Predicting Human Mobility via Variational Attention

Qiang Gao
University of Electronic Science and Technology of China, Chengdu, China
qianggao@std.uestc.edu.cn

Fan Zhou*
University of Electronic Science and Technology of China, Chengdu, China
fan.zhou@uestc.edu.cn

Goce Trajcevski
Iowa State University, Ames
gocet25@iastate.edu

Kunpeng Zhang
University of Maryland, College park
kpzhang@umd.edu

Ting Zhong
University of Electronic Science and Technology of China, Chengdu, China
zhongting@uestc.edu.cn

Fengli Zhang
University of Electronic Science and Technology of China, Chengdu, China
fzhang@uestc.edu.cn

ABSTRACT

An important task in Location based Social Network applications is to predict mobility – specifically, user's next point-of-interest (POI) – challenging due to the implicit feedback of footprints, sparsity of generated check-ins, and the joint impact of historical periodicity and recent check-ins. Motivated by recent success of deep variational inference, we propose VANext (Variational Attention based Next) POI prediction: a latent variable model for inferring user's next footprint, with historical mobility attention. The variational encoding captures latent features of recent mobility, followed by searching the similar historical trajectories for periodical patterns. A trajectory convolutional network is then used to learn historical mobility, significantly improving the efficiency over often used recurrent networks. A novel variational attention mechanism is proposed to exploit the periodicity of historical mobility patterns, combined with recent check-in preference to predict next POIs. We also implement a semi-supervised variant – VANext-S, which relies on variational encoding for pre-training all current trajectories in an unsupervised manner, and uses the latent variables to initialize the current trajectory learning. Experiments conducted on real-world datasets demonstrate that VANext and VANext-S outperform the state-of-the-art human mobility prediction models.

CCS CONCEPTS

• **Information systems → Location based services**; **Geographic information systems**.

KEYWORDS

human mobility, variational attention, convolutional neural network

ACM Reference Format:
Qiang Gao, Fan Zhou, Goce Trajcevski, Kunpeng Zhang, Ting Zhong, and Fengli Zhang. 2019. Predicting Human Mobility via Variational Attention. In *Proceedings of the 2019 World Wide Web Conference (WWW '19), May 13–17, 2019, San Francisco, CA, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3308558.3313610

*Corresponding author: Fan Zhou (fan.zhou@uestc.edu.cn)

ACM ISBN 978-1-4503-6674-8/19/05.
https://doi.org/10.1145/3308558.3313610

1 INTRODUCTION

Many users of Location-Based Social Networks (LBSNs) such as Foursquare and Twitter leave and/or share their footprints with friends. This has enabled new research in learning human mobility and using it in various settings – e.g., social relation inference [1], friendship prediction [24], trajectory-user linking [11, 34], etc. A critical task in understanding human mobility patterns is the next Point-of-Interest (POI) prediction – extensively studied in recent years [6, 7, 17, 27] and becomes of interest in many application domains – e.g., location sharing or recommendation for users of enterprises in social media [16]; route planing for taxi driver to maximize potential passenger pickups [27]; forecasting criminals/terrorists next location for public safety [11]; etc.

Many methods exploiting human recent mobility and their historical visits have been proposed for next POI prediction, including Matrix Factorization (MF) [16], Markov Chain (MC) [9], Factorizing Personalized Markov Chains (FPMC) [8] and Tensor Factorization (TF) [32], all incorporating human visit preferences and investigating sequential patterns. Deep learning techniques have also been applied in learning users' mobility – ST-RNN [17], POI2Vec [7], and DeepMove [6] – employing recurrent neural networks (RNN – e.g., LSTM [21] and GRU [4]) to capture the sequential patterns of users' mobility preferences. Given historical trajectories, these works train a RNN module to predict the next POI, together with corresponding contextual (e.g., spatial and temporal) features.

However, there are certain challenges for the existing works: (1) *Sparsity of recent mobility* – within a given period, a user may generate only a few check-ins (different from GPS data that is passively/continuously sampled), making the prediction difficult. (2) *Density of historical mobility* – LBSNs often collect user's check-ins for months and even years. In comparison with recent mobility, it is more difficult to represent and exploit the whole historical trajectory effectively and efficiently in a training model. (3) *Complex data* – user's historical moving patterns affect the subsequent footprints [3, 17], whereas individual check-in behaviors are complex and personalized.

To address the above challenges, we propose a novel model – Variational Attention based Next (VANext) POI prediction, leveraging the recent advances in variational Bayesian techniques [14] to encode recent mobility into a latent variable and use it to query one's historical mobility. We also introduce a historical trajectory-based learning module implemented by a convolutional neural network (CNN) kernel – capturing individual's long-term moving patterns in a more efficient way, compared to widely used RNNs. Specifically, a variational attention on historical trajectory is combined with the latent representation learned from a recent trajectory to capture the similar check-in sequence (not necessarily continuous) which reflects user long-term visit intention. Finally, we concatenate representations of recent mobility and historical trajectory features to predict user's next POI. We also propose VANext-S – a semi-supervised variant, to better capture the latent variables by pre-training all current trajectories in an unsupervised manner within the framework of Variational Auto-Encoder (VAE) [14]. It samples from latent variables to generate and reconstruct each current trajectory, and utilizes the encoder to initialize VANext. In summary, our main contributions are:

• We frame the mobility prediction task using a generative probabilistic model together with neural networks, and propose VANext that can simultaneously learn implicit relationships between users and POIs and capture sequential patterns of user check-in behavior for next POI prediction.

• We introduce a novel variational attention mechanism to learn the attention on the historical trajectories, from which the latent representation of a recent trajectory is leveraged to match user's most similar moving patterns in the past.

• We conduct experiments on real-world datasets consisting of users' check-in behavior in four cities, demonstrating that VANext outperforms existing models in terms of both prediction accuracy and training efficiency.

To our knowledge, VANext is the first attempt to incorporate CNN to capture long term and structural dependency among user check-ins, achieving comparable learning ability with the state-of-the-art RNN based methods, while significantly improving the learning efficiency. For the rest of this paper, we first review the related work, then formalize the problem and present our main results, followed by experimental evaluations and concluding remarks.

2 RELATED WORK

POI recommendation and prediction are two different but related and extensively studied topics in LBSN: the former usually learns users' preferences over POIs while the latter is more interested in mobility pattern recognition. Collaborative filtering (CF)-based models, such as Matrix Factorization (MF) [2, 15, 16] and tensor factorization (TF) [32], are widely used in POI recommendation for learning users latent preferences. Geographical information [15, 16, 31], as well as time-aware influence and content-aware influence [10, 26, 30] are most effective features incorporated in enhancing recommendation performance [18]. However, these model-based methods are not suitable to predict user's next behavior/visits, due to the lack of taking sequential and periodical patterns into account. Markov Chains (MC)-based methods are widely used for modeling sequential influence. For example, FPMC [20] and FPMC-LR [3] aim to predict the user's next visit based on factorization of the probability transition matrix. In addition, embedding has also been applied for addressing next POI prediction [7, 8] – e.g., a Personalized Ranking Metric Embedding (PRME) [8], attempting to capture user preference with the latent space. Inspired by the word2vec [19] technique in NLP, POI2Vec [7] was proposed for next POI prediction with geographical influence included. However, both MC and embedding based methods are not capable of learning long term dependency of POIs and the periodicity of individual's historical moving.

Recently, deep learning techniques – especially recurrent neural networks (RNNs) such as LSTM [21] and GRU [4] – have been widely used to capture the long term sequential influence and mobility patterns. Spatial Temporal Recurrent Neural Networks (ST-RNN) [17] extend the RNN model by incorporating temporal and spatial context in each time unit for predicting next POIs. A unified RNN-based framework jointly learning the embeddings of multiple factors (e.g., user identity, location and time, etc.), was presented in [29]. However, these methods do not explicitly model user's historical visit patterns and personal preferences, but greatly focusing on current locations and short term dependencies among POIs. More recently, [6] propose an attentional recurrent network for mobility prediction from lengthy and sparse trajectories, where two RNN models – learning the current and the historical trajectory, respectively – together exploit user's mobility and location preference with attention on multi-level periodicity of historical trajectories. However, it is complicated to train due to the relative high density of historical trajectories.

CNNs have been widely applied in image classification and video tagging, but are less common for sequence embedding. A CNN-based architecture successfully improving the machine translation performance was recently proposed in [12]. Compared to RNN-based model, CNNs can easily capture hierarchical representations of the underlying context, and, most importantly, are faster to be trained due to parallelization with GPU. Variational Auto-Encoder (VAE) [14] has been applied in many NLP tasks, such as text classification [25] and dialogue generation [22], and has shown promising performance in exploiting underlying patterns of trajectories [34]. In addition, these models neither consider the contextual information associated with POIs, nor capture periodical mobility patterns, thereby cannot be directly employed for POI prediction.

Our proposed model VANext differs from these VAE-based methods in terms of its periodical pattern learning and the variational attention mechanism.

3 PRELIMINARIES

We now proceed with introducing the basic terminology and formalizing the problem, followed by the details of the proposed model – including causal POI embedding, variational current trajectory learning, historical trajectory attention module and the classifier.

A POI is defined as a location of relevance obtained, for example, as a GPS point; a centroid of a region; an address of an object, etc. – which can be uniquely identified in a suitable coordinate system. It is represented as a triplet $l = <id,lo,la>$, where id, lo and la denote the POI id, longitude and latitude. T^u denotes a trajectory generated by a user u, represented as a sequence $\mathrm{T}^u = (l^u_{t_1}, l^u_{t_2}, \cdots, l^u_{t_n})$, where $l^u_{t_i}$ is i-th POI visited by user u at time t_i. When there is no ambiguity, we will omit the superscript ^{u}. A trajectory T can be segmented as $\mathrm{T} = (\mathrm{T}^1, \mathrm{T}^2, \cdots, \mathrm{T}^m)$, meaning that there are m

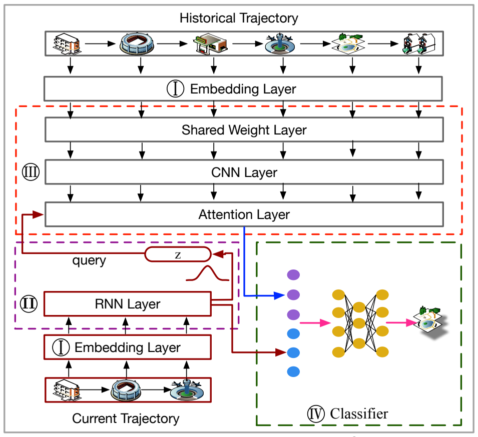


Figure 1: Main components of VANext

sub-trajectories within the time interval $[t_1, t_n]$, ordered along the temporal dimension. The trajectories are separated from each other by application dependent thresholds δ^j_{T} $(1 \leq j \leq m)$, which could indicate, for example: – the time span of each T^j is no more than δ^j_{T}; – the last time-stamp of T^j is at least δ^j_{T} time units smaller than the time-stamp of the first location in T^{j+1}; etc.

Let $\mathrm{T}_h = (\mathrm{T}^1, \cdots, \mathrm{T}^m)$ denote the entire historical trajectory of the user u (note that its time span may vary, from weeks, to months or even years for different users) and let $\mathrm{T}_c = \mathrm{T}^{m+1} = (l_{t_{n+1}}, l_{t_{n+2}}, \cdots, l_{t_{n+k-1}})$ denote the current/most recent sub-trajectory.

Problem formulation: We consider next POI prediction as a multi-classification problem, which is formulated as: given a user u with the historical trajectory T_h, and the recently visited sequence of POIs $\mathrm{T}_c = (l_{t_{n+1}}, l_{t_{n+2}}, \cdots, l_{t_{n+k-1}})$, train a model M to predict the next POI $\gamma = l_{t_{n+k}}$ for user u – i.e., $M(\mathrm{T}_h, \mathrm{T}_c) \mapsto \gamma$.

VAE [14] is a generative model that learns the probability $p(X|\mathbf{z})$ of the data X (e.g., a trajectory) given a latent variable $\mathbf{z}$, and a recognition model $q(\mathbf{z}|X)$ simultaneously. The lower bound of the true marginal log likelihood is:

$$\log p_\theta(X) \geq \mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z}|X)}[\log p_\theta(X|\mathbf{z})] - \mathcal{KL}(q_\phi(\mathbf{z}|X)||p(\mathbf{z}))$$

where both $q_\phi(\cdot)$ and $p_\theta(\cdot)$ are learned with deep neural networks, parameterized by ϕ and θ, respectively. $\mathcal{KL}$ is the KL-divergence.

4 METHODOLOGY

4.1 Basic Framework

Figure 1 shows the four main modules of our proposed model VANext:

(I) *Causal POI embedding*: To better present the semantic relationships among different POIs, we first embed POIs, from both current trajectory and historical trajectory, into a low dimensional representation using a causal embedding method.

(II) *Current Trajectory Learning Module*: We apply a recurrent neural network (RNN) – in this paper, we choose the GRU cell – to encode the current trajectory T_c, and use the last hidden state to represent it , and further learns the distribution of a latent variable $\mathbf{z}$ using variational encoding as $q_\phi(\mathbf{z}|\mathrm{T}_c)$.

(III) *Historic Trajectory Learning Module*: VANext utilizes a shared weight matrix and a convolutional layer to encode the historical trajectory T_h. The attention mechanism with latent variable $\mathbf{z}$ is considered as a query to exploit users' (historical) mobility patterns. It seeks the most similar mobility to the current trajectory T_c by searching the historical trajectory T_h, and aggregates its hidden state from T_h to form the representation. The formalized representation of aggregated result is $\tilde{\mathrm{T}}_h$. In other words, attention module will force the historical trajectory representation $\tilde{\mathrm{T}}_h$ to automatically match T_c.

(IV) *Classifier*: This is the final component which unifies the last hidden state from current trajectory and the aggregated attention from historical trajectory into a feature representation, and predicts the next POI γ.

4.2 Trajectory Generative Model

Before presenting the details of each component, we first adapt our VANext into a trajectory generative model. The primary task of VANext is to predict γ from T_c. Hence, the likelihood can be specified as:

$$\begin{aligned}\log p_\theta(\gamma|\mathrm{T}_c) &= \log \int_{\mathbf{z}} p_\theta(\gamma|\mathrm{T}_c, \mathbf{z}) p(\mathbf{z}) d\mathbf{z} \\ &= \log \int_{\mathbf{z}} q_\phi(\mathbf{z}|\mathrm{T}_c, \gamma) \frac{p_\theta(\gamma|\mathrm{T}_c, \mathbf{z})}{q_\phi(\mathbf{z}|\mathrm{T}_c, \gamma)} p(\mathbf{z}) d\mathbf{z},\end{aligned}$$

With Jensen's inequality, we have

$$\begin{aligned}&\log p_\theta(\gamma|\mathrm{T}_c) \geq \mathbb{E}_{q_\phi(\mathbf{z})}[\log p_\theta(\gamma|\mathrm{T}_c, \mathbf{z}) + \log \frac{p(\mathbf{z})}{q_\phi(\mathbf{z}|\gamma, \mathrm{T}_c)}] \\ &= \mathbb{E}_{q_\phi(\mathbf{z})} p_\theta(\gamma|\mathrm{T}_c, \mathbf{z}) - \mathcal{KL}(q_\phi(\mathbf{z}|\gamma, \mathrm{T}_c)||p(\mathbf{z})). \qquad (1)\end{aligned}$$

Since γ is a function of input T_c, we can absorb γ into T_c following [33] and therefore obtain the following variant of Eq.(1):

$$\log p_\theta(\gamma|\mathrm{T}_c) \geq \mathbb{E}_{q_\phi(\mathbf{z})} p_\theta(\gamma|\mathrm{T}_c, \mathbf{z}) - \mathcal{KL}(q_\phi(\mathbf{z}|\mathrm{T}_c)||p(\mathbf{z})). \qquad (2)$$

In VaNext, we propose a variational attention mechanism to obtain an attentional representation $\tilde{\mathrm{T}}_h$. In detail, we use the latent variable $\mathbf{z}$ as a query vector to find the most similar mobility $\tilde{\mathrm{T}}_h$ from T_h. Since $\mathbf{z}$ determines $\tilde{\mathrm{T}}_h$, hence we obtain our training objective that is to maximize the lower bound of likelihood $L(\theta, \phi)$:

$$L(\theta, \phi) = \mathbb{E}_{q_\phi(\mathbf{z})}[\log p_\theta(\gamma|\mathrm{T}_c, \tilde{\mathrm{T}}_h)] - \lambda \mathcal{KL}[q_\phi(\mathbf{z}|\mathrm{T}_c)||p(\mathbf{z})], \qquad (3)$$

where λ is harmonious factor.

4.3 Causal POI Embeddings

Inspired by word2vec [19], previous works [6, 11, 34] embed POIs with the context information using CBOW or Skip-Gram models. However, for the next POI prediction problem, we are more interested in embedding the proceeding part of a trajectory instead of the surrounding context – the rationale behind is that the current POI is determined by its previous footprint rather than the ones behind it, which is very similar to the idea of high-order Markov Process. That is, there can be no leakage from the future into the past when embedding the POIs. This can be accomplished with a *causal* embedding where the probability of check-in l_τ is maximized only with the given elements from earlier footprints $l_{\tau-\omega} : l_{\tau-1}$, where ω is the fixed size of the sliding window.

Specifically, the check-in representations $\mathbf{v} \in \mathbb{R}^{|\mathcal{L}| \times d}$ – $|\mathcal{L}|$ is the number of check-ins in the dataset and d is the dimensionality – are obtained by predicting each current location l_τ in both T_h

and T_c given the proceeding context locations $l_{\tau-\omega} : l_{\tau-1}$. The probability $p(l_\tau|C(l_\tau))$ is defined by the *softmax* function as:

$$p(l_\tau|C(l_\tau)) = \prod_{l' \in C(l_\tau)} \frac{\exp(\mathbf{v}(l_\tau) \cdot \mathbf{v}(l'))}{\sum_{l'' \in \mathcal{L}} \exp(\mathbf{v}(l'') \cdot \mathbf{v}(l'))}$$

where we adopt the *Negative Sampling* technique to avoid the enumeration of all check-ins $l'' \in \mathcal{L}$ for the efficiency purpose.

4.4 Current Trajectory Learning Module

This module processes the current trajectory T_c with a recurrent network and meanwhile learns a distribution of a latent variable $\mathbf{z}$.

4.4.1 – Current trajectory with GRU:. To capture users' current sequential check-in pattern, we use a vanilla GRU to exploit the mobility of T_c. Note that user's recent trajectories are usually shorter sequence compared to the extremely long historical trajectory, i.e., $|T_c| \ll |T_h|$. Therefore, we use a recurrent network instead of convolutional operation that would be employed in the historical module introduced in next section. Moreover, to make a well-targeted comparison with previous work [6], we also select GRU kernel as the neural network unit:

$$h_\tau = (1 - g_\tau)h_{\tau-1} + g_\tau \tilde{h}_\tau \tag{4}$$

where g_τ is the update gate in time τ which decides how much the unit updates its activation by

$$g_\tau = \sigma(W_g \mathbf{v}(l_\tau) + U_g h_{\tau-1}), \tag{5}$$

where σ denotes the sigmoid activation function. The candidate state $\tilde{h}_t$ is computed similarly to traditional RNN unit

$$\tilde{h}_\tau = \tanh(W_{\tilde{h}} \mathbf{v}(l_\tau) + U_{\tilde{h}}(s_\tau \odot h_{\tau-1})) \tag{6}$$

where $\odot$ is element-pair product, s_τ is a set of reset gates and is computed similarly to update the gate

$$s_\tau = \sigma(W_s \mathbf{v}(l_\tau) + U_s h_{\tau-1}) \tag{7}$$

where W_* and U_* are both parameterized matrices.

4.4.2 – Latent variable learning: Variational auto-encoding provides an efficient way to approximate the posterior distribution of latent variables. In our trajectory generative model, we sample the empirical posterior $q_\phi(\mathbf{z}|T_c)$ of latent variables $\mathbf{z}$ from the last hidden state of current trajectories. The objective is to minimize the KL-divergence between posterior and prior as shown in Eq.(3). Typically, we assume the prior distribution of latent variable $\mathbf{z}$ following a standard Gaussian: $p(\mathbf{z}) \sim N(0, I)$.

4.5 Historical Trajectory Learning Module

This module involves the convolutions on historical trajectories and the variational attention layer.

4.5.1 – Historical trajectory convolution: Before applying the convolution operation over the historical trajectories, we first use a shared weight layer (e.g, a MLP) to incorporate each POI representation vector into a hidden state:

$$c_{t_i} = \mathrm{ReLU}(W_h(\mathbf{v}(l_{t_i})) + b_h), i \in [1, 2, \cdots, n] \tag{8}$$

where ReLU is the activation function. Then, we obtain all hidden states of embedded POIs as: $c_t = (c_{t_1}, c_{t_2}, \cdots, c_{t_n})$.

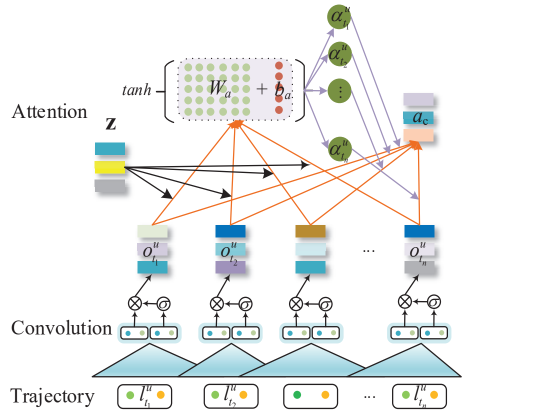


Figure 2: Attention mechanism in Historical Trajectory Module.

Subsequently, a 1×D convolution with non-linearity neural network is involved to exploit the hidden state patterns. The convolution kernel is parameterized with matrix $W_c \in \mathbb{R}^{2d \times nd}$ and bias $b_c \in \mathbb{R}^{2d}$, where n is the number of the hidden states. Therefore, each input hidden state $c_{t_i} \in \mathbb{R}^d$ will generate an output $O_i \in \mathbb{R}^{2d}$ by a CNN. By splitting each element O_i into O_{i_1} and O_{i_2} (both $\in \mathbb{R}^d$), the final output o_{t_i} will be obtained by activating O_{i_1} and O_{i_2} to a non-linearity with gated linear units, as follows:

$$o_{t_i} = O_{i_1} \odot \sigma(O_{i_2}) \tag{9}$$

Thus, the convolutional state $o_t = (o_{t_1}, o_{t_2}, \cdots, o_{t_n})$ is obtained as the input of the next attention layer.

4.5.2 – Variational attention. As the Figure 2 shows, we present a variational attention method to learn the weight of convolutional states in historical trajectories. We first use a MLP to obtain the query vector $\mathbf{q}$ from $\mathbf{z}$:

$$\mathbf{q} = \mathrm{ReLU}(\mathbf{z}W_z + b_z) \tag{10}$$

which is then used as a query to calculate the similarity score between $\mathbf{q}$ and o_t as

$$u_{t_i} = \tanh(\mathbf{q}(W_u o_{t_i} + b_u)) \tag{11}$$

We next get a normalized importance weight α_i with a softmax function.

$$\alpha_i = \frac{\exp(u_{t_i}^\top u_w)}{\sum_{i=1}^n \exp(u_{t_i}^\top u_w)} \tag{12}$$

where u_w is a vector variable learned during the training process.

Now we obtain the representation of historical trajectory as a weighted sum of the POIs.

$$\tilde{a} = \sum \alpha_i o_{t_i} \tag{13}$$

Note that above attention method is different from traditional attention methods in that the learned mobility pattern representation not only depends on the check-ins of historical trajectory but also considers the variational query variable $\mathbf{z}$ from current trajectory. This variational attention technique allows us to sample from a continuous distribution $q_\phi(\mathbf{z}|T_c)$ rather than from discrete/sparse current trajectory T_c, and thus is able to better match the historical trajectory. As we will show in experiments, this novel variational attention method introduces significant performance improvement compared to the traditional vanilla attention used in previous work [6].

Next POI prediction To predict the next POI of T_c, we first concatenate the last hidden state of current trajectory and the attentional historical trajectory to obtain Φ as follows:

$$\Phi = W_e * \begin{bmatrix} h_{n+k-1} \\ \tilde{a} \end{bmatrix} + b_e \tag{14}$$

Then, we feed Φ into a softmax function to calculate the probability of each POI $p(l)$ in the dataset. During the training of VANext, we are interested in minimizing the empirical risk of predicting the next POI γ.

Semi-supervised VANext: VANext-S We propose a variant of VANext, called VANext-S, which utilizes VAE to pretrain all current trajectories in an unsupervised manner, and uses the encoder and latent variables to initiate the current trajectory learning.

5 EXPERIMENTS

In this section, we conduct experiments on two real-world datasets to compare the performance of the proposed models against several baselines.

Datasets. To ease reproducing the results, we conduct all experiments on the publicly available datasets including Foursquare and Gowalla [7, 28, 30]. We select two cities (New York, Singapore) from Foursquare, and two cities (Houston, California) from Gowalla. For each dataset, we remove the POIs which have been visited by fewer than 5 users. For each user, we concatenate all check-in locations to form a single trajectory – subsequently, we divide it into sub-trajectories with the time interval of 6 hours each, as it was done in previous related works [7, 11, 17]. Further, we filter out the users who have fewer than 5 sub-trajectories. For all datasets, we choose each user's first 80% sub-trajectories as the training set, and the remaining 20% as testing data. Table 1 summarizes the statistics of the datasets after pre-processing.

Table 1: Descriptive statistics of Datasets.

City	Users	POIs	Check-ins	Sub-trajectories
New York	1,083	9,815	120,007	36,182
Singapore	2,321	5,596	194,108	34,713
Houston	4,627	15,234	362,783	18,501
California	3,987	21,354	239,493	66,612

Table 2: Experimental settings of VANext and VANext-S.

Parameter	Values	Model Setting	Chooses
window size	5	embedding size	256
HS/NS	NS	hidden size	300
learning rate	1e-3	attention size	300
dropout rate	0.5	$\|\mathbf{z}\|$ size	128
λ	0.01	batch size	16

Settings. Table 2 characterizes the specific settings and hyper-parameters when implementing our VANext and VANext-S models. In particular, we use negative sampling (NS) instead of Hierarchical Softmax (HS) for our causal POI embedding. We implement two models, as well as all deep neural network-based models, on the Tensorflow platform and speed up using a GTX1080ti GPU.

Metrics. To make fair comparisons, we use the standard evaluation performance metrics, such as **Top@k**, Area under the ROC curve (**AUC**) and Mean Average Precision (**MAP**) used in [6, 17].

Baselines. We compare our models with several classical methods and the most recently proposed methods regarding next POI prediction, including:

- **Markov chain** [9] – Markov chain is one of the classical methods which predict the POIs using the estimated transition probability.
- **PRME** [8] – Personalized ranking metric embedding is a pair-wise metric embedding method embedding every POI into a latent Euclidean space and computing the location transition in a Markov chain model.
- **ST-RNN** [17] – It is a deep learning model that incorporates spatial and temporal context and predicts user's next visit within the framework of RNN.
- **Bi-LSTM** [26] – It is a method extending ST-RNN with a more sophisticated Bidirectional-LSTM model that has been used in [26] for learning individual trajectory patterns and predicting POIs.
- **POI2vec**[7] – It is an embedding method which incorporates the geographical influence for next POI prediction based on word2vec technique.
- **DeepMove**[6] – The first historical attention method for learning human mobility, consists of a sequential encoding module with RNN for learning motion patterns from both recent and historical trajectories.

Experimental Results

***Overall performance comparison*.** Table 3 shows the **Top@k** performance comparison on four citywide datasets. As we can see, overall, the proposed VANext and VANext-S yield the best performance among all methods in terms of prediction precision. Specifically, VANext and VANext-S outperform the second-best method DeepMove by 31.52%, 36.81% and 32.68%, in terms of *Top@1*, *Top@5* and *Top@10* respectively on Singapore dataset.

As for the performance of baselines, deep neural network-based methods (i.e., ST-RNN, Bi-LSTM, POI2vec and DeepMove) exhibit higher prediction accuracy than traditional feature-based embedding method and Markov-based methods. Furthermore, DeepMove generally gives better performance than other baselines due to its ability of exploiting historical trajectories while others can only learn the sequential patterns in (short) current trajectories during training. Although all of these methods have employed recurrent networks in order to capture long term POI dependency, even LSTM and GRU cannot work well on very long historical trajectories due to the gradient vanishing problem inherent in recurrent networks. This result is in accordance with the limitations of applying RNN in natural language processing [5, 13]. Therefore, the main advantage of DeepMove lies in its attention on historical trajectories, which also confirms the superiority of attention mechanism in processing sequential data, e.g., attention mechanism along achieves comparable results on machine translation with CNN or RNN based seq2seq methods [23].

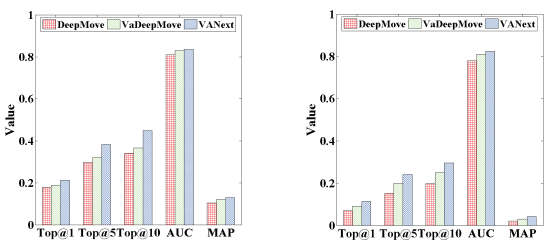
On the other hand, our methods outperform DeepMove mainly because the introduced trajectory generative model which not only captures short-term human mobility patterns more effectively but also yields better historical trajectory attention with the novel variational attention method.

Additionally, VANext-S outperforms VANext because of its pre-training on the current trajectories in an unsupervised manner. Although autoencoder based pretraining is a well-known training

Table 3: Comparison of overall prediction accuracy on four datasets. The best method is shown in bold, and the second best is shown as underlined

Method	New York			Singapore			Houston			California		
	Top@1	Top@5	Top@10	Top@1	Top@5	Top@10	Top@1	Top@5	Top@10	Top@1	Top@5	Top@10
Markov	18.29%	28.01%	29.59%	6.57%	13.36%	15.81%	6.32%	11.57%	13.12%	5.35%	10.10%	11.71%
PRME	12.83%	24.38%	19.66%	2.36%	6.27%	8.29%	2.36%	6.27%	8.29%	1.57%	4.10%	5.91%
ST-RNN	18.64%	28.01%	30.23%	6.64%	15.01%	18.34%	6.32%	11.38%	13.37%	5.23%	10.30%	11.94%
Bi-LSTM	18.42%	28.85%	31.27%	6.19%	13.67%	17.58%	5.65%	11.14%	13.67%	5.01%	10.15%	11.68%
POI2Vec	18.79%	28.85%	30.81%	6.89%	14.87%	18.27%	6.32%	11.21%	13.52%	4.64%	8.77%	10.23%
DeepMove	19.29%	35.37%	38.35%	8.82%	17.93%	22.80%	6.65%	13.43%	15.87%	6.62%	13.79%	17.15%
VANext	21.09%	38.22%	44.88%	11.60%	24.53%	30.25%	7.29%	15.78%	20.21%	7.34%	14.71%	17.78%
VANext-S	**23.21%**	**41.03%**	**47.45%**	**12.13%**	**24.78%**	**30.69%**	**7.69%**	**16.98%**	**21.36%**	**8.35%**	**17.16%**	**20.31%**

trick in NLP, we are the first to adapt it in a variational autoencoder framework for learning human mobility. Since this is not core part of this paper, we focus on the VANext in following experiments for further explanations.



(a) New York. (b) Singapore.

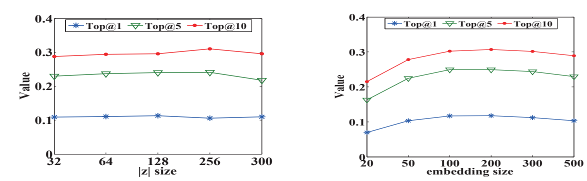
Figure 3: Module performance comparison.

Individual Module Performance. To investigate the effectiveness of various components in VANext, we conduct experiments by manipulating modules in different ways. Specifically, we replace the attention mechanism of *DeepMove* with our proposed variational attention module, yielding a new method *VaDeepMove*. Thus, VaDeepMove is similar to VANext only except the historical trajectory learning – VANext uses CNN while GRU units are employed in VaDeepMove. We compare our VANext with VaDeepMove and DeepMove upon New York and Singapore datasets, as illustrated in Figure 3(a) and 3(b). The result clearly indicates the superiority of CNN and variational attention used in VANext. Further, it proves that CNN exhibits better structural learning than RNN in the case of extremely long history trajectories which encodes more structural and periodical patterns of human mobility. The results of VaDeepMove and DeepMove reflect the advantage of variational attention which is also their difference.

Efficiency. We now investigate the efficiency of the proposed methods. Table 4 shows the time consumed when training VANext (CNN units) and VaDeepMove (RNN units). Note that both VANext and (Va)DeepMove use RNN for training current trajectory T_c. However, it is negligible due to few POIs in most current trajectory. Obviously, CNN based historical trajectory learning significantly reduces the training time. Combined with above results on learning structural/periodical patterns of historical trajectories, we conclude that, arguably, CNN is the most suitable method for learning human mobility, at least for long term motion patterns.

Table 4: Comparison of time consumption during the model training. $|T_c|$: average length of the current trajectories; $|T_h|$: average length of the historical trajectories.

City in Dataset	$\lvert T_c\rvert$	$\lvert T_h\rvert$	Cell	Time Cost(min)
New York	3.23	207.70	RNN CNN	≈487 ≈277
Singapore	2.92	119.80	RNN CNN	≈267 ≈117
Houston	9.53	1027.94	RNN CNN	≈1810 ≈463
California	3.64	129.66	RNN CNN	≈790 ≈247



(a) Sensitivity of $|\mathbf{z}|$ size. (b) Sensitivity of embedding.

Figure 4: Sensitivity of parameters in Singapore data.

Parameter sensitivity. Finally, Figure 4(a) evaluates impact of latent variable size $|\mathbf{z}|$ and causal POI embedding size d on the performance of VANext. Simply speaking, VANext demonstrates robust performance to the two parameters and achieves stable results when both are small. We observe similar robust results for other parameters and omit to report due to the space limitation.

6 CONCLUSIONS AND FUTURE WORK

We propose a novel model VANext and its variant VANext-S to learn human mobility patterns for next POI prediction. It introduces an attention mechanism affected by the latent variable learned from current mobility to capture the moving pattern and hierarchical semantics of historical trajectories. We also apply convolutional operations on trajectory to greatly improve efficiency over the existing RNN-based methods. Part of our future work will focus on incorporating other contextual POIs attributes (e.g., ”museum”; ”restaurant”) to further improve VANext effectiveness. Another possible extension is to include transportation modes (e.g., ”walk”; ”bus”; ”taxi”) of users trajectories for performance boosting.

ACKNOWLEDGMENT

This work was supported by National Natural Science Foundation of China (Grant No.61602097 and No.61472064), NSF grants III 1213038 and CNS 1646107, and ONR grant N00014-14-10215.

REFERENCES

[1] Michael Backes, Mathias Humbert, Jun Pang, and Yang Zhang. 2017. walk2friends: Inferring Social Links from Mobility Profiles. In *SIGSAC*.

[2] Chen Cheng, Haiqin Yang, Irwin King, and Michael R Lyu. 2012. Fused Matrix Factorization with Geographical and Social Influence in Location-Based Social Networks.. In *AAAI*.

[3] Chen Cheng, Haiqin Yang, Michael R Lyu, and Irwin King. 2013. Where You Like to Go Next: Successive Point-of-Interest Recommendation.. In *IJCAI*.

[4] Junyoung Chung, Caglar Gulcehre, Kyung Hyun Cho, and Yoshua Bengio. 2014. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *Eprint Arxiv* (2014).

[5] Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron C Courville, and Yoshua Bengio. 2015. A Recurrent Latent Variable Model for Sequential Data. In *NIPS*.

[6] Jie Feng, Yong Li, Chao Zhang, Funing Sun, Fanchao Meng, Ang Guo, and Depeng Jin. 2018. DeepMove: Predicting Human Mobility with Attentional Recurrent Networks. In *WWW*.

[7] Shanshan Feng, Gao Cong, Bo An, and Yeow Meng Chee. 2017. POI2Vec: Geographical Latent Representation for Predicting Future Visitors.. In *AAAI*.

[8] Shanshan Feng, Xutao Li, Yifeng Zeng, Gao Cong, Yeow Meng Chee, and Quan Yuan. 2015. Personalized Ranking Metric Embedding for Next New POI Recommendation.. In *IJCAI*.

[9] Sébastien Gambs, Marc-Olivier Killijian, and Miguel Núñez del Prado Cortez. 2012. Next Place Prediction Using Mobility Markov Chains. In *MPM*.

[10] Huiji Gao, Jiliang Tang, Xia Hu, and Huan Liu. 2015. Content-aware Point of Interest Recommendation on Location-based Social Networks. In *AAAI*.

[11] Qiang Gao, Fan Zhou, Kunpeng Zhang, Goce Trajcevski, Xucheng Luo, and Fengli Zhang. 2017. Identifying human mobility via trajectory embeddings. In *IJCAI*.

[12] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional sequence to sequence learning. In *ICML*.

[13] Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P Xing. 2017. Toward Controlled Generation of Text. In *ICML*.

[14] Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).

[15] Xutao Li, Gao Cong, Xiao-Li Li, Tuan-Anh Nguyen Pham, and Shonali Krishnaswamy. 2015. Rank-GeoFM: A Ranking Based Geographical Factorization Method for Point of Interest Recommendation. In *SIGIR*.

[16] Defu Lian, Cong Zhao, Xing Xie, Guangzhong Sun, Enhong Chen, and Yong Rui. 2014. GeoMF: Joint Geographical Modeling and Matrix Factorization for Point-of-interest Recommendation. In *KDD*.

[17] Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. 2016. Predicting the Next Location : A Recurrent Model with Spatial and Temporal Contexts. In *AAAI*.

[18] Yiding Liu, Tuan-Anh Pham, Gao Cong, and Quan Yuan. 2017. An Experimental Evaluation of Point-of-interest Recommendation in Location-based Social Networks. *PVLDB* (2017).

[19] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).

[20] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing Personalized Markov Chains for Next-basket Recommendation. In *WWW*.

[21] Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing* 45, 11 (1997), 2673–2681.

[22] Iulian Vlad Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron C Courville, and Yoshua Bengio. 2017. A Hierarchical Latent Variable Encoder-Decoder Model for Generating Dialogues. In *AAAI*.

[23] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *NIPS*.

[24] Hongjian Wang, Zhenhui Li, and Wang Chien Lee. 2015. PGT: Measuring Mobility Relationship Using Personal, Global and Temporal Factors. In *ICDM*.

[25] Weidi Xu, Haoze Sun, Chao Deng, and Ying Tan. 2017. Variational Autoencoder for Semi-Supervised Text Classification. In *AAAI*.

[26] Carl Yang, Lanxiao Bai, Chao Zhang, Quan Yuan, and Jiawei Han. 2017. Bridging Collaborative Filtering and Semi-Supervised Learning: A Neural Approach for POI Recommendation. In *KDD*.

[27] Cheng Yang, Maosong Sun, Wayne Xin Zhao, Zhiyuan Liu, and Edward Y. Chan. 2017. A Neural Network Approach to Jointly Modeling Social Networks and Mobile Trajectories. *ACM Trans. Inf. Syst.* 35, 4 (2017), 36:1–36:28.

[28] Dingqi Yang, Daqing Zhang, Vincent W Zheng, and Zhiyong Yu. 2015. Modeling user activity preference by leveraging user spatial temporal characteristics in LBSNs. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 45, 1 (2015), 129–142.

[29] Di Yao, Chao Zhang, Jianhui Huang, and Jingping Bi. 2017. SERM: A Recurrent Model for Next Location Prediction in Semantic Trajectories. In *CIKM*.

[30] Quan Yuan, Gao Cong, Zongyang Ma, Aixin Sun, and Nadia Magnenat Thalmann. 2013. Time-aware point-of-interest recommendation. In *SIGIR*.

[31] Shenglin Zhao, Tong Zhao, Irwin King, and Michael R Lyu. 2017. Geo-Teaser: Geo-Temporal Sequential Embedding Rank for Point-of-interest Recommendation. In *WWW*.

[32] Vincent Wenchen Zheng, Bin Cao, Yu Zheng, Xing Xie, and Qiang Yang. 2010. Collaborative Filtering Meets Mobile Recommendation: A User-centered Approach. In *AAAI*.

[33] Chunting Zhou and Graham Neubig. 2017. Morphological Inflection Generation with Multi-space Variational Encoder-Decoders. In *CoNLL*.

[34] Fan Zhou, Qiang Gao, Kunpeng Zhang, Goce Trajcevski, Zhong Ting, and Fengli Zhang. 2018. Trajectory-User Linking via Variational AutoEncoder. In *IJCAI*.