}^j, \ldots, \mathbf{T}_{l,s}^j$ whenever the corresponding distance (e.g., Euclidian, road network, and travel time) between the last POI in $\mathbf{T}_{l,i}^j$ and the first POI in $\mathbf{T}_{l,i+1}^j$ exceeds a certain threshold. For instance, in [35], it is reported that transition distances are usually less than 50 km. After the above preprocessing, a trajectory is embedded into a 2-D latent matrix, which can be treated as an image.

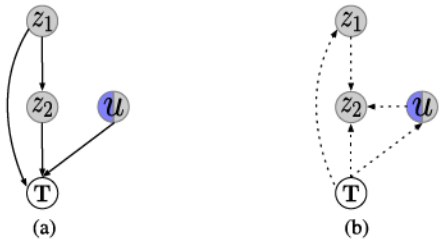

Fig. 1. Probabilistic graphical models of STULIG. (a) Generative model. (b) Inference model.

B. Interpretable Latent Factor Models

We now develop a more interpretable representation of mobility data and explain its generative and inference model.

When learning the human mobility behavior, one confronts the challenges of data scarcity and limited labeled data [1], [6]. We are interested in leveraging abundant unlabeled data to improve the performance of the supervised tasks, such as TUL. However, there are variations in the mobility that are easy to understand, e.g., identity and spatiotemporal features, and other variations that are less explainable, such as the moving patterns and trajectory semantics. Thus, it is desirable to partially specify the explicit variation from which we can extract supervision signals for disentangled representation with probabilistic graphical models while leaving the rest to be in an entangled manner that can be learned with deep generative models.

To enable semisupervised learning and disentangled mobility representation, we consider a more general class of probabilistic graphical models in which the trajectory data are generated hierarchically and the approximate posterior can be conditioned on different distributions depending on the latent factors we can partially specify.

1) Generative Model: Previous VAE-based works usually rely on an isotropic Gaussian as the prior of the latent variable [6], [22], [36]. This, however, is limited because the learned representation is unimodal and does not allow for interpretability. A number of works have tackled this limitation. One group of ELBO extensions [25], [26] focused on improving the quality of latent representation by introducing more sophisticated regularization of the ELBO. Other recent works have suggested that one should improve the priors rather than paying attention only to the reconstruction part (see [37], [38]).

In STULIG, we choose an MoG as the prior, which results in a structural trajectory generative model [as shown in Fig. 1(a)]

$$p(\mathbf{T}, u, \mathbf{z}_1, \mathbf{z}_2) = p(u)p(\mathbf{z}_1)p(\mathbf{z}_2|\mathbf{z}_1)p(\mathbf{T}|u, \mathbf{z}_1, \mathbf{z}_2) \quad (2)$$

where the prior $p(u)$ is a multinomial distribution, treated as the latent variable if the class label (e.g., the user) is unavailable; the latent factor $\mathbf{z}_1$ is an unimodal Gaussian

$p(\mathbf{z}_1) = \mathcal{N}(0, \mathbf{I})$, $p(\mathbf{T}|u, \mathbf{z}_1, \mathbf{z}_2)$ can be considered as a trajectory reconstruction from the latent space, and $\mathbf{z}_2$ is an MoG parameterized by a neural network

$$p(\mathbf{z}_2|\mathbf{z}_1) = \sum_{k=1}^{K} \pi_k \mathcal{N}\big(\mathbf{z}_2|\mu_k(\mathbf{z}_1), \text{diag}\big(\sigma_k^2(\mathbf{z}_1)\big)\big) \tag{3}$$

$$p(\mathbf{T}|u, \mathbf{z}_1, \mathbf{z}_2) = \mathcal{N}\big(\mathbf{T}|\mu_\lambda(u, \mathbf{z}_1, \mathbf{z}_2), \text{diag}\big(\sigma_\lambda^2(u, \mathbf{z}_1, \mathbf{z}_2)\big)\big). \tag{4}$$

In (3), K is the number of components in the mixture and π_k is the mixture weight representing the prior probability of the kth component such that $\sum_k \pi_k = 1$. In our implementation, we set $\pi_k = 1/K$ to make it uniformly distributed. As a hierarchical VAE, a data sample $\mathbf{T}$ is generated from a nonlinear transformation $p(\mathbf{T}|u, \mathbf{z}_1, \mathbf{z}_2)$ conditioned on the latent variables, i.e., the partially observed label u, $\mathbf{z}_1$, and MoG $\mathbf{z}_2$. These latent variables are marginally independent and allow us, in the case of trajectory generation for example, to disentangle the generating user from the moving patterns of a particular trajectory. This MoG prior not only provides rich and multimodal interpretation of the latent variables—and correspondingly disentangles the factors governing human mobility from the trajectory semantics—but also prevents the KL term from pulling individual posteriors toward a simple prior, also known as the inactive stochastic units problem [27], [38].

2) Inference Model: In a fully unsupervised learning generative model, there is generally no guarantee that the inference on a mobility data set with N users will actually recover the trajectories belonging to N different individuals because of the unknown factors governing human mobility patterns. For example, the trajectories vary both in terms of users (whom each belongs to) and spatiotemporal patterns (where and when the trajectory is produced). This argument also holds for images, e.g., the images of handwritten digits vary both in terms of content (which digit is present) and style (how the digit is written) [28]. In recent studies [39], [40], researchers have demonstrated that purely unsupervised disentangled representation learning methods are brittle. In contrast, a few number of labeled trajectories makes the data inference significantly easier [41], and, more importantly, one can specify a disentangled factor governing the data generation. As the mobility data estimation of the unsupervised data is exactly the same as the conventional VAE [see (1)], we focus on the case where the user label u is observed.

Specifically, we use a two-layered inference model in STULIG

$$q(u, \mathbf{z}_1, \mathbf{z}_2|\mathbf{T}) = q(\mathbf{z}_2|\mathbf{T}, u, \mathbf{z}_1) q(z_1|\mathbf{T}) q(u|\mathbf{T}) \tag{5}$$

where $q(u|\mathbf{T})$ indicates the probability that trajectory $\mathbf{T}$ is generated by user u and acts as a classifier for TUL task that should be estimated; both $q(\mathbf{z}_1|\mathbf{T})$ and $q(\mathbf{z}_2|\mathbf{T}, u, \mathbf{z}_1)$ are conditional distributions that perform approximate inference and are parameterized by neural networks

$$q(\mathbf{z}_1|\mathbf{T}) = \mathcal{N}\big(\mathbf{z}_1|\mu_\psi(\mathbf{T}), \text{diag}\big(\sigma_\psi^2(\mathbf{T})\big)\big)$$
$$q(\mathbf{z}_2|\mathbf{T}, u, \mathbf{z}_1) = \mathcal{N}\big(\mathbf{z}_2|\mu_\xi(\mathbf{T}, u, \mathbf{z}_1), \text{diag}\big(\sigma_\xi^2(\mathbf{T}, u, \mathbf{z}_1)\big)\big). \tag{6}$$

Note that the inference model explicitly indicates a conditional dependence among the latent variables, which disentangles the partially observed identity u from the trajectory patterns $\mathbf{z}_2$, as shown in Fig. 1(b).

3) Objective: The generative model and the inference model act as decoder and encoder, respectively, and together define a probabilistic autoencoder. Therefore, our STULIG model is fit by maximizing on labeled mobility data

$$\log p(\mathbf{T}, u) = \log \iint q(\mathbf{z}_1, \mathbf{z}_2|\mathbf{T}, u) \frac{p(\mathbf{T}, u, \mathbf{z}_1, \mathbf{z}_2)}{q(\mathbf{z}_1, \mathbf{z}_2|\mathbf{T}, u)} d\mathbf{z}_1 d\mathbf{z}_2$$
$$\geq \mathbb{E}_{q(\mathbf{z}_1, \mathbf{z}_2|\mathbf{T}, u)} \log \frac{p(\mathbf{T}, u, \mathbf{z}_1, \mathbf{z}_2)}{q(\mathbf{z}_1, \mathbf{z}_2|\mathbf{T}, u)} \tag{7}$$
$$= \mathbb{E}_{\mathbf{z}_1 \sim q(\mathbf{z}_1|\mathbf{T}), \mathbf{z}_2 \sim q(\mathbf{z}_2|\mathbf{T}, u, \mathbf{z}_1)} \log \frac{p(\mathbf{T}, u, \mathbf{z}_1, \mathbf{z}_2)}{q(\mathbf{z}_1, \mathbf{z}_2|\mathbf{T}, u)} \tag{8}$$

which is obtained via Jensen inequality for multiple variables [42]. We note that in (7), the factorization $q(\mathbf{z}_1, \mathbf{z}_2|\mathbf{T}, u) = q(\mathbf{z}_1|\mathbf{T}) q(\mathbf{z}_2|\mathbf{T}, u, \mathbf{z}_1)$ is implicitly assumed—however, it can be easily verified based on the inference network depicted by the graphical model in Fig. 1(b).

Now, we can derive the ELBO $\mathcal{L}_l$ for **labeled** data in STULIG as

$$\begin{aligned}\mathcal{L}_l &= \mathbb{E}_{\mathbf{z}_1, \mathbf{z}_2}\left[\log \frac{p(\mathbf{T}, u, \mathbf{z}_1, \mathbf{z}_2)}{q(\mathbf{z}_2|\mathbf{T}, u, \mathbf{z}_1)} - \log q(\mathbf{z}_1|\mathbf{T})\right] \\ &= \mathbb{E}_{\mathbf{z}_1, \mathbf{z}_2}[\log p(\mathbf{T}|u, \mathbf{z}_1, \mathbf{z}_2) + \log p(u) + \log p(\mathbf{z}_1) \\ &\quad + \log p(\mathbf{z}_2|\mathbf{z}_1) - \log q(\mathbf{z}_2|\mathbf{T}, u, \mathbf{z}_1) - \log q(\mathbf{z}_1|\mathbf{T})] \\ &= \mathbb{E}_{\mathbf{z}_1, \mathbf{z}_2}[\log p(\mathbf{T}|u, \mathbf{z}_1, \mathbf{z}_2)] + \log p(u) \\ &\quad - \mathbb{KL}[q(\mathbf{z}_1|\mathbf{T})||p(\mathbf{z}_1)] - \mathbb{KL}[q(\mathbf{z}_2|\mathbf{T}, u, \mathbf{z}_1)||p(\mathbf{z}_2|\mathbf{z}_1)]\end{aligned} \tag{9}$$

where the first term is the reconstruction cost, encouraging the model to encode the trajectory data into a set of latent variables $\mathbf{z}_1$ and $\mathbf{z}_2$, combined with partially observed identity u, which can efficiently reconstruct the data, and the two KL terms are regularizers that encourage the inferred latent factors [see (6)] to match the two priors—isotropic multivariate Gaussian and MoG, respectively.

As for the unlabeled mobility data, the user identity is predicted by performing posterior inference with the classifier $q(u|\mathbf{T})$, that is, we consider u as a latent factor and have the following ELBO:

$$\begin{aligned}\log p(\mathbf{T}) \geq\ & \mathbb{E}_{\mathbf{z}_1 \sim q(\mathbf{z}_1|\mathbf{T}), \mathbf{z}_2 \sim q(\mathbf{z}_2, u|\mathbf{T}, \mathbf{z}_1)} \\ & \times [\log p(\mathbf{T}|u, \mathbf{z}_1, \mathbf{z}_2) + \log p(u) + \log p(\mathbf{z}_1) \\ & + \log p(\mathbf{z}_2|\mathbf{z}_1) - \log q(\mathbf{z}_2, u|\mathbf{T}, \mathbf{z}_1) \\ & - \log q(\mathbf{z}_1|\mathbf{T})]\end{aligned} \tag{10}$$

$$\begin{aligned}=\ & \mathbb{E}_{\mathbf{z}_1 \sim q(\mathbf{z}_1|\mathbf{T}), \mathbf{z}_2 \sim q(\mathbf{z}_2|\mathbf{T}, u, \mathbf{z}_1)} \\ & \times (q(u|\mathbf{T})\mathcal{L}_l - q(u|\mathbf{T}) \log q(u|\mathbf{T}))\end{aligned} \tag{11}$$

$$= \sum_u q(u|\mathbf{T})\mathcal{L}_l + \mathcal{S}(q(u|\mathbf{T})) \triangleq \mathcal{L}_u \tag{12}$$

where (11) is obtained with the factorization $q(\mathbf{z}_2, u|\mathbf{T}, \mathbf{z}_1) = q(\mathbf{z}_2|\mathbf{T}, u, \mathbf{z}_1) q(u|\mathbf{T})$, and $\mathcal{S}(q(u|\mathbf{T}))$ is the information entropy of $q(u|\mathbf{T})$. The loss of the classifier $q(u|\mathbf{T})$ during training is measured by L_2 reconstruction error between the

predicted identity and the pseudolabel. Here, we need to evaluate the generative likelihood for each class during training, meaning that we assume a pseudolabel (e.g., a particular user, in this case) and calculate its loss in each iteration. We repeat this process for all classes, similar to the semi-VAE-based image [36] and text classification [22].

Finally, we have the ELBO $\mathcal{L}^{\text{STULIG}}$ on the entire data set

$$\mathcal{L}^{\text{STULIG}} = -\sum_{(\mathbf{T},u)\sim D_l} (\mathcal{L}_l + \alpha \log q(u|\mathbf{T})) - \sum_{\mathbf{T}\sim D_u} \mathcal{L}_u \quad (13)$$

where the first RHS term includes an additional classification loss of classifier $q(u|\mathbf{T})$ when learning from the labeled data, and hyperparameter α controls the relative strength of the labeled data. Therefore, instead of directly performing maximum likelihood estimation on the intractable marginal log-likelihood, training is done by maximizing the tractable ELBO $\mathcal{L}^{\text{STULIG}}$.

C. Semisupervised Trace Discriminating

1) Efficiency Issues: Note that the extra probability $q(u|\mathbf{T})$ in the labeled data [see (13)] is similar to previous semi-VAE-based models [6], [22], [36]. However, training classifier $q(u|\mathbf{T})$ only on D_u (unlabeled data) is biased and cannot be accurately verified without the labels. In contrast, the classification loss is easily estimated with the labeled data and, therefore, can be leveraged to construct the best knowledge of the classifier through the entire data set if the distribution of labeled data and unlabeled data is consistent, i.e., the learned classifier on labeled data can be generalized to data without labels.

In the previous works [6], [22], [36], a classifier $q(u|\mathbf{T})$ is trained to predict the labels of unlabeled data D_u. However, since one cannot directly evaluate the loss on D_u, the predicted label probabilities are employed as the (normalized) weight to label the data and iteratively compare them with the pseudolabels. More specifically, let $\mathbf{u} \in \mathbb{R}^{1\times N}$ be the user vector, and its ith value $u_i = 1$ indicates that the (predicted) label for a trajectory $\widetilde{\mathbf{T}}$ is u_i. Then, one can predict its label with a softmax function and obtain the probabilities of label distribution $\mathbf{p} = p_1, \ldots, p_N$, meaning that $\widetilde{\mathbf{T}}$ has p_j probability generated by user u_j. Thus, it is easy to calculate the loss between the predicted label distribution $\mathbf{p}$ and the pseudolabels to train the unlabeled data as in (12). The major drawbacks of this evaluation on unlabeled data are that the computational overhead is expensive, especially for a larger number of labels N. For example, it requires $N \times M$ evaluations for M trajectories in a single epoch during training.

2) Geographical Regularization: In this work, we present a simple but efficient method to overcome the limitations of previous semi-VAE models. The motivation is that if the number N of labels can be largely reduced in each epoch, we can significantly reduce the training time—since the number M of trajectories is constant. Fortunately, this goal can be achieved by leveraging the geographical distribution of human trajectories. In fact, it has been observed that users' mobility preference is constrained by geographical distance [43], [44], i.e., users prefer to visit nearby POIs and the activity of most (if not all) users is within a small number of regions, which is also observed in our experiments—the detailed description and discussions can be found in Section V.

More specifically, for each user $u \in \mathcal{U}$, we first cluster his/her trajectories (labeled) into R regions based on their locations using DBSCAN [45], where R varies from user to user. Then, we can build an index table for all unlabeled trajectories $\mathcal{T}$ such that each $\widetilde{\mathbf{T}} \in \mathcal{T}$ is (at most) associated with Q users, who have labeled historical trajectories located within the region that $\widetilde{\mathbf{T}}$ belongs to—which can be simply calculated based on the distance between the center of the region and the center of the trajectory. Therefore, we incorporate the prior knowledge of such geographical distribution to limit the possible labels Q of each unknown trajectory such that $Q \ll N$. As we will show in Section V, our model can not only save considerable training time but also alleviates the biased estimation of unlabeled trajectories by leveraging the geographical features of human mobility.

D. Implementation Issues

We now discuss the implementation details of the STULIG model and its variant STUL.

1) Trajectory Convolution: Instead of using RNNs as most previous work in modeling human mobility [2], [5], [6], [12], [13], [46], [47], we use CNNs as the implementation of encoder–decoder. Recall that we segment each trajectory every 6 h and treat it as a 2-D matrix where, as a consequence, the temporal periodicity has been embedded into the matrix. In this way, it is easy for our model to capture the multilevel periodic patterns of human mobility with the convolutional operations. This choice is motivated by the fact that a growing number of works have successfully replaced RNNs, partially or entirely, with CNNs in some important tasks where RNNs were dominant for a long while. For example, convolutional seq2seq [48] explored CNNs to encode/decode sentences and achieves remarkable results on machine translation. In this spirit, recent work [49], [50] shows empirically and theoretically that there is an equivalence between recurrent models and feedforward architectures. Another reason for this choice is that RNN models are often more delicate to tune and more brittle to train, accompanied with significant computational burden, in comparison with the standard feedforward architectures such as CNNs.

2) Variant: We also consider a variant of STULIG, called semisupervised TUL (STUL), which is similar to STULIG except that it only uses an isotropic Gaussian as the prior of latent variable $\mathbf{z}$. The reason for considering this variant is to investigate: 1) the disentangled representation ability of STULIG and 2) the advantage of our models, in terms of both effectiveness (the TUL performance) and learning efficiency (compared to previous RNN-based methods).

3) Attention: In order to investigate the performance of STULIG, we add a deterministic attention mechanism [51] into our two models. Note that the existing works have found that the attention mechanism itself is powerful enough to capture sequence information and thus results in a useless

TABLE I
DATA DESCRIPTION

Dataset	$\|\mathcal{U}\|$	$\|\mathcal{T}_n\|/\|\mathcal{T}_e\|$	$\|C\|$	$\overline{\mathcal{R}}$	$\mathcal{T}_r$
Foursquare	270	12,800/12,928	7,195	242	[1,35]
	109	5,312/5,376	4,227	246	[1,35]
Brightkite	92	9,920/9,984	2,123	471	[1,184]
	34	4,928/4,992	1,359	652	[1,44]
Gowalla	201	9,920/10,048	10,958	219	[1,131]
	112	4,928/4,992	6,683	191	[1,95]

TABLE II
ARCHITECTURES OF ENCODER AND DECODER. CONV: CONVOLUTION. MP: MAX POOLING. FC: FULLY CONNECTED

Layer		Maps Size	Kernel	Stride	Activation	Padding
Input	1	256 × 324	–	–	–	–
Conv	16	256 × 324	5 × 5	1	ReLU	'SAME'
MP	16	128 × 162	2 × 2	2	–	'VALID'
Conv	32	128 × 162	5 × 5	1	ReLU	'SAME'
MP	32	64 × 81	2 × 2	2	–	'VALID'
Conv	16	64 × 81	5 × 5	1	ReLU	'SAME'
MP	16	32 × 40	2 × 2	2	–	'VALID'
FC	–	–	–	–	softmax	

variational latent space [52]. However, we surprisingly observe that the attention mechanism only affects STUL but has little effects on the performance of STULIG—which empirically demonstrates, at least to some extent, the capabilities of the structural disentangled representation of STULIG.

4) Generating Synthetic Trajectories: We adapted the VAE to human mobility learning by using convolutional seq2seq for the encoder and the decoder, essentially forming a sequence autoencoder with the MoG prior acting as a regularizer on the hidden factors. The decoder, therefore, serves as a special trajectory generator model, conditioned on the hidden factors. We will show that the proposed generative model has the potential to model the human-mobility generating distribution. Besides, it allows us to train a generative model on the original data and only publish the synthetic data, so as to preserve the location privacy of users.

V. EXPERIMENTAL EVALUATION

We now present the results of our experiments using three real-world data sets regarding the ability to discriminate motion traces, learn disentangled representations, as well as generate plausible synthetic traces. To ease the reproducibility of our results, we have made the source code publicly available.[1]

Data Sets: We conduct all the experiments on three publicly available LBSN data sets: Gowalla,[2] Foursquare,[3] and Brightkite.[4] We prepare the data set for our needs in two phases. We first randomly select $|\mathcal{U}|$ users and their corresponding trajectories from the data sets for evaluation. Subsequently, for each data set, we select two different sets of users (i.e., the labels of trajectories) for model robustness check. For each user, we randomly select half of her trajectories for training and the rest for testing. When training the semisupervised models, the testing data are treated as unlabeled data. Table I shows the basic statistics of the data sets, where $|\mathcal{U}|$ is the number of users, $|\mathcal{T}_n|/|\mathcal{T}_e|$ is the number of trajectories for training/testing, $|C|$ is number of check-ins, $\overline{\mathcal{R}}$ is the average length of trajectories (before splitting), and $\mathcal{T}_r$ denotes the range of the trajectory length (after splitting).

A. Motion Discrimination and Interpretability

TUL Baselines: We compare STUL and STULIG with several state-of-the-art approaches from the field of RNN-based human-mobility classification and prediction. We omit the comparison to traditional trajectory classification methods such as SVM, decision tree, and long common subsequence (LCSS) that have been demonstrated inferior to TULERs [5] and TULVAE [6]. The baselines can be broadly grouped as follows.

1) *RNN-Based TUL:* Including HTULER-L, TULER-GRU, TULER-LSTM-S, TULER-GRU-S, and Bi-TULER proposed in [5]—the first set of methods for TUL.
2) *Hierarchical RNN-Based TUL:* Including TULER-LSTM, HTULER-G, HTULER-B, TULER-GRU-S, and Bi-TULER proposed in [6], respectively, implemented with the hierarchical LSTM, GRU and Bi-LSTM.
3) *TULVAE [6]:* The state-of-the-art TUL method using bidirectional LSTM as the encoder–decoder and isotropic multivariate Gaussian as the prior of latent space.

Model Parameters: The learning rate of all models is initialized with 0.001 and decays with a rate of 0.9. The activation function for all methods is ReLU, and the dropout rate is set to 0.5, while the batch size is 64 for all RNN-based models (following [6]) and 32 for STUL and STULIG. We embed each POI id into a 250-D vector. We represent the geographical location of each POI as a 50-D vector after mapping its longitude and latitude value through an FCN, and use a 24-D one-hot vector to embed the timestamps. Therefore, each POI is represented as a 324-D vector. Furthermore, we use 300 neuron units for the classifier and 512 units for both encoders and decoders for all the models. The dimensions of $\mathbf{z}_1$ and $\mathbf{z}_2$ of STULIG are 128 and 50, respectively, and the dimensions of $\mathbf{z}$ in STUL and TULVAE is 256, which ensures that the size of the input in all decoders is the same across the three models. The parameters K and α for STULIG are set to 15 and 0.3, respectively, unless otherwise specified. For STUL and STULIG, both the encoder and the decoder are three-layer CNN architectures, as described in Table II.

TUL Metrics: We use the standard ACC@K (K = 1, 5), macro-P, macro-R, and macro-F1 to evaluate TUL performance for all methods, following the previous works [5], [6].

1) Performance Comparison: Table III summarizes the performance comparisons among the proposed methods and all the baselines on three data sets. The improvements of STULIG over the compared algorithms all passed the paired t-tests with a significance value $p < 0.01$. One can observe that STULIG consistently outperforms all the baselines (as

[1]https://github.com/gcooq/STULIG
[2]http://snap.stanford.edu/data/loc-gowalla.html
[3]https://sites.google.com/site/yangdingqi/home
[4]http://snap.stanford.edu/data/loc-brightkite.html

TABLE III

TUL PERFORMANCE COMPARISON AMONG DIFFERENT METHODS ON THREE DATA SETS. THE BEST METHOD IS SHOWN IN BOLD, AND THE SECOND BEST IS SHOWN AS UNDERLINED. $*$ INDICATES THE STATISTICAL SIGNIFICANCE $P < 0.001$ COMPARED WITH THE BEST BASELINE METHOD BASED ON THE PAIRED T-TEST

Dataset	Method \ Metric	ACC@1	ACC@5	macro-P	macro-R	macro-F1	ACC@1	ACC@5	macro-P	macro-R	macro-F1
Brightkite		$\|\mathcal{U}\|$=34					$\|\mathcal{U}\|$=92				
	TULER-LSTM	48.26%	67.39%	49.90%	47.20%	48.51%	43.01%	59.84%	38.45%	35.81%	37.08%
	TULER-GRU	47.84%	67.42%	48.88%	46.87%	47.85%	44.03%	61.36%	38.86%	36.47%	37.62%
	TULER-LSTM-S	47.88%	67.38%	48.81%	47.03%	47.62%	44.23%	61.00%	38.02%	36.33%	37.16%
	TULER-GRU-S	48.08%	68.23%	48.87%	46.74%	47.78%	43.93%	61.85%	37.93%	36.01%	36.94%
	Bi-TULER	48.13%	68.17%	49.15%	47.06%	48.08%	43.54%	60.68%	38.20%	36.47%	37.31%
	HTULER-L	49.44%	71.13%	51.51%	47.31%	49.32%	45.26%	63.55%	41.61%	38.13%	39.79%
	HTULER-G	49.12%	70.81%	51.85%	46.88%	49.24%	44.50%	63.17%	41.10%	37.51%	39.22%
	HTULER-B	49.78%	70.69%	51.03%	46.18%	48.48%	45.30%	63.93%	41.82%	39.32%	38.60%
	TULVAE	49.82%	71.71%	51.26%	46.43%	48.72%	45.92%	64.84%	43.15%	39.65%	41.32%
	STUL	49.76%	70.12%	51.18%	46.54%	48.75%	45.66%	64.03%	42.84%	39.27%	41.03%
	STULIG	***50.44%**	***73.04%**	***52.45%**	***47.98%**	***50.12%**	***47.21%**	***68.34%**	***44.21%**	***39.92%**	***41.96%**
Foursquare		$\|\mathcal{U}\|$=109					$\|\mathcal{U}\|$=270				
	TULER-LSTM	57.24%	69.27%	49.35%	47.61%	48.46%	50.69%	62.11%	46.27%	41.84%	43.95%
	TULER-GRU	56.85%	69.40%	49.05%	47.34%	48.18%	50.65%	62.68%	46.38%	41.65%	43.89%
	TULER-LSTM-S	57.14%	69.57%	48.48%	47.59%	48.03%	49.55%	62.65%	43.40%	42.11%	42.75%
	TULER-GRU-S	56.31%	69.56%	49.04%	46.98%	47.99%	50.21%	62.33%	46.17%	41.01%	43.44%
	Bi-TULER	58.31%	71.17%	50.84%	48.88%	49.84%	52.31%	64.03%	47.15%	44.95%	46.03%
	HTULER-L	56.66%	71.46%	48.33%	47.28%	47.80%	51.59%	65.53%	45.82%	44.06%	44.92%
	HTULER-G	55.92%	71.37%	48.10%	46.47%	47.27%	51.46%	65.15%	45.34%	43.03%	43.97%
	HTULER-B	59.10%	72.40%	51.37%	49.85%	50.03%	54.91%	67.76%	48.94%	47.82%	48.37%
	TULVAE	59.91%	73.60%	53.59%	50.93%	52.23%	55.54%	68.27%	51.07%	48.63%	49.83%
	STUL	61.29%	75.45%	***54.52%**	51.28%	52.85%	56.35%	70.56%	***52.90%**	49.07%	50.91%
	STULIG	***61.98%**	***75.83%**	54.45%	***51.95%**	***53.17%**	***57.77%**	***71.28%**	52.67%	***50.12%**	***51.36%**
Gowalla		$\|\mathcal{U}\|$=112					$\|\mathcal{U}\|$=201				
	TULER-LSTM	41.79%	57.89%	33.61%	31.33%	32.43%	41.24%	56.88%	31.70%	28.60%	30.07%
	TULER-GRU	42.61%	57.95%	35.22%	32.69%	33.91%	40.85%	57.31%	29.52%	27.80%	28.64%
	TULER-LSTM-S	42.11%	58.01%	33.49%	31.97%	32.71%	41.22%	57.70%	29.34%	28.68%	29.01%
	TULER-GRU-S	41.35%	58.45%	32.51%	31.79%	32.15%	41.07%	57.49%	29.08%	27.17%	28.09%
	Bi-TULER	42.67%	59.54 %	37.55%	33.04%	35.15%	41.95%	57.58%	32.15%	31.66%	31.90%
	HTULER-L	43.89%	60.90%	35.95%	33.14%	35.12%	43.40%	60.25%	34.43%	33.63%	34.02%
	HTULER-G	43.33%	60.74%	37.71%	31.74%	36.01%	42.88%	59.41%	32.72%	32.54%	32.63%
	HTULER-B	44.21%	62.28%	36.48%	33.51%	34.93%	44.50%	60.93%	34.89%	34.46%	34.67%
	TULVAE	44.14%	64.28%	40.02%	32.64%	35.96%	45.02%	62.37%	36.01%	34.13%	35.24%
	STUL	44.54%	65.01%	40.36%	34.32%	37.09%	45.14%	62.31%	36.54%	34.43%	35.56%
	STULIG	***47.98%**	***68.02%**	***42.31%**	***34.47%**	***37.99%**	***45.70%**	***64.40%**	***38.41%**	***34.94%**	***36.59%**

well as STUL) for all metrics in all data sets. These results imply that the proposed CNN-based semisupervised model with structural MoG is more effective than the state-of-the-art RNN-based human-mobility learning models, as well as the regular VAE-based models. We note that the improvement over RNN-based encoder–decoder methods and regular VAE-based models (e.g., TULVAE and STUL) is less significant, especially for the smaller size of data sets (e.g., $|\mathcal{U}| = 34$ in Brightkite). This, in a sense, explains the "bypassing" problem of attention-based encoder–decoder models, i.e., variational latent space does not need to learn much as the attention mechanism alone can capture enough sequence information. However, when examining the discrepancy between our two models, we can quantitatively obtain the TUL performance gain of the proposed disentangled representation learning model, which demonstrates the ability of the STULIG model for addressing the bypassing issue.

If we inspect the performance between STUL—a simplified version of STULIG without disentangled latent space—versus TULVAE, we can observe that they achieve almost the same performance (usually exhibiting the second-best performance). This confirms our motivation that CNN-like feedforward methods can achieve comparable performance on the TUL task with RNN-based models. While RNN is slightly better on sequential pattern learning, CNN is more effective on learning periodical patterns (quantitative investigation of the two different mobility patterns is still an open question and is part of the future work). We utilized the convolutional seq2seq model instead of the autoregressive encoder–decoder in TULVAE. This choice enables our models, to some extent, avoiding the "posterior collapse" problem, i.e., the model can ignore the latent variables if autoregressive encoder–decoder is expressive enough to model the data density, resulting in a trivial posterior that collapses to the prior [51], [53]. To overcome this problem, the existing methods either introduce annealing factor to weaken the KL terms in (1) [51], [52] or augment the objective, so it does not only maximize the likelihood of the data [53], [54], both of which are often challenging to tune and highly sensitive to hyperparameters. This problem would be aggravated by incorporating attention mechanism [52], which, however, has not been observed in our models—albeit the attention mechanism plays less important role on improving the TUL performance of STULIG.

As part of our work, we investigated the interpretability of mobile traces learning and the explicit independence of disentangled representation of our models. Fig. 2(a) shows the latent space of our STULIG model, which clearly displays the learned $\mathbf{z}_2$ as MoG. In contrast, Fig. 2(b) shows the learned $\mathbf{z}$

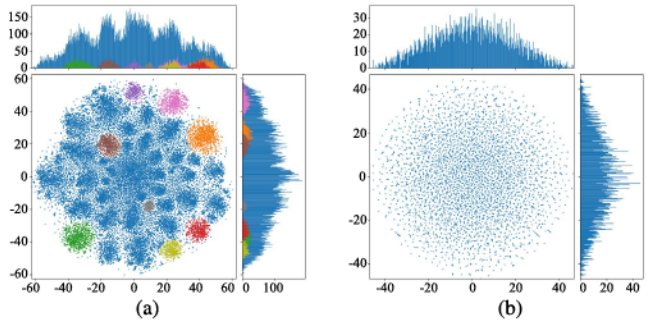

Fig. 2. Latent space visualization using t-SNE ($|\mathcal{U}| = 112$ in Gowalla). (a) Learned latent space of STULIG $\mathbf{z}_2 \sim q(\mathbf{z}_2|\mathbf{T}, u, \mathbf{z}_1)$ ($K = 15$), colored components indicating normal distributions. (b) Learned latent space of TULVAE.

of TULVAE which is isotropic Gaussian prior based. These results demonstrate the capability of our STULIG model on learning multimodal and complex representations. Although the learned latent space is more distinct and expressible, it does not mean that the trajectories of the same user are clustered together. Instead, we conjecture that the trajectories sharing similar motion patterns should be closely clustered. However, we, respectively, note that such "conceptual" motion patterns are highly subjective and they even lack qualitative measures—unlike image and natural language settings, where one can directly apply visual or linguistic-based understanding. Therefore, we recognize as open research question left as our future work how to explain the representations of human-mobility data captured by the latent factor models.

STULIG achieves the disentangled embedding with the prior knowledge of spatiotemporal features of user mobility and partially specified data labels (users) rather than the unsupervised disentangled learning in previous works [24], [55]. Interestingly, our results are coincident with a very recent comprehensive study [39], which has theoretically and empirically proved that unsupervised learning of disentangled representations is fundamentally impossible without inductive biases. In contrast, introducing a few of supervision can reliably learn disentangled representations of data [41]. Furthermore, although in STULIG we do not explicitly distinguish the spatial and temporal factors, which are tightly coupled in human mobilities, it can be easily extended to incorporate additional latent factor(s) for each axis, as many existing works have done [24], [28].

2) Efficiency Notes: Another advantage of the proposed methods is the model efficiency, which benefits from two parts: RNN-free trajectory modeling and leveraging geographical distribution for unlabeled data. In this context, CNN-based methods (STUL and STULIG) require significantly less training time compared with TULVAE, an RNN-based TUL method. This is demonstrated by the results shown in Fig. 3(a), where we report the time of the first 20 training epochs for the four models—with a note that we omit the other RNN-based methods due to their similar performances to TULVAE. Apparently, STULIG incurs a slightly larger overhead on structural mixture Gaussians approximation compared with the isotropic Gaussian prior-based method STUL. This figure also illustrates

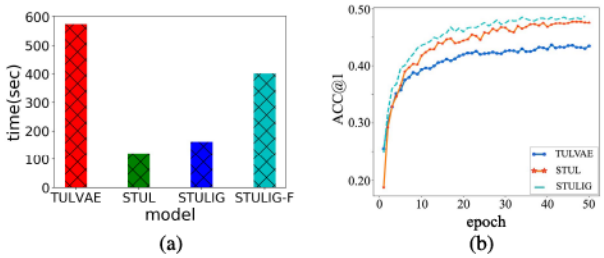

Fig. 3. Efficiency comparison on the Gowalla data. (a) Time (20 epochs). (b) Training convergence.

the efficiency improvement in constraining the possible users for unlabeled data, where the variant STULIG-F was implemented by replacing the semi-VAE training in STULIG with the complete iterations with all N users as in TULVAE.

Complementary to the time per epoch, Fig. 3(b) plots the training process of three models, which clearly shows that our methods converge faster than RNN-based TULVAE. One can alternate the convolutional seq2seq model in STULIG with fully attentional feedforward architectures, such as the transformer model [56]. However, due to that the data used in the experiments are relatively small and are very sparse, we were unable to reap significant gains using the self-attention mechanism in our experiments. Another possible improvement on larger LBSN data sets is to pretrain a model for embedding the locations considering both forward and backward semantics using the masked encoder–decoder architecture such as the remarkable language representation model BERT [57], which is beyond the scope of this work.

3) Parameter Sensitivity: We investigated the impact of two important parameters of STULIG: K and α, respectively, denoting the number of mixture components of $\mathbf{z}_2$ and the weight of classifier $q(u|\mathbf{T})$. Intuitively, more components (i.e., larger value of K) may increase the ability of STULIG to approximate more complex distribution of $\mathbf{z}_2$. However, beyond a certain threshold [e.g., 15 in Fig. 4(a)], increasing K results in overfitting problem. Theoretically, α plays an important role in balancing the weight of learning from labeled data and unlabeled data, which is empirically decided in our implementation. However, it seems that the performance of STULIG is not sensitive to α when the value is in the range of [0.1, 1], as shown in Fig. 4(b). This is because the trajectory distribution of each user is relatively stable—recall that we randomly select 50% trajectory for testing. This result also implies that our assumption is reasonable, i.e., the classifier $q_\phi(u|\mathbf{T})$ learned from labeled data can be used for constructing the knowledge of unlabeled data if the data distribution is nearly consistent [see (13)].

B. On Generating Plausible Traces

We also designed experiments to quantitatively validate our models in terms of generating plausible trajectories and quantify the privacy-preserving impact.

1) Quantitative Evaluation: Toward quantifying the generation, we conduct experiments to compare STULIG with several trajectory synthesis methods, including the following.

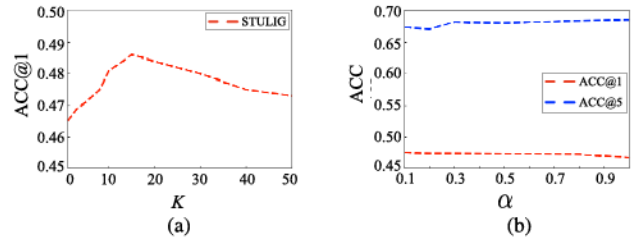

Fig. 4. Impact of parameters. $|\mathcal{U}| = 112$ in Gowalla data. (a) Impact of K. (b) Impact of α.

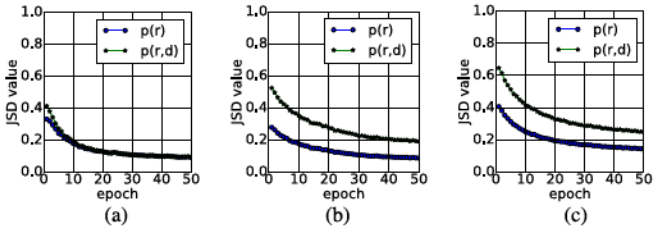

Fig. 5. JSD convergence with respect to the training epochs. (a) Brightkite. (b) Foursquare. (c) Gowalla.

1) *MC:* A stochastic model describing a sequence of locations in which the probability of each location depends only on the state attained in the previous location, which has been used for trajectory generation in [58].
2) *HMM:* A classical dynamic Bayesian network in which the system being modeled is assumed to be a Markov process with unobserved hidden states, which has been used for generating human traces in [59].
3) *Deep Transport (DTran) [60]:* An LSTM-based mobility generation model maximizing the log-likelihood of location transition in a trajectory.
4) *T-WGAN [31]:* A most recent nonparametric trajectory synthesis method using Wasserstein GAN with gradient penalty [32]. The trajectories are embedded into a 2-D matrix, where each cell of the matrix corresponds to one POI and contains information about the time and duration of visiting.
5) *TULVAE [6]:* As a deep generative model, TULVAE is capable of generating synthetic mobility traces, and the main difference from STULIG is the learned latent factors.

Note that we omit the comparison to the trajectory synthesis method in [30] because it requires the semantics of the check-ins (e.g., categorical information) which are missing in our data sets. Following [31], we evaluate the generation quality of models by the Jensen–Shannon divergence (JSD) between the aggregate real R and generated G trajectory distribution:

$$\text{JSD}(R||G) = \frac{1}{2}\mathbb{KL}(R||M) + \frac{1}{2}\mathbb{KL}(G||M) \tag{14}$$

where $M = (R + G)/2$, and trajectory distribution R and G are marginal distributions measured by either one of the two different metrics [31]: 1) $p(c)$—which measures the probability of a check-in $c_{i,u}$ at time t_i, implying the personal check-in preference (both spatial and temporal) of user u and 2) $p(c, d)$— which measures the probability of visiting a location $c_{i,u}$ for a duration d, reflecting the staying patterns and interests of user u in difference places.

As the previous experiments, we used 50% of the trajectories of each user for training and another 50% as the test data. In addition, we generated the same number and the same length of synthetic trajectories for each user and measured the mean JSD between the generated trajectories and the real trajectories in test data. Fig. 5 shows the convergence of the JSD measure on three data sets.

TABLE IV
JSD RESULTS ON THREE DATA SETS, WHERE THE $p(c)/p(c, d)$ MEASURES ARE REPORTED

Method	Foursquare	Brightkite	Gowalla
MC	0.2686/0.3127	0.2371/0.1748	0.3845/0.4204
HMM	0.2647/0.3011	0.2226/0.1616	0.3735/0.4043
DTran	0.2402/0.2805	0.1942/0.1436	0.3522/0.3996
T-WGAN	0.2321/0.2834	0.1945/0.1445	0.3527/0.4009
TULVAE	0.1893/0.2262	0.1430/0.1274	0.2549/0.3428
STULIG	**0.0873/0.1907**	**0.0857/0.0907**	**0.1474/0.2544**

The comparison results are shown in Table IV. Obviously, STULIG has a significant advantage on generating plausible trajectories compared with baselines. Unsurprisingly, Bayesian models, e.g., TULVAE and STULIG, achieve better results than GAN-based model W-TGAN since its training procedure relies on backpropagation through the discriminator into the generator, which is not applicable for discrete data, such as trajectories and languages. Although the gradient can be estimated with an alternative approach such as REINFORCE policy, GAN training combined with deep reinforcement learning has proved to be problematic, e.g., unstable and largely biased. This result is coincident with recent observations [33] that GAN-based generative models usually perform worse even than maximum likelihood estimation models, such as LSTM and MC on sequential data.

2) STULIG and Privacy: JSD measure alone may be misleading due to the dilemma between data sharing and privacy preserving: while collected data can be used to offer important social, economic, and democratic services and facilitate much needed research, sharing real data records carries privacy risks. To demonstrate that STULIG can generate plausible trajectories while preserving mobility patterns of user traces, we generated the same number of synthetic trajectories for each person using the STULIG model and then evaluated its TUL performance on mixed data sets (see Table V).

1) *T1 (Fully Synthetic Trajectories):* We replace the training and testing data (real traces) with the generated trajectories.
2) *T2 (Infusion of Synthetic Data):* We inject the same number of synthetic data into original training/testing set.
3) *T3 (Augmenting Training Set With Synthetic Data):* We augment the training data with synthetic trajectories, but still use the real traces for testing.

TABLE V

RESULTS ON PRIVACY-PRESERVING TRAJECTORY DATA. THE TWO NUMBERS IN PARENTHESES ARE THE PERCENTAGES OF INCREASES/DECREASES COMPARED WITH THE RESULTS OF STUL AND TULVAE, RESPECTIVELY. HERE, $|\mathcal{U}| = 92, 270, 201$ FOR BRIGHTKITE, FOURSQUARE, AND GOWALLA, RESPECTIVELY

		ACC@1 (%)	ACC@5 (%)	macro-F1 (%)
Brightkite	T1	45.69 (+0.03, -0.23)	63.71 (-0.32, -1.13)	39.76 (-1.27, -1.56)
	T2	46.23 (+0.57, +0.31)	63.86 (-0.17, -0.98)	41.01 (-0.02, -0.31)
	T3	46.29 (+0.63, +0.37)	64.13 (+0.10, -0.73)	40.03 (-1.00, -1.29)
Foursquare	T1	54.83 (-1.52, -0.71)	67.07 (-3.49, -0.23)	48.77 (-2.14, -1.06)
	T2	55.17 (-1.18, -0.37)	67.51 (-3.05, -0.76)	48.96 (-1.95, -0.87)
	T3	55.61 (-0.74, +0.07)	67.30 (-3.26, -0.97)	49.41 (-1.50, -0.42)
Gowalla	T1	47.47 (+2.33, +2.45)	63.78 (+1.47, +1.41)	36.62 (+1.06, +1.38)
	T2	47.23 (+2.09, +2.21)	63.12 (+0.81, +0.75)	36.74 (+1.18, +1.50)
	T3	46.22 (+1.08, +1.20)	61.96 (-0.35, -0.40)	35.83 (+0.27, +0.63)

While looking similar, the three experiments reflect different aspects/levels of preserving the privacy of user locations. Table V shows that the TUL performance slightly decreases in most metrics on Brightkite and Foursquare but increases on Gowalla to a certain extent for T1. This result is managerially attractive because it allows researchers to access the full synthetic data, rather than some limited set of statistics such as certain counting queries or histograms, for achieving acceptable utility in various analytics and downstream machine learning tasks [30], [61]. In other words, the results on T2 and T3 imply that the performance of machine learning task, such as TUL, could be maintained at the same level when the real traces are obfuscated with well-generated synthetic data. We note that there is a subtle difference between T2 and T3. while both augmenting the training data with synthetic samples, T3 validates the trained model only on real data. This makes it particularly appealing for collaboratively building a machine learning model (e.g., crowdsourced) without sharing the sensitive validation data. Another potential application of the presented generative model is to augment the training (and testing) data set with the generated data, especially for those very sparse data in LBSN that we used (e.g., the density of check-ins from Foursquare and Gowalla are usually around 0.1% [6]).

However, these results raise another security problem; while the generative models, including our STULIG, are capable of generating plausible trajectories to camouflage the real locations of users, they do not promise the data anonymization. In fact, as models that can discriminate the human traces, STULIG, as well as STUL, would increase the risk of deanonymization attack. To ensure the privacy preserving of user identity, rather than location privacy, one should rely on more sophisticated approaches to sanitize sensitive information associated with the data sets. In this spirit, differential privacy [62] and its many applications, such as differentially private SGD [63] and plausible deniability [29], are worthwhile to be investigated for protecting against recovering private information from the published (synthetic) data. More broadly,

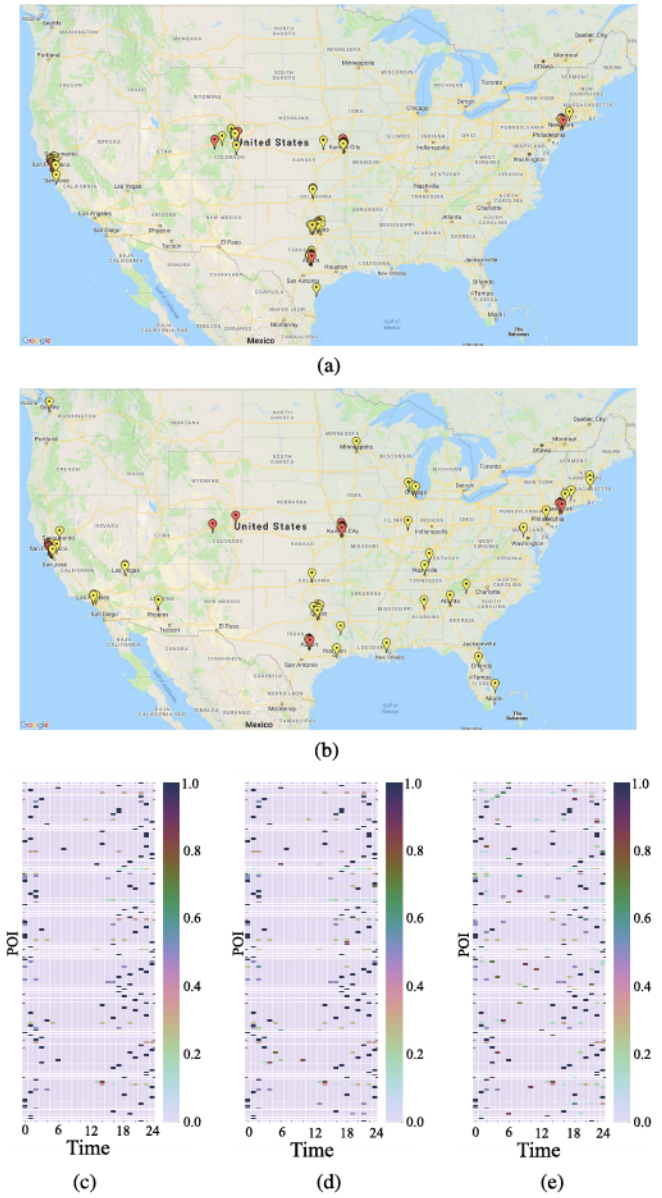

Fig. 6. Visualization of trajectory generation. Red and yellow flags in (a) and (b) denote the locations from the real and generated trajectories, respectively (to avoid clutter, we did not draw all the sequential line- segments traveled). In (c), (d), and (e), the color codes represent the check-in frequency, i.e., the darker the color, the more check-ins in that POI (with accompanying normalized numerical values with respect to the maximal visits for a given hour). (a) and (d) STULIG. (b) and (e) TULVAE. (c) Ground truth.

it is of interest to investigate the performance of other mobility learning problem in addition to TUL, such as POI recommendation/prediction, by (partially) leveraging the synthetic data, e.g., aiming at fine-grained location recommendations with differential privacy protection [64].

3) Qualitative Samples: We randomly selected a user in the Gowalla data set to visualize the impact of the generated results. The particular user has 134 real trajectory records, and we generated the same number of synthetic trajectories for this user. Fig. 6(a) and (b) shows the trajectories by STULIG and TULVAE, respectively. The bottom part of Fig. 6 shows the time distribution of check-ins. Specifically, Fig. 6(c) shows

the true check-in time distribution, whereas Fig. 6(d) and (e) shows the time distribution of check-ins for the trajectories generated using STULIG and TULVAE, respectively.

a) Geographic distribution: We observe that most of the real trajectories of this user—red ones in Fig. 6(a) and (b)—are distributed in six states in USA. We then manipulate the latent variables to generate trajectories with the same length as the real trajectories obtained via STULIG and TULVAE, respectively. This is achieved by fixing the value of the generative factor u while randomly sampling another latent factor—$\mathbf{z}_2$ for STULIG and $\mathbf{z}$ for TULVAE. In general, there would be less variance between the generated samples and the real ones, if the latent factor captures more meaningful spatial distributions. Fig. 6(a) and (b) shows the generated spatial distribution (yellow) of each respective model. As for the quality of generation, we observe that STULIG generates the trajectories around the real trajectory distribution of this user, whereas the trajectories generated by TULVAE are rather sparse and further away from his/her real traces. It is also interesting to observe that the same latent factor $\mathbf{z}_2$ of STULIG may generate trajectories around a local area, e.g., those trajectories in Austin, Kansas City, and so on [see Fig. 6(a)].

b) Temporal distribution: In addition, Fig. 6(c) shows the ground truth of the check-in times of this user, which explicitly represents his/her temporal check-in preferences in POIs in terms of the hour of a day, and with respect to his/her entire history (over one year). Fig. 6(d) and (e) shows the time distribution of the trajectories generated by STULIG and TULVAE. Apparently, STULIG is also better in capturing and approximating the real check-in time than TULVAE, which generates a lot of unreasonable check-in time. As a specific example, looking at the time interval between 2:00 A.M. and 13:00 PM [see Fig. 6(e)], one can observe that much fewer check-ins are actually made by the user. These results further illustrate the benefits of the disentangled representation of identity and sequential check-in preference captured by our STULIG model—while variable u carries the identity information, latent factor $\mathbf{z}_2$ encodes the mobility patterns, e.g., spatial and temporal features of a particular user.

The qualitative results suggest that STULIG can be used to generate plausible trajectories of some individuals with consistent lifestyles and meaningful mobilities while protecting the privacy and preventing them from location inference attacks—i.e., camouflaging user's actual location with fakes (a popular obfuscation technique for protecting location privacy [30]).

C. Discussion

We have shown that STULIG outperforms baselines on human trace classification while being able to generate plausible synthesis mobility data. Closer investigation reveals that the proposed hierarchical latent factor model captures more flexible multimodel posterior of the latent variables.

As a simple CNN architecture for matrix-style trajectory convolutions used in our model, it becomes nontrivial to capture the long-term check-in dependences. For example, without the compensation of MoG approximation, STUL sometimes shows inferior performance compared with the RNN-based model TULVAE. Possible solutions include exploiting deeper CNN architectures such as ResNet [65], while broadening the receptive field of filters with dilated convolution [66] may be necessary for long-range traces.

We only considered MoG latent variables and note that high computational cost may arise in evaluating posterior expectations as the number of components K increases. In addition, the bottleneck of multimodal Gaussian approximation is evident for a larger value of K [see Fig. 4(a)]. To overcome this issue, more flexible and exact density estimation methods, such as normalizing flows [67], are desirable.

VI. Conclusion

We presented STULIG, a semisupervised generative model to mine human-mobility patterns and learn their interpretable latent factors. STULIG achieves a significant performance improvement for the TUL task in comparison to existing methods and is capable of learning disentangled representation of human traces. In addition, we demonstrated that STULIG can be used to generate plausible synthetic human traces, assisting machine learning tasks, while enabling a form of location privacy of the users.

As for our future work, we are interested in augmenting STULIG to discriminate different factors, for example, coupling spatial and temporal with other contexts related to POIs (e.g., restaurant, theater, and ATM) as well as the impact of mobility patterns (e.g., driving and walking). We believe that these may benefit many downstream applications, such as mobility knowledge transfer by reusing latent representations and mobility inference (e.g., predicting duration in next check-in) by combining learned latent factors. In addition, we will investigate the options for improving the accuracy via tighter ELBO bounds, as well as the broader challenge of how to provide privacy assurance to each mobile individual with deep generative models.

References

[1] C. Yang, L. Bai, C. Zhang, Q. Yuan, and J. Han, "Bridging collaborative filtering and semi-supervised learning: A neural approach for POI recommendation," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2017, pp. 1245–1254.

[2] Q. Liu, S. Wu, L. Wang, and T. Tan, "Predicting the next location: A recurrent model with spatial and temporal contexts," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, 2016, pp. 194–200.

[3] Q. Gao, F. Zhou, G. Trajcevski, K. Zhang, T. Zhong, and F. Zhang, "Predicting human mobility via variational attention," in *Proc. World Wide Web Conf.*, 2019, pp. 2750–2756.

[4] Q. Gao, G. Trajcevski, F. Zhou, K. Zhang, T. Zhong, and F. Zhang, "Trajectory-based social circle inference," in *Proc. 26th ACM SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst.*, Nov. 2018, pp. 369–378.

[5] Q. Gao, F. Zhou, K. Zhang, G. Trajcevski, X. Luo, and F. Zhang, "Identifying human mobility via trajectory embeddings," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 1689–1695.

[6] F. Zhou, Q. Gao, G. Trajcevski, K. Zhang, T. Zhong, and F. Zhang, "Trajectory-user linking via variational AutoEncoder," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 3212–3218.

[7] C. Cheng, H. Yang, M. R. Lyu, and I. King, "Where you like to go next: Successive point-of-interest recommendation," in *Proc. Int. Joint Conf. Artif. Intell. (IJCAI)*, 2013, pp. 2605–2611.

[8] D. Lian, C. Zhao, X. Xie, G. Sun, E. Chen, and Y. Rui, "GeoMF: Joint geographical modeling and matrix factorization for point-of-interest recommendation," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2014, pp. 831–840.

[9] H. Li, Y. Ge, D. Lian, and H. Liu, "Learning User's intrinsic and extrinsic interests for point-of-interest recommendation: A unified approach," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 2117–2123.

[10] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[11] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014, *arXiv:1412.3555*. [Online]. Available: http://arxiv.org/abs/1412.3555

[12] J. Feng *et al.*, "DeepMove: Predicting human mobility with attentional recurrent networks," in *Proc. World Wide Web Conf.*, 2018, pp. 1459–1468.

[13] J. Manotumruksa, C. Macdonald, and I. Ounis, "A contextual attention recurrent architecture for context-aware venue recommendation," in *Proc. 41st Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jun. 2018, pp. 555–564.

[14] F. Zhou, R. Yin, K. Zhang, G. Trajcevski, T. Zhong, and J. Wu, "Adversarial point-of-interest recommendation," in *Proc. World Wide Web Conf.*, 2019, pp. 3434–3462.

[15] C. Yan, H. Xie, J. Chen, Z. Zha, X. Hao, and Y. Zhang, "A fast uyghur text detector for complex background images," *IEEE Trans. Multimedia*, vol. 20, no. 12, pp. 3389–3398, Dec. 2018.

[16] C. Yan *et al.*, "Stat: Spatial-temporal attention mechanism for video captioning," *IEEE Trans. Multimedia*, vol. 22, no. 1, pp. 229–241, Feb. 2020.

[17] C. Yan, L. Li, C. Zhang, B. Liu, Y. Zhang, and Q. Dai, "Cross-modality bridging and knowledge transferring for image understanding," *IEEE Trans. Multimedia*, vol. 21, no. 10, pp. 2675–2685, Oct. 2019.

[18] H. Issa and M. L. Damiani, "Efficient access to temporally overlaying spatial and textual trajectories," in *Proc. 17th IEEE Int. Conf. Mobile Data Manage. (MDM)*, Jun. 2016, pp. 262–271.

[19] D. Chen, C. S. Ong, and L. Xie, "Learning points and routes to recommend trajectories," in *Proc. 25th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2016, pp. 2227–2232.

[20] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2014, pp. 1–5.

[21] B. Esmaeili, H. Wu, S. Jain, and A. Bozkurt, "Structured disentangled representations," in *Proc. 22nd Int. Conf. Artif. Intell. Statist.*, 2019, pp. 2525–2534.

[22] W. Xu, H. Sun, C. Deng, and Y. Tan, "Variational autoencoder for semi-supervised text classification," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, 2017, pp. 3358–3364.

[23] W. Xu and Y. Tan, "Semisupervised text classification by variational autoencoder," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 1, pp. 295–308, Jan. 2019.

[24] I. Higgins *et al.*, "Beta-vae: Learning basic visual concepts with a constrained variational framework," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2017, pp. 1–13.

[25] S. R. Bowman, L. Vilnis, O. Vinyals, A. Dai, R. Jozefowicz, and S. Bengio, "Generating sentences from a continuous space," in *Proc. 20th SIGNLL Conf. Comput. Natural Lang. Learn.*, 2016, pp. 1–12.

[26] A. A. Alemi, B. Poole, I. Fischer, J. V. Dillon, R. A. Saurous, and K. Murphy, "Fixing a broken elbo," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2018, pp. 159–168.

[27] N. Dilokthanakul *et al.*, "Deep unsupervised clustering with Gaussian mixture variational autoencoders," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2017, pp. 1–12.

[28] S. Narayanaswamy *et al.*, "Learning disentangled representations with semi-supervised deep generative models," in *Proc. Annu. Conf. Neural Inf. Process. Syst. (NIPS)*, 2017, pp. 5925–5935.

[29] V. Bindschaedler, R. Shokri, and C. A. Gunter, "Plausible deniability for privacy-preserving data synthesis," *Proc. VLDB Endowment*, vol. 10, no. 5, pp. 481–492, Jan. 2017.

[30] V. Bindschaedler and R. Shokri, "Synthesizing plausible privacy-preserving location traces," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2016, pp. 546–563.

[31] K. Ouyang, R. Shokri, D. S. Rosenblum, and W. Yang, "A non-parametric generative model for human trajectories," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 3812–3817.

[32] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," in *Proc. Annu. Conf. Neural Inf. Process. Syst. (NIPS)*, 2017, pp. 5767–5777.

[33] M. Caccia, L. Caccia, W. Fedus, H. Larochelle, J. Pineau, and L. Charlin, "Language GANs falling short," 2018, *arXiv:1811.02549*. [Online]. Available: http://arxiv.org/abs/1811.02549

[34] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013, *arXiv:1301.3781*. [Online]. Available: http://arxiv.org/abs/1301.3781

[35] F. Xu, Z. Tu, Y. Li, P. Zhang, X. Fu, and D. Jin, "Trajectory recovery from ash: User privacy is NOT preserved in aggregated mobility data," in *Proc. 26th Int. Conf. World Wide Web*, Apr. 2017, pp. 1241–1250.

[36] D. P. Kingma, S. Mohamed, D. J. Rezende, and M. Welling, "Semi-supervised learning with deep generative models," in *Proc. Annu. Conf. Neural Inf. Process. Syst. (NIPS)*, 2014, pp. 3581–3589.

[37] M. D. Hoffman and M. J. Johnson, "Elbo surgery: Yet another way to carve up the evidence lower bound," in *Proc. Workshop Adv. Approx. Bayesian Inference*, 2016, p. 2.

[38] J. M. Tomczak and M. Welling, "Vae with a vampprior," in *Proc. Int. Conf. Artif. Intell. Statist. (AISTATS)*, 2018, pp. 1214–1223.

[39] F. Locatello *et al.*, "Challenging common assumptions in the unsupervised learning of disentangled representations," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2019, pp. 4114–4124.

[40] R. Shu, Y. Chen, A. Kumar, S. Ermon, and B. Poole, "Weakly supervised disentanglement with guarantees," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2020, pp. 1–36.

[41] F. Locatello, M. Tschannen, S. Bauer, G. Rätsch, B. Schölkopf, and O. Bachem, "Disentangling factors of variation using few labels," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2019, pp. 1–24.

[42] H. Araki and F. Hansen, "Jensen's operator inequality for functions of several variables," *Proc. Amer. Math. Soc.*, vol. 128, no. 7, pp. 2075–2084, 2000.

[43] Q. Yuan, G. Cong, and A. Sun, "Graph-based Point-of-interest recommendation with geographical and temporal influences," in *Proc. 23rd ACM Int. Conf. Conf. Inf. Knowl. Manage.*, 2014, pp. 659–668.

[44] Y. Liu, T.-A. N. Pham, G. Cong, and Q. Yuan, "An experimental evaluation of point-of-interest recommendation in location-based social networks," *Proc. VLDB Endowment*, vol. 10, no. 10, pp. 1010–1021, Jun. 2017.

[45] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. KDD*, vol. 96, 1996, pp. 226–231.

[46] J. Manotumruksa, C. Macdonald, and I. Ounis, "A deep recurrent collaborative filtering framework for venue recommendation," in *Proc. ACM Conf. Inf. Knowl. Manage.*, Nov. 2017, pp. 1429–1438.

[47] C. Yang, M. Sun, W. X. Zhao, Z. Liu, and E. Y. Chang, "A neural network approach to jointly modeling social networks and mobile trajectories," *ACM Trans. Inf. Syst.*, vol. 35, no. 4, p. 36, Aug. 2017.

[48] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, "Convolutional sequence to sequence learning," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2017, pp. 1243–1252.

[49] S. Bai, J. Zico Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," 2018, *arXiv:1803.01271*. [Online]. Available: http://arxiv.org/abs/1803.01271

[50] J. Miller and M. Hardt, "Stable recurrent models," 2018, *arXiv:1805.10369*. [Online]. Available: http://arxiv.org/abs/1805.10369

[51] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Int. Conf. Learn. Represent. (ICLR)*, 2015, pp. 1–15.

[52] R. Prabhavalkar, T. N. Sainath, B. Li, K. Rao, and N. Jaitly, "An analysis of–attention–in sequence-to-sequence models," in *Proc. Interspeech*, Aug. 2017, pp. 1672–1682.

[53] A. Razavi, A. V. D. Oord, B. Poole, and O. Vinyals, "Preventing posterior collapse with delta-vaes," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2019, pp. 1–8.

[54] X. Chen *et al.*, "Variational lossy autoencoder," in *Proc. Int. Conf. Learn. Represent.(ICLR)*, 2017, pp. 1–7.

[55] T. Q. Chen, X. Li, and R. G. Neural, "Isolating sources of disentanglement in variational autoencoders," in *Proc. Conf. Neural Inf. Process. Syst. (NeurIPS)*, 2018, pp. 2610–2620.

[56] A. Vaswani *et al.*, "Attention is all you need," in *Annu. Conf. Neural Inf. Process. Syst. (NIPS)*, 2017, pp. 5998–6008.

[57] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*. [Online]. Available: http://arxiv.org/abs/1810.04805

[58] L. Song, D. Kotz, R. Jain, and X. He, "Evaluating location predictors with extensive Wi-Fi mobility data," in *Proc. IEEE INFOCOM*, 2004, pp. 1414–1424.

[59] M. Yin, M. Sheehan, S. Feygin, J.-F. Paiement, and A. Pozdnoukhov, "A generative model of urban activities from cellular data," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 6, pp. 1682–1696, Jun. 2018.

[60] X. Song, H. Kanasugi, and R. Shibasaki, "Deeptransport: Prediction and simulation of human mobility and transportation mode at a city-wide level," in *Proc. Int. Joint Conf. Artif. Intell. (IJCAI)*, 2016, pp. 2618–2624.
[61] G. Acs, L. Melis, C. Castelluccia, and E. De Cristofaro, "Differentially private mixture of generative neural networks," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 6, pp. 1109–1121, Jun. 2019.
[62] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," *J. Privacy Confidentiality*, vol. 7, no. 3, pp. 17–51, 2006.
[63] M. Abadi *et al.*, "Deep learning with differential privacy," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2016, pp. 308–318.
[64] T. Zhang, M. Huang, and L. Zhao, "Learning structured representation for text classification via reinforcement learning," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, 2018, pp. 6053–6060.
[65] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[66] A. Gupta and A. M. Rush, "Dilated convolutions for modeling long-distance genomic dependencies," 2017, *arXiv:1710.01278*. [Online]. Available: http://arxiv.org/abs/1710.01278
[67] D. J. Rezende and S. Mohamed, "Variational inference with normalizing flows," in *Proc. ICML*, 2015, pp. 1530–1538.

Fan Zhou (Member, IEEE) received the B.S. degree in computer science from Sichuan University, Chengdu, China, in 2003, and the M.S. and Ph.D. degrees from the University of Electronic Science and Technology of China, Chengdu, in 2006 and 2012, respectively.

He is currently an Associate Professor with the School of Information and Software Engineering, University of Electronic Science and Technology of China. His research interests include machine learning, neural networks, spatiotemporal data management, graph learning, recommender systems, and social network data mining and knowledge discovery.

Xin Liu received the M.S. degree from the University of Electronic Science and Technology of China, Chengdu, China, in 2020.

She is currently a Researcher with the 360 AI Security Research Labs. Her research interests include spatiotemporal data mining, deep generative models, and AI security.

Kunpeng Zhang received the Ph.D. degree in computer science from Northwestern University, Evanston, IL, USA.

He is currently a Researcher in the area of large-scale data analysis, with particular focuses on social data mining, image understanding via machine learning, social network analysis, and natural language processing. He is also an Assistant Professor with the Department of Information Systems, Smith School of Business, University of Maryland, College Park, MD, USA. He has published papers in the areas of social media, artificial intelligence, network analysis, and information systems on top conferences and journals.

Dr. Zhang serves on the program committee for many conferences and an associate editor for many journals.

Goce Trajcevski (Member, IEEE) received the B.Sc. degree from the University of Sts. Kiril i Metodij, Skopje, Macedonia, in 1989, and the M.S. and Ph.D. degrees from the University of Illinois at Chicago, Chicago, IL, USA, in 1995 and 2002, respectively.

He is currently an Associate Professor with the Department of Electrical and Computer Engineering, Iowa State University, Ames, IA, USA. In addition to a book chapter and three encyclopedia chapters, he has coauthored over 140 publications in refereed conferences and journals. His research has been funded by the NSF, ONR, BEA, and Northrop Grumman Corporation. His main research interests are in the areas of spatiotemporal data management, uncertainty and reactive behavior management in different application settings, and incorporating multiple contexts.

Dr. Trajcevski was the General Co-Chair of the IEEE ICDE 2014 and ACM SIGSPATIAL 2019 and the PC Co-Chair of the ADBIS 2018 and ACM SIGSPATIAL 2016 and 2017. He has served in various roles in organizing committees in numerous conferences and workshops. He is also an Associate Editor of the ACM TSAS and *Geoinformatica*.