Multi-Task Travel Route Planning With a Flexible Deep Learning Framework

Feiran Huang, Jie Xu, and Jian Weng, *Member, IEEE*

***Abstract*— Travel route planning aims to map out a feasible sightseeing itinerary for a traveler covering famous attractions and meeting the tourist's desire. It is very useful for tourists to plan their travel routes when they want to travel at unfamiliar scenic cities. Existing methods for travel route planning mainly concentrate on a single planning problem for a special task, but is not capable of being applied to other tasks. For example, previous must-visit planning methods cannot be applied to the next-point recommendation, despite these two tasks are closely related to each other in travel route planning. Besides, most of the existing work do not consider the important auxiliary information such as Point of Interests (POI) attributes, user preference, and historical route data in their approaches. In this paper, we propose a flexible Multi-task Deep Travel Route Planning framework named MDTRP to integrate rich auxiliary information for more effective planning. Specifically, we first construct a heterogeneous network through the relations between users and POIs and employ a heterogeneous network embedding method to learn the features of users and POIs. Then we present an attention-based deep model to integrate the auxiliary information and focus on important visited points for the prediction of next POIs. Finally, a beam search algorithm is introduced to flexibly generate multiple feasible route candidates for three types of planning tasks (next-point recommendation, general route planning, and must-visit planning). We introduce six public datasets to conduct extensive experiments, of which the results demonstrate the flexibility and superiority of the proposed approach in travel route planning.**

***Index Terms*— Travel route planning, deep learning, heterogeneous network embedding, attention model.**

I. Introduction

TRAVELING is a very important part of life as it allows us to see different cultures and diverse lifestyles. Route planning is an effective method to optimize the route such that less time is spent on driving or riding, and more time can be spent on visiting interesting attractions. Especially for tourists coming to an unfamiliar city, travel route planning can heavily reduce the time of travel preparation and recommend considerable travel routes fitting the various demands of tourists [1]–[4]. Therefore, how to automatically generate travel route plan to help tourists enjoy better travel experience has been increasingly attracting attention in both academia and industry.

Existing methods for Travel route planning can be roughly divided into three categories. Early methods calculate the total geographical distance among different Point of Interests (POIs) chronologically to find the shortest travel path, which is also called *traveling salesman problem (TSP)* [2], [5]–[7]. TSP is widely used in trip planning because it can recommend the shortest possible tour routes, thus largely saves traveling time. The second strategy tries to treat this problem as route recommendation modeling, which aims to generate the candidate routes based on the historical travel route data of the visitors [8]–[14]. Therefore, a large number of approaches in the field of recommendation systems have been employed to solve the travel route planning problem. The last category of approaches combines both POI or user preference with historical route data with hybrid models for joint optimization [15]–[19].

Although many methods have been proposed to study the problem of travel route planning, there still exist two major limitations for the existing approaches. First, these works mainly concentrate on a single planning problem on a special task. However, different tourists usually have different demands for traveling, thus designing a specific model is not capable of dealing with various kinds of tour demands. We collect hundreds of questions submitted by travelers from several popular tourism platforms (e.g., Expedia, Ctrip). After analysis of the collected data, three most common tasks of travel route planning have been summarized as shown in Table I. For example, [20]–[23] focus on the task of *next-point recommendation* while [2], [8], [24] try to deal with the task of *general route planning*. It is difficult to apply these methods to other tasks directly. Therefore, a more general and flexible route planning approach is needed to adapt to more tasks. Second, multiple types of data from different sources exist in the scenario of travel route planning, such as POI attributes, user preference, and historical route data of existed users. Since each of these data contains information not found in other data, merging these data together will be helpful in designing a more effective travel route planning model. However, most of existing methods just consider single data source and cannot integrate data from multi-sources effectively. In addition, there exist close correlations between

Manuscript received January 26, 2020; revised March 14, 2020; accepted April 9, 2020. Date of publication June 11, 2020; date of current version July 12, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant U1736203, Grant 61825203, Grant 61906075, Grant 61932010, and Grant 61732021, in part by the Major Program of Guangdong Basic and Applied Basic Research under Grant 2019B030302008, in part by the National Key Research and Development Program of China under Grant 2018YFB1402600, Grant SQ2017YFGX110188-03, and Grant 2017YFB0802203, in part by the Guangdong Science and Technology Program under Grant 2017B010111005 and Grant 2019A1515011920, and in part by the Guangzhou Science and Technology Program under Grant 201802010061. The Associate Editor for this article was S. Mumtaz. *(Corresponding author: Jian Weng.)*

Feiran Huang and Jian Weng are with the College of Cyber Security and the College of Information Science and Technology, Jinan University, Guangzhou 510632, China, and also with the Guangdong Key Laboratory of Data Security and Privacy Preserving, Guangzhou 510632, China (e-mail: huangfr@jnu.edu.cn; cryptjweng@gmail.com).

Jie Xu is with the Beijing Key Laboratory of Network Technology, Beihang University, Beijing 100191, China (e-mail: xujie1020@buaa.edu.cn).

Digital Object Identifier 10.1109/TITS.2020.2987645

TABLE I
THREE MOST COMMON ROUTE PLANNING TASKS

Task	Description
Next-Point Recommendation	For tourists who have visited target cities several times, they want to be recommended which POI to visit next. POI attributes, tourist preference, and time consuming will be taken into account.
General Route Planning	For tourists who visit a city for the first time, they need appropriate routes that contain hot interests and have the least detours. Meanwhile, the time budget for traveling is limited.
Must-Visit Planning	For tourists who are eager to visit some specific POIs, travel routes covering these must-visit POIs are preferred. Tourists need routes that schedule the user-preferred POIs and other appropriate POIs in a reasonable order.

the different types of information, e.g., the user preference can be learned based on the features of POI the user visited, while POI features can be further improved by analyzing all the users who have traveled the POI. Traditional approaches, such as Hidden Markov Model (HMM) [25] and topic model [26], [27], are not effective to excavate the deep non-linear correlations among the data from multi-sources.

In this work, our aim is to integrate POI attributes, user preference, and historical route data while fitting different task requirements of travelers for travel route planning. It is of great difficulty due to the challenges as follows. First, there exist various constraints including travel cost, time, and must-visit point restriction in the above-mentioned tasks. All of these factors could affect the process of travel routes generation. It is non-trivial to design a general and flexible framework which well meets these constraints. Second, although it is expected that incorporating POI features, user preference, and historical route data can achieve better performance, it still needs to explore how to make full use of these data under a unified framework. Close correlations exist between the data from different sources, and it is not easy to fully excavate the correlations.

To deal with the challenges, a flexible Multi-task Deep Travel Route Planning framework named MDTRP is proposed. It can effectively integrate user preference, POI attributes, and historical route data in a holistic model, and can be applied to multiple travel route planning tasks at the same time. We regard the problem of route planning as a process of sequence generation, which is completely suitable to be addressed by deep neural networks. The proposed MDTRP is generally composed of three stages: In the feature extraction stage, we employ a heterogeneous network embedding method to learn the features of users and POIs. The historical travel route is also converted to a set of POI patterns and next POIs. In the model learning stage, a novel deep attention model is introduced to learn the probability of visiting the next POIs given the input patterns. The proposed model contains a Long Short-Term Memory (LSTM), a points attention network, and a Multi-layer Perceptron (MLP), which can learn the probability distribution of the next POIs. Finally, a beam search algorithm is introduced to flexibly generate multiple feasible route candidates for three types of planning tasks (next-point recommendation, general route planning, and must-visit planning). For each type of the three tasks of travel route planning, we design an appropriate generation strategy considering user preference and demands. The main contributions of this work can be summarized as follows:

- A general and flexible framework named MDTRP is proposed for travel route planning. The proposed approach can be applied to the three most common route planning tasks, i.e., next-point recommendation, general route planning, and must-visit planning.
- A novel points attention-based deep model is proposed to learn the probability distributions of next visiting POIs given the previous points in a route. The proposed model can effectively integrate the data from multi-sources and automatically focus on important visited points.
- Based on the predicted distributions of POIs, a beam search algorithm is further employed to generate candidate routes for different travel route planning tasks, in which various types of user demands can be integrated.
- Six public datasets are introduced to evaluate the proposed approach with extensive experiments. Evaluation results demonstrate the superiority of MDTRP over state-of-the-art models.

This paper is an extended version of its preliminary paper [28] with additional contributions of both modeling and experiments. First, we employ heterogeneous network embedding methods to learn the features of users and POIs, which can represent the relations between users and POIs more effectively. Second, we build a points attention network upon the LSTM to deeply excavate the correlation between the predicted POI and previous points for more accurate prediction. Third, two more datasets are introduced and more extensive experiments are conducted to evaluate the proposed method.

The remainder of this paper is organized as follows. The related work is reviewed in section II, we formulate the problem in Section III, and then present the details of MDTRP in section IV. The experiments are presented in section V. Finally, we draw conclusions and discuss future work.

II. RELATED WORK

In recent years, travel route planning, which helps build an efficient and sustainable transportation system in a smart city [29]–[32], has drawn many research interests. A desirable route planning approach not only can help tourists better enjoy their trips, but also contribute to free them from the time-consuming and trivial preparation works. Existing related works can be roughly divided into three categories.

Traditional methods try to find the shortest geographical path connecting a given departure POI and destination POI, which is also called *traveling salesman problem* [2], [5]–[7], [33], [34]. Matai *et al.* [33] construct the solution greedily by choosing the trajectory of the nearest POI with a TSP greedy heuristic method. Brilhante *et al.* [2] propose an unsupervised framework to generate the budgeted trajectory by composing the popular itineraries with a specific instance of the Traveling Salesman Problem, aiming to seek out the shortest route for planning personalized sightseeing tours. Beirigo and dos Santos [5] propose a parallel Iterated Local Search (ILS) heuristic to decrease time-consuming by searching promising candidate routes in realistic tour networks. Faigl *et al.* [34] build an unsupervised algorithm based on the self-organizing map (SOM) for the Traveling salesman problem. The generalized traveling Orienteering problem can be well addressed by collecting rewards through traveling around the neighborhood of locations. Rani *et al.* [7] combine k-means clustering method with traveling salesman problem (TSP) to generate tour itineraries. Both the distance and travel time are considered to recommend an optimal itinerary.

The second strategy tries to treat the route planning task as route recommendation modeling, which aims to generate the candidate routes based on the historical travel route data of the visitors [8]–[12]. Zheng and Xie [9] propose a travel recommendation model by incorporating hypertext induced topic search and collaborative filtering, which provide not only top interesting points and routes as generic recommendation but also preferred individual locations as personalized recommendation. Zheng *et al.* [10] analyze users' tourist movement patterns and topological characteristics of travel routes by leveraging social photos with corresponding timestamps, tags, and geographic location. Kurashima *et al.* [11] combine Markov models with topic models to perform travel route recommendation by considering both POI information and user preference. Xu *et al.* [12] recommend travel locations based on topic distribution of the user tour histories in other cities and given contexts such as weather and season.

The third category of approaches usually combines both POI features and historical route data with hybrid models for joint optimization [15]–[19], [35]. Rodrígues *et al.* [35] propose a mathematical method and an interactive multi-criteria design to plan trajectories that satisfies the demands of users. Brilhante *et al.* [15] regard the planning tourism task as an special case of the problem of Generalized Maximum Coverage, by maximizing the measurement of interest for the traveler given visiting time budget and user preference. Yu *et al.* [17] propose a heuristic search-based travel route planning algorithm with a prototype system to generate travel packages containing multiple POIs and sequences. Wen *et al.* [18] introduce knowledge extracted from users' historical records and social interaction for travel route planning with an efficient representative model. A keyword extraction module is proposed to classify POI-related tags for matching queries, and a route reconstruction algorithm is designed to generate route candidates.

The major limitation of related research is that they are not capable of incorporating POI attributes, user preference, historical route data with a unified framework effectively. In addition, most of existing methods are designed for a specific travel task, and it is difficult to apply these methods to a new travel planning task directly with different tour demands.

III. Problem Statement

Travel route planning helps tourists optimize the tourism route, thus enjoy a better travel experience. The existing planning methods are insufficient to well solve the problem of travel route planning due to the above-mentioned reasons. Different from existing methods of travel route planning, the proposed MDTRP can integrate POI attributes, user preference, and historical route data into a holistic framework. In addition, due to the generality and flexibility of the proposed model, it can be applied to three most common route planning tasks, i.e., next-point recommendation, general route planning, and must-visit planning. Next, we will introduce several key terminologies and then formally define the studied problem of travel route planning.

A. Key Terminologies

Definition 1 (POI Attribute Vector): Given the set of POIs $\mathcal{P} = \{p_1, \ldots, p_{|\mathcal{P}|}\}$, where each element p_i represents a specific travel attraction. We define the attribute vector of POI p_i as: $v_{p_i} \in \mathbb{R}^{1\times m}$, where m is the dimension of attribute vector. Vector v_{p_i} preserves attribute information of POI p_i, such as category (e.g., park, religious, shopping), popularity, geographical coordinates and visit time interval.

Definition 2 (User Preference Vector): Given the POI category set $\mathcal{C}$ and a tourist u_i in the tourist set $\mathcal{U}$, we define the user preference vector $v_{u_i} \in \mathbb{R}^{1\times|\mathcal{C}|}$ to measure the normalized interests of u_i on different POI categories. Elements in v_{u_i} represent the interests distribution of u_i based on the number of his/her previous visiting times to different POI categories.

Definition 3 (Historical Travel Routes): Historical route set $\mathcal{T}$ contains travel routes shared by other tourists. A single route is defined as $T_u = ((p_1), \cdots, (p_k))$, which is an ordered sequence of k POIs visited by tourist u at a time.

B. Problem Definition

Based on the above definitions, we formally define the studied problem as follows.

Definition 4 (Travel Route Planning): Given the POI set $\mathcal{P}$, historic travel route set $\mathcal{T}$ and the corresponding tourist set $\mathcal{U}$, we aim to generate a personalized travel route for each user. Specifically, we firstly try to learn the probabilities of visiting the next POI given the previous POIs with their features and the tourist preference: $P(p_t|u_i, p_{t-1}, p_{t-2}, \cdots), \forall p_t \in \mathcal{P}$. Then based on the learned probability, we try to generate a desirable candidate route for a new user considering his preference and demands.

IV. Multi-Task Deep Travel Route Planning

In this section, we first introduce the framework of MDTRP. Then we elaborate on the three stages of the approach, i.e., feature extraction stage, model learning stage, and route generation stage.

A. Framework

The framework of the proposed MDTRP is presented in Figure 1. Given a city for traveling, we first collect the historical travel routes shared by other tourists, the corresponding POI attributes, and user preference as the input to our method. In the feature extraction stage, we transform the visit relations between users and POIs to a heterogeneous network, which has two types of nodes i.e., users and POIs, and has

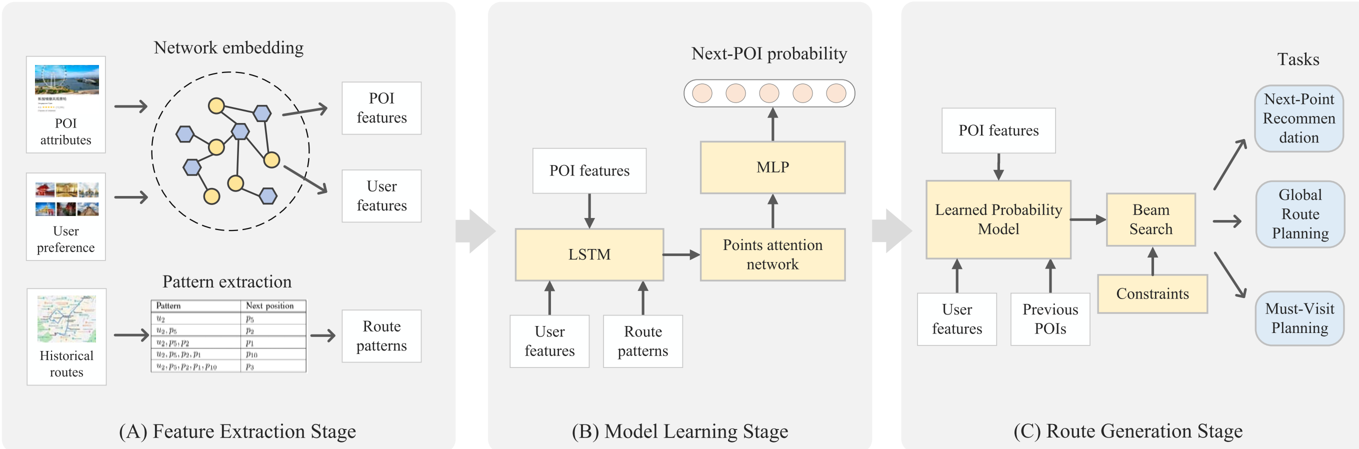

Fig. 1. The framework of MDTRP for multi-task travel route planning.

edges to represent the visiting relations between node pairs. A heterogeneous network embedding method is then employed to learn the features of users and POIs. Each historical travel route is also converted to a set of POI patterns and their next POIs as route features and corresponding labels. In the model learning stage, a novel deep attention model is introduced to learn the probability of visiting the next POIs given the input pattern. The proposed framework contains two major components: A Long Short-Term Memory (LSTM), a points attention network, and a Multi-layer Perceptron (MLP). The LSTM is used to encode the input route patterns and user features into a low-dimensional representation vector, which preserves the corresponding POI property and user information. The points attention network automatically focuses on important visited POIs for more effective representation learning. With the learned sequence feature vector, we utilize the MLP model to learn the probability distribution of the next POIs. Finally, in the route generation stage, based on the predicted probability distribution, a beam search strategy is employed to generate the candidate routes as well as satisfying constraints in different tasks. This method is flexible as it can be used to generate travel routes satisfying different constraints in multiple travel route planning tasks.

B. Feature Extraction Stage

In this subsection, we present the feature extraction process of POIs, users, and routes. We first collect the historical travel routes shared by other tourists, the corresponding POI attributes, and user preference from users' previous visiting times to different POIs as the input vectors. For the POIs and users, we transform the POI attributes and user preference to a heterogeneous network and employ a heterogeneous network embedding method to extract the user and POI features. For routes, we transform each historical travel route into a set of POI patterns and their next POIs as route pattern features and labels.

1) POI and User Features: Given POI attributes and user preference, we can obtain the POIs each user has visited and how many times each POI has been visited by the user. Motivated by recent works on graph embedding [36]–[39], we consider the relations between POIs and users as a heterogeneous network, which has two types of nodes, i.e., users and POIs, and has edges to represent the visiting relations between node pairs. We hope that through network embedding, the user preference features can be improved based on the visited POIs, while POI features can be further learned by analyzing all the users who have traveled the POI. The heterogeneous graph can be formally defined as $G = \{V, T, E\}$, in which V is the set of nodes, T denotes the set of note types, including POI (P) and user (U), E represent undirected edges between POI and user pairs. To leverage these visiting relations, we introduce a heterogeneous network embedding method to learn the latent feature vector for each POI and user.

To learn the node features, we first use Skip-gram model [40] to transform the network structure into meta-paths. Formally, given a predefined meta-path UPU, the transition probability of the i-th step in the random walk can be denoted as:

$$p(v^{i+1}|v^i) = \begin{cases} \frac{1}{|N_U(v^i)|} & (v^i, v^{i+1}) \in E,\ v^i \in U,\ v^{i+1} \in P \\ \frac{1}{|N_P(v^i)|} & (v^i, v^{i+1}) \in E,\ v^i \in P,\ v^{i+1} \in U \\ 0 & Others \end{cases} \tag{1}$$

where $N_U(v^i)$ is number of neighbor nodes of the center node v_i with the category of U. Then Skip-gram model is used to embed the node into representation vectors. Like other embedding methods such as node2vec [41] and DeepWalk [42], the network probability is maximized according to the local structures. The objective of the heterogeneous network embedding can be write as:

$$\arg\min_{\theta} \sum_{v \in V} \sum_{t \in T} \sum_{c_t \in N_t(v)} log(p(c_t|v;\theta)) \tag{2}$$

where T is the set of node types and $N_{t(v)}$ denotes the type t neighbor nodes of the target node v. Similar to [40], [43], $logp(c_t|v;\theta)$ is calculated by a $softmax$ function. To optimize the objective function efficiently, negative sampling [40] is also employed to sample several nodes to construct $softmax$. After the learning process, node embedding of the heterogeneous network can be obtained as the POI and user features. Specifically, the original user preference vector

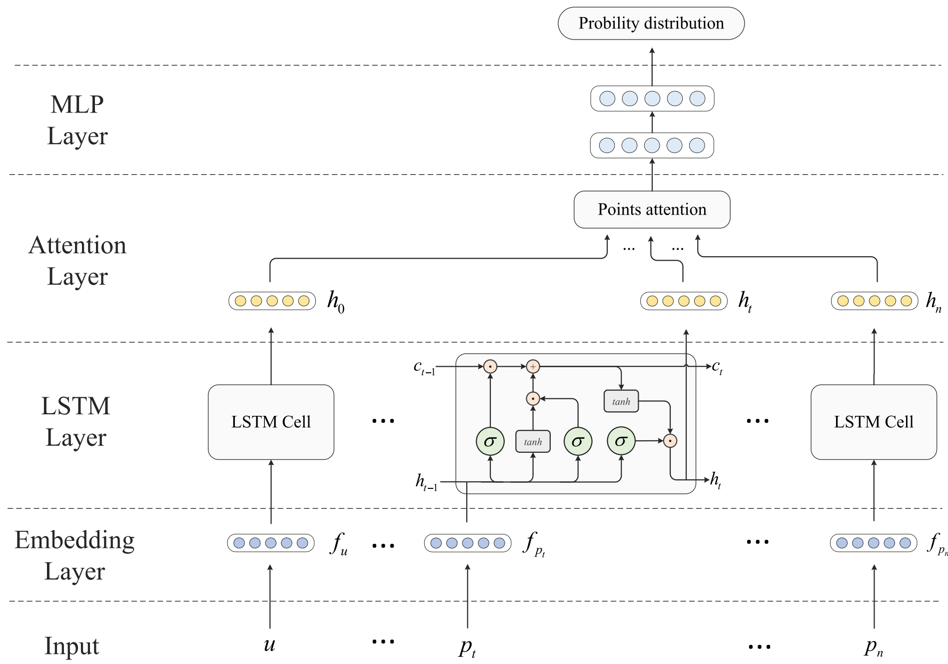

Fig. 2. The process of model learning stage.

TABLE II
AN EXAMPLE OF ROUTE PATTERN GENERATION

(a) A sample of input route.

Travel route	$\{p_5, p_2, p_1, p_{10}, p_3\}$
User	u_2

(b) Patterns extracted from the above data.

Pattern	Next position
u_2	p_5
u_2, p_5	p_2
u_2, p_5, p_2	p_1
u_2, p_5, p_2, p_1	p_{10}
$u_2, p_5, p_2, p_1, p_{10}$	p_3

of i-th node v_{u_i} is transformed to f_{u_i}, and the original POI attribute of jth node v_{p_j} is transformed to f_{p_j}.

2) Route Pattern Features and Labels: Given a user and his historical routes in a particular city, we need to transform each input route into a set of POI patterns and their next POIs for training. As shown in Table II, given a travel route, the user feature and the first few POIs are merged as an input pattern, and the next POI is regarded as the predicted POI as the label. A straight forward method to learn the probability of next POI is to count the frequencies of POIs given each training pattern, and then apply the prior probabilities to predict the next POI given a test pattern. Although this strategy cannot cover patterns which have never appeared in the training set, most statistic characteristics of the route patterns can be captured. After this process, route pattern features and the corresponding labels can be obtained.

C. Model Learning Stage

We describe the detailed process of model learning in this subsection. Related works often encode predefined assumptions into the learning process [24], [35], such as users tending to visit the nearest POI, which is not always true due to the rapid development of public transportations. Nowadays there are plenty of visit routes shared by other tourists in a city. These valuable historical routes can reflect the key factors that affect tourist's choices, such as the tourist preference, distances between POIs or POI categories. Therefore, a data-driven model is proposed to exploit data from multiple sources for travel routes planning.

Based on the route pattern $x = \{u, p_1, p_2, \ldots, p_n\}$, we utilize a deep learning model to learn such a probability: $\{P(p_i|x), \forall p_i \in \mathcal{P}\}$. As shown in Fig. 2, the proposed model is convenient to address the multi-sourced data. The input is the route patterns, with the user features and POI features. We propose a points attention-based LSTM (Long Short Term Memory) model [44] encode the input pattern, with focusing on important points, and the corresponding POI features into a distributed vector. LSTM can process input sequences of arbitrary length [45] and can learn long-term dependencies [46]. Moreover, the incorporated points attention mechanism is capable of automatically concentrating on important visited POIs through parameter learning. In many cases, some previous POIs in a route are more important than other POIs to infer the next POI. Since the attention mechanism has been proved to be effective in many vision and language tasks [47]–[52], we employ points attention to capture this type of focusing relation. The output is the final probability distribution on the POIs for next position considering the user features. Next, the detailed structure of the proposed model is introduced as follows:

- **Input**: The input pattern $x = \{u, p_1, p_2, \ldots, p_n\}$ with non-fixed length of $n + 1$ is used as the input.
- **Embedding Layer**: The embedding layer map each POI p_i to its corresponding feature $f_{p_i} \in \mathbb{R}^{1\times m}$. The user

u in the pattern is also embedded to corresponding preference feature $f_u \in \mathbb{R}^{1\times|\mathcal{C}|}$, where $\mathcal{C}$ is the set of POI categories. LSTM model requires the embedding of each vector in the input sequence be the same size. Therefore, the padding strategy [53] is employed to insert zeros to fill the shorter vectors, which is commonly used in sequence based deep models. After padding, each vector in the pattern sequence is mapped to a $maximum(m, |\mathcal{C}|)$-dimensional feature. Here we set $m = maximum(m, |\mathcal{C}|)$. After embedding, the input pattern features to the next layer is $y = \{f_u, f_{p_1}, f_{p_2}, \ldots, f_{p_n}\}$.

- **LSTM Layer**: The LSTM layer encodes the input sequence into a $d-$dimensional representation vector through the recursive of a transition function on a recurrent state h_t. For the user u, the input of LSTM layer is f_u, and the output is the recurrent state h_0. For the $t-$th POI p_t, the input is f_{p_t}, and the output is h_t. The recurrent state h_t is a non-linear transformation of input vector and its previous recurrent state h_{t-1}. LSTM unit utilizes an input gate i_t, an forget gate f_t and an output gate o_t to control the information transferred. The specific parameterization of LSTM is defined by the following equations:

$$\begin{aligned} i_t &= \sigma(W^{(i)}v_{p_t} + U^{(i)}h_{t-1}) + b^{(i)} \\ f_t &= \sigma(W^{(f)}v_{p_t} + U^{(f)}h_{t-1}) + b^{(f)} \\ o_t &= \sigma(W^{(o)}v_{p_t} + U^{(o)}h_{t-1}) + b^{(o)} \\ u_t &= tanh(W^{(u)}v_{p_t} + U^{(u)}h_{t-1}) + b^{(u)} \\ c_t &= i_t \otimes u_t + f_t \otimes c_{(t-1)} \\ h_t &= o_t \otimes tanh(c_t) \end{aligned} \tag{3}$$

where $\sigma(\cdot)$ is a sigmoid function, $\otimes$ is element-wise multiplication and $\tanh(\cdot)$ is a hyperbolic tangent function. $W^{(i)}$, $W^{(f)}$, $W^{(o)}$, $W^{(u)}$, $U^{(i)}$, $U^{(f)}$, $U^{(o)}$, $U^{(u)}$, $b^{(f)}$, $b^{(o)}$, $b^{(u)}$ are parameters to be trained.

- **Points Attention Layer**: Traditional ways take the final recurrent state vector h_n as the sequence representation [46]. However h_n largely relies on the latter part of the input sequence, and directly using it may introduce noises. Another strategy is to use mean pooling layer [28], [54], which generate representation vector by averaging the state values h_t at every time step. However, average pooling treats every POI equally, but in many cases, some specific previous POIs in a route are more important. Hence, we employ points attention to capture this type of focusing relation. For each time-step, an attention score α_t is assigned to each POI (or the user) based on its importance for next point prediction:

$$\begin{aligned} \alpha_t &= \frac{exp(e_t)}{\sum_{t=0}^{n} exp(e_t)} \\ e_t &= tanh(W^{(a)}h_t + b^{(a)}) \end{aligned} \tag{4}$$

where e_j is the unnormalized attention score which measures how well the vector h_t helps prediction, $W^{(a)}$ and $b^{(a)}$ are parameters to be learned. α is used to normalize the attention over the state sequence $\{h_0, h_1, \ldots, h_n\}$. The attended state features is then calculated as the weighted average over the sequence:

$$h = \sum_{t=0}^{n} \alpha_t h_t \tag{5}$$

- **MLP**: For the MLP, we use a two-layer fully connected network, of which the representation vector h is taken as input. The output vector $y' \in \mathbb{R}^{1\times|\mathcal{P}|}$ contains the conditional probability over all POIs given the input route patterns and the user feature. The formulation of this layer is:

$$\mathcal{O} = W^{(2)}(tanh(b^{(1)} + W^{(1)}o_{merge})) + b^{(2)} \tag{6}$$

where $W^{(1)}$,$W^{(2)}$ and parameter matrices, $b^{(1)}$,$b^{(2)}$ are biases. Then a softmax function is used to activating the output layer:

$$y'_i = P(p_i|x) = \frac{e^{\mathcal{O}_i}}{\sum_{k=1}^{|\mathcal{P}|} e^{\mathcal{O}_k}}$$

For the input pattern x and its next position p_j, categorical cross-entropy is introduced as the loss function:

$$L = -\sum_{k=1}^{|\mathcal{P}|} y_k \cdot log(y'_k) \tag{7}$$

where

$$y_k = \begin{cases} 1, & k = j \\ 0, & k \neq j \end{cases}$$

is the groundtruth label. We utilize the Adam [55] optimizer to minimize the loss L. The parameters in the model are automatically learned by mini-batch back propagation.

After learning of the model, we can achieve a trained model which can feed an input pattern to calculate the probability distribution of the next POI. This stage lays a solid foundation for the route generation stage.

D. Route Generation Stage

In this subsection, the detailed route generation strategies designed for three route planning tasks are presented. First, the POI prediction process is introduced for the task of next-point recommendation. Then a beam search method is described to generate candidate routes for general route planning. Finally, the generation approach for must-visit planning is presented.

1) Next-Point Recommendation: For the next-point recommendation problem, the learned model can be directly used to predict the next visiting position. Algorithm 1 presents the detailed recommending process. The input to our method consists of a POI pattern, POI attributes, and user preference. Our method can still recommend the first POI if only the target user is included in the input pattern. y' is the output of Algorithm 1, which provides the probability distribution of next POI. The top K probable POIs satisfying the constrains of time limitation are chosen as the final recommending results.

2) General Route Planning: General route planning differs from next-point recommendation, because it needs to generate a full route instead of predicting a single POI. To this end, a greedy beam search algorithm with pruning is introduced to find top-K candidate routes [56]. As a heuristic search algorithm, beam search is an optimization of the best-first

Algorithm 1 The Algorithm for Next-Point Recommendation

Input:

$Input1$: input pattern $\{u_i, p_1, p_2, \cdots, p_n\}$,

$Input2$: user preference vector v_{u_i},

$Input3$: POI attribute vectors $v_{p_1}, v_{p_2}, \ldots, v_{p_n}$,

Output: POI probability vector $y' \in \mathbb{R}^{1\times|\mathcal{P}|}$

model = load_model()

$inputs = [Input1, Input2, Input3]$

y' = model.predict($inputs$)

return y'

Algorithm 2 The Algorithm for General Route Planning

Input:

User preference vector v_{u_i},

Travel time budget d,

Acceptable time range ϵ,

Beam size n,

Output: Candidate route set $\mathcal{A}$

1: set $\mathcal{A} = \Phi$,

2: set the temp route set $\mathcal{T} = \Phi$,

3: insert initial route $\{u_i\}$ into $\mathcal{T}$,

4: **repeat**

5: **for** $T_o \in \mathcal{T}$ **do**

6: $y' = Next\_Prediction(T_o, v_{u_i})$;

7: $S_{nextpois} \leftarrow$ get the top-n POIs from y';

8: **for** poi in $S_{nextpois}$ **do**

9: $T_n = T_o$.append(poi);

10: $p(T_n) = p(T_o) \times y'_{poi}$;

11: $t(T_n) = t(T_o) + t(poi) + travel\ time$;

12: **if** $t(T_n) > d + \epsilon$ **then**

13: continue;

14: **end if**

15: **if** $d - \epsilon \le t(T_n) \le d + \epsilon$ **then**

16: insert T_n into $\mathcal{A}$;

17: **end if**

18: **if** $t(T_n) < d + \epsilon$ **then**

19: insert T_n into $\mathcal{T}$;

20: **end if**

21: **end for**

22: remove T_o from $\mathcal{T}$

23: **end for**

24: **until** $\mathcal{T} = \Phi$

25: return $\mathcal{A}$

search, which explores a candidate route by expanding the most promising POIs in a limited set. Beam search algorithm only keeps a pre-defined number of best partial solutions as the candidates. Algorithm 2 illustrates the process of route generation for the route planning task.

The inputs of the general route planning algorithm include target user preference vector v_{u_i}, travel time budget d, acceptable time range ϵ, and beam size n. The output is a set of generated candidate routes along with their probabilities. Here we define the function $p(T_n)$ returns probability score of route T_n, and $t(T_n)$ returns the time cost of T_n.

After the initialization of the candidate route set $\mathcal{A}$ and the temporal route set $\mathcal{T}$, we insert the initial temp route (u_i) into $\mathcal{T}$. In each iteration from step 4 to 24 in Algorithm 2, for each old temp route T_o in $\mathcal{T}$, we utilize the learned model to estimate the probabilities of its next positions as shown in step 7. For each poi in top n highest probability set $S_{nextpois}$, we append it into T_o to form a new temporal route T_n. Then the probability score and time cost of the new route T_n are calculated in step 10 and 11. We utilize the pruning technique to incorporate user's constraints into the generation process. If the time cost exceeds the budget, the new route T_n is discarded. If its time cost satisfies the budget, T_n is considered as an appropriate candidate route and is appended into set $\mathcal{A}$. If the time cost is lower than the upper limit, we insert the new route T_n into $\mathcal{T}$ to further process it. After the update procedure, in step 22 we remove the old route T_o from T. This iteration will be repeated until there are no routes in $\mathcal{T}$. Finally $\mathcal{A}$ contains the candidate routes which satisfy the user's constraints with the top-k highest visiting probabilities. Finally, the candidate routes with the highest probabilities in $\mathcal{A}$ are selected as the final results.

3) Must-Visit Planning: For the task of must-visit planning, the reasoning process is the same with Algorithm 2 but needs two major modifications. Supposing that POI set $P_{must} = \{p_1, p_2, \cdots\}$ consists of all the POIs the user want to visit. Firstly at step 12 in Algorithm 2, except the top n probable POIs, all the POIs in P_{must} are also appended into the set $S_{nextpois}$. This strategy ensures that the must-visit POIs are included in the candidate routes. Second, in the pruning step, if a POI p_i in P_n has been included in a old route T_o, it will not be added to T_o to form a new route T_n.

V. Experiments

In this section, we conduct experiments on six public datasets to evaluate the proposed method. First, the datasets and baselines used in this work are introduced. Then for the three travel planning tasks, i.e., next-point recommendation, general route planning, and must-visit planning, we present experimental results on the datasets and make a thorough analysis.

A. Experiment Setup

1) Datasets: In this work, we introduce the User-POI Visits Dataset on Flickr [57],[1] which extracts photographs of eight cities with geo-tags from the dataset of YFCC100M [58]. These geo-tagged photos are mapped to specific POIs locations, which comprise many users and their visit records to different POIs. We filter the successively visited POIs, of which visit-time intervals are not more than eight hours, as travel routes [58]. In the experiments, Six cities, including Toronto, Vienna, Edinburgh, Glasgow, Budapest, and Osaka, are used as the tourism data. The statistics of the datasets are shown in Table III.

In the datasets, each route is associated with a user and consists of a series of ordered POIs. We represent each POI in the routes with the combination of its ID, category (e.g., park,

[1] https://sites.google.com/site/limkwanhui/datacode#ijcai15

TABLE III
THE STATISTICS OF THE DATASETS

City	POI Photos	Users	POIs	Travel Sequences	Categories
Toronto	39,419	1,395	30	6,057	6
Vienna	34,515	1,155	29	3,193	8
Edinburgh	33,944	1,454	29	5,028	6
Glasgow	11,434	601	28	2,227	7
Budapest	18,513	935	39	2,361	6
Osaka	7,747	450	28	1,115	4

religion, culture, shopping), travel time cost, popularity, and geographical coordinates. Each user is represented with the categories of the visited POIs. In such a way, we can obtain the POI attribute vector, user preference vector, and historical travel routes as the input to our model.

2) Compared Methods: The proposed MDTRP approach is compared with five baseline methods as follows:

- **Multinomial** [16]: This method predicts the next position based on the popularity of each POI, which is calculated by utilizing the multinomial probability distribution over POIs. But it only considers the POI popularity and ignores the user's preference and current location.
- **Markov Model** [59]: For Markov models, the next POI is predicted only depend on the user's current position. Given the current POI, this model will recommend the most frequently visited POI. However, it does not take the preference of the user into account.
- **Topic Model** [11]: This model uses Latent Dirichlet Allocation (LDA) to predict the next POI based on the preference of users. However, the user's current position and previous travel route are not considered.
- **RNN** [60]: Comparing with the proposed multi-sourced model, this model does not consider the user preference and only takes travel route patterns into account in the LSTM.
- **DTRP** [28]: The preliminary version of the proposed method. It does not employ heterogeneous network embedding for feature learning and uses average pooling instead of points attention network for route pattern representation learning.

3) Parameter Setup: In the feature extraction stage, the neighborhood size of network embedding is set to be 5, the negative sample size is set to be 3, and the node feature size is embedded with the dimension of 128. In the model learning stage, the number of hidden neurons in LSTM is set to be 256, the MLP is set to be a two-layer fully connected network with the structure of $512 - num\_class$ through $tanh$ and $softmax$ activation respectively. In the route generation stage, the size of the greedy beam search is set to be 3, and the acceptable time range is set to $\epsilon = 0.5h$.

B. Next-Point Recommendation

For the task of next-point recommendation, the next position is predicted based on the given user and previous POI patterns in a route. The travel routes in each city are randomly divided into 80% and 20% as training and testing sets, respectively. We evaluate the performance of the proposed model and compared methods with the metrics as follows:

- **Accuracy@k**: Accuracy@k computes whether the next real POI is included in the top-k recommended POIs. Assume $\mathcal{R}(T)$ is the top k recommended POIs of the input pattern T, v is the real POI, and N is the number of test data, accuracy@k is defined as: $accuracy@k = \frac{1}{N}\sum_{T=1}^{N} |v \cap \mathcal{R}(T)|$.
- **MAP**: Mean average precision (MAP) [60] is a standard metric for ranking tasks, which can evaluate the quality of the whole ranked lists. Let $I(T)$ be the location of the real next point in the ranked list of the route pattern T, MAP is defined as: $MAP = \frac{1}{N}\sum_{T=1}^{N} \frac{1}{I(T)}$

Table IV presents the results of different models for next-point recommendation. One can see that deep models (the last three methods) largely surpass shallow methods, which validates that deep models excel at modeling sequential date for prediction. By introducing the auxiliary information of user preference and POI features, DTRP improves the performance of RNN by about an average 3% in six datasets. This result proves the effectiveness of our preliminary model to fuse the data from multiple sources for next-point prediction. Comparing the proposed MDTRP with DTRP, one can see that MDTRP improves about 2% on all evaluation metrics. It demonstrates that using heterogeneous network embedding methods to learn the features and building points attention network is effective in training the model for more accurate prediction. Overall, it can be seen that MDTRP can effectively integrate POI attribute, user preference, and historical travel routes and outperforms compared methods in the task of next-point recommendation.

C. General Route Planning

The task of general route planning aims to generate a complete route for tourism. To this end, we first train the models of different methods to give predictions for next-points. Then the beam search algorithm (Section IV.D) is used to generate the next POI iteratively to form a complete travel route for each method. The following evaluating metrics are selected to measure the quality of generated routes.

- **Route Interest**: The route interest of a generated route T to a user u is defined as: $Int_u(T) = \sum_{p \in T} Int_u(p)$, where $Int_u(p) = \frac{n(u,\{p.cat\})}{n(u)}$ denotes the user's preference on the category of POI p. $n(u)$ is the number of POIs visited by the user u.
- **Route Popularity**: The number of times the POI has been visited. On this basis, we define the route popularity as total popularities of all the POIs in a route.
- **Edit Distance (ED)**: The edit operations from the generated route to the real route in the experiments.

The results of general route planning are shown in Table V. One can see that on the metric of route popularity, Multinomial model shows the best performance. The reason is that Multinomial model always selects the most popular POIs, thus outperforms other methods. On the other metrics, Markov model generates most frequently visited POIs after the

TABLE IV
THE EXPERIMENTAL RESULTS ON THE TASK OF NEXT-POINT RECOMMENDATION

Toronto

Algorithm	Acc@1	Acc@3	MAP
Multinomial	0.1119	0.2631	0.2653
Topic model	0.1121	0.2721	0.2626
Markov model	0.1457	0.3182	0.2994
RNN	0.3168	0.6299	0.5134
DTRP	0.3707	0.6505	0.5425
MDTRP	**0.3888**	**0.6737**	**0.5650**

Vienna

Algorithm	Acc@1	Acc@3	MAP
Multinomial	0.1533	0.2603	0.2824
Topic model	0.1573	0.2932	0.2842
Markov model	0.1778	0.3488	0.3354
RNN	0.3041	0.5843	0.4892
DTRP	0.3359	0.6121	0.5162
MDTRP	**0.3564**	**0.6420**	**0.5366**

Edinburgh

Algorithm	Acc@1	Acc@3	MAP
Multinomial	0.1251	0.2728	0.2764
Topic model	0.1330	0.2877	0.2891
Markov model	0.1652	0.3470	0.3211
RNN	0.2937	0.5854	0.4705
DTRP	0.3332	0.6067	0.5129
MDTRP	**0.3545**	**0.6392**	**0.5327**

Glasgow

Algorithm	Acc@1	Acc@3	MAP
Multinomial	0.0972	0.2743	0.2677
Topic model	0.1068	0.2937	0.2799
Markov model	0.1166	0.3267	0.2951
RNN	0.3462	0.6372	0.5155
DTRP	0.3739	0.6519	0.5519
MDTRP	**0.3905**	**0.6838**	**0.5704**

Budapest

Algorithm	Acc@1	Acc@3	MAP
Multinomial	0.0978	0.2513	0.2520
Topic model	0.1159	0.2631	0.2743
Markov model	0.1482	0.3294	0.2979
RNN	0.2705	0.5700	0.4479
DTRP	0.3153	0.5903	0.4936
MDTRP	**0.3292**	**0.6143**	**0.5058**

Osaka

Algorithm	Acc@1	Acc@3	MAP
Multinomial	0.1120	0.2902	0.2901
Topic model	0.1262	0.3093	0.3010
Markov model	0.1399	0.3385	0.3122
RNN	0.3689	0.6634	0.5425
DTRP	0.3880	0.6676	0.5752
MDTRP	**0.4185**	**0.6943**	**0.5814**

TABLE V
THE EXPERIMENTAL RESULTS ON THE TASK OF GENERAL ROUTE PLANNING

Toronto

Algorithm	Route interest	Route popularity	Edit distance
Multinomial	0.2528	**0.1326**	1.1870
Topic model	0.2563	0.0835	1.0785
Markov model	0.2665	0.0871	1.0249
RNN	0.4419	0.0911	0.6671
DTRP	0.4767	0.0924	0.6342
MDTRP	**0.4979**	0.0982	**0.6111**

Vienna

Algorithm	Route interest	Route popularity	Edit distance
Multinomial	0.4695	**0.1409**	1.4112
Topic model	0.4891	0.0934	1.1993
Markov model	0.4927	0.1117	1.1855
RNN	0.6978	0.1009	1.1203
DTRP	0.7232	0.1131	1.0888
MDTRP	**0.7416**	0.1053	**1.0480**

Edinburgh

Algorithm	Route interest	Route popularity	Edit distance
Multinomial	0.3362	**0.1351**	1.4114
Topic model	0.3961	0.1051	1.3862
Markov model	0.3958	0.1138	1.2935
RNN	0.6032	0.1108	1.1301
DTRP	0.6280	0.1216	1.0894
MDTRP	**0.6460**	0.1189	**1.0311**

Glasgow

Algorithm	Route interest	Route popularity	Edit distance
Multinomial	0.2746	**0.1235**	1.3353
Topic model	0.3008	0.0944	1.1141
Markov model	0.3024	0.1103	0.9898
RNN	0.4715	0.0896	0.7250
DTRP	0.4945	0.1026	0.7017
MDTRP	**0.5133**	0.1119	**0.6711**

Budapest

Algorithm	Route interest	Route popularity	Edit distance
Multinomial	0.3193	**0.1130**	1.3945
Topic model	0.3781	0.0874	1.3676
Markov model	0.3778	0.0990	1.2787
RNN	0.5851	0.0946	1.1154
DTRP	0.6061	0.1018	1.0710
MDTRP	**0.6253**	0.1035	**1.0667**

Osaka

Algorithm	Route interest	Route popularity	Edit distance
Multinomial	0.2585	**0.1043**	1.3148
Topic model	0.2796	0.0789	1.1007
Markov model	0.2860	0.0933	0.9726
RNN	0.4550	0.0747	0.7039
DTRP	0.4768	0.0836	0.6876
MDTRP	**0.4949**	0.0859	**0.6618**

current position, which is the choice of most travelers. Hence the recommended routes are better than Multinomial and Topic models. By integrating POI attributes, user preference, and historical travel routes, the proposed approach MDTRP surpasses shallow baselines by over 20% and deep baselines by over 2% on the metric of route interest. In addition, MDTRP also improves a lot than compared methods on the metric of edit distance. Overall, the proposed MDTRP can better satisfy the user's preference and generate higher quality routes with lower edit distance.

D. Must-Visit Planning

On occasion, users hope to plan their travel routes with extra constraints of including some must-visit POIs. Therefore, we measure the quality of must-visit travel routes to evaluate the proposed approach on this task. We employ the metrics of route precision, recall, and F1-score for evaluation as follows:

- **Precision**: The proportion of POIs in the generated route that are also in the actual route, which is defined as $P(T) = \frac{|P_g \cap P_r|}{|P_g|}$.
- **Recall**: The proportion of POIs in the actual P_r that are also in the generated route P_g, which is defined as $R(T) = \frac{|P_g \cap P_r|}{|P_r|}$.
- **F1-score**: The harmonic mean of the precision and recall of the generated route T, defined as $F_1 = \frac{2 \times P(T) \times R(T)}{P(T)+R(T)}$.

The results of must-visit planning are shown in Table VI. From the table, we can see that MDTRP performs solidly better than compared methods on metrics of precision and recall. Since F1 score provides the balance score between the precision and recall, the proposed approach also presents the

TABLE VI
THE EXPERIMENTAL RESULTS ON THE TASK OF MUST-VISIT ROUTE PLANNING

Toronto

Algorithm	Pre.	Rec.	F1
Multinomial	0.4525	0.5171	0.4827
Topic model	0.4905	0.5720	0.5281
Markov model	0.5486	0.6267	0.5850
RNN	0.5945	0.6949	0.6408
DTRP	0.6275	0.7134	0.6677
MDTRP	**0.6375**	**0.7250**	**0.6784**

Vienna

Algorithm	Pre.	Rec.	F1
Multinomial	0.4344	0.4563	0.4451
Topic model	0.4903	0.5549	0.5206
Markov model	0.5197	0.5495	0.5342
RNN	0.6098	0.6928	0.6487
DTRP	0.6373	0.7071	0.6704
MDTRP	**0.6577**	**0.7181**	**0.6865**

Edinburgh

Algorithm	Pre.	Rec.	F1
Multinomial	0.4037	0.4778	0.4377
Topic model	0.4883	0.5566	0.5202
Markov model	0.4970	0.5314	0.5136
RNN	0.5808	0.6621	0.6188
DTRP	0.6078	0.6824	0.6429
MDTRP	**0.6242**	**0.7025**	**0.6610**

Glasgow

Algorithm	Pre.	Rec.	F1
Multinomial	0.4927	0.5358	0.5134
Topic model	0.5062	0.5891	0.5445
Markov model	0.5362	0.5791	0.5568
RNN	0.6755	0.7532	0.7122
DTRP	0.6997	0.7765	0.7361
MDTRP	**0.7192**	**0.7923**	**0.7540**

Budapest

Algorithm	Pre.	Rec.	F1
Multinomial	0.3912	0.4611	0.4233
Topic model	0.4715	0.5368	0.5020
Markov model	0.4796	0.5128	0.4957
RNN	0.5599	0.6458	0.5998
DTRP	0.6147	0.6975	0.6535
MDTRP	**0.6246**	**0.7275**	**0.6721**

Osaka

Algorithm	Pre.	Rec.	F1
Multinomial	0.4846	0.5273	0.5050
Topic model	0.4970	0.5792	0.5349
Markov model	0.5299	0.5741	0.5511
RNN	0.6698	0.7411	0.7036
DTRP	0.6279	0.6976	0.6609
MDTRP	**0.6435**	**0.7169**	**0.6782**

best F1-score on the six datasets. It confirms that MDTRP is effective to plan the travel routes with the constraint of must-visit POIs.

VI. CONCLUSIONS AND FUTURE WORK

In this work, we propose a flexible multi-task deep travel route planning framework named MDTRP to integrate tourism data from multiple sources and fit various types of travel demands. We treat the problem of route planning as a POI sequence prediction process, which can be well addressed by deep neural networks. In the feature extraction stage, we employ a heterogeneous network embedding method to learn the features of users and POIs, and convert each travel route to a set of POI patterns. In the model learning stage, a novel deep attention model is introduced to learn the probability of visiting the next POIs given the input patterns. In the route generation stage, we employ a beam search algorithm to flexibly generate feasible route candidates for multiple tasks. Experimental results on the task of next-point recommendation, general route planning, and must-visit planning over six datasets validate the effectiveness of MDTRP.

Different from existing methods of travel route planning, the proposed MDTRP can integrate POI attributes, user preference, and historical route data into a holistic framework. In addition, due to the generality and flexibility of the proposed model, it can be applied to the three most common route planning tasks. As for the future work, we want to exploit the heterogeneous network relation between users and POIs with deep graph convolutional networks to represent node features more effectively. Furthermore, the network structures can be explored to merge the stage of feature extraction and model learning into an end-to-end framework.

REFERENCES

[1] S. Edelkamp, S. Jabbar, and T. Willhalm, "Geometric travel planning," *IEEE Trans. Intell. Transp. Syst.*, vol. 6, no. 1, pp. 5–16, Mar. 2005.

[2] I. R. Brilhante, J. A. Macedo, F. M. Nardini, R. Perego, and C. Renso, "On planning sightseeing tours with TripBuilder," *Inf. Process. Manage.*, vol. 51, no. 2, pp. 1–15, Mar. 2015.

[3] C. Chen, D. Zhang, B. Guo, X. Ma, G. Pan, and Z. Wu, "TripPlanner: Personalized trip planning leveraging heterogeneous crowdsourced digital footprints," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 3, pp. 1259–1273, Jun. 2015.

[4] P. Vansteenwegen, W. Souffriau, G. V. Berghe, and D. V. Oudheusden, "The city trip planner: An expert system for tourists," *Expert Syst. Appl.*, vol. 38, no. 6, pp. 6540–6546, Jun. 2011.

[5] B. A. Beirigo and A. G. dos Santos, "A parallel heuristic for the travel planning problem," in *Proc. 15th Int. Conf. Intell. Syst. Design Appl. (ISDA)*, Marrakech, Morocco, Dec. 2015, pp. 283–288.

[6] A. Gunawan, H. C. Lau, and P. Vansteenwegen, "Orienteering problem: A survey of recent variants, solution approaches and applications," *Eur. J. Oper. Res.*, vol. 255, no. 2, pp. 315–332, Dec. 2016.

[7] S. Rani, K. N. Kholidah, and S. N. Huda, "A development of travel itinerary planning application using traveling salesman problem and K-means clustering approach," in *Proc. 7th Int. Conf. Softw. Comput. Appl. (ICSCA)*, Kuantan, Malaysia, 2018, pp. 327–331.

[8] Y. Arase, X. Xie, T. Hara, and S. Nishio, "Mining people's trips from large scale geo-tagged photos," in *Proc. Int. Conf. Multimedia (MM)*, Firenze, Italy, 2010, pp. 133–142.

[9] Y. Zheng and X. Xie, "Learning travel recommendations from user-generated GPS traces," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 1, pp. 2:1–2:29, Jan. 2011.

[10] Y.-T. Zheng, Z.-J. Zha, and T.-S. Chua, "Mining travel patterns from geotagged photos," *ACM Trans. Intell. Syst. Technol.*, vol. 3, no. 3, pp. 56:1–56:18. May 2012.

[11] T. Kurashima, T. Iwata, G. Irie, and K. Fujimura, "Travel route recommendation using geotagged photos," *Knowl. Inf. Syst.*, vol. 37, no. 1, pp. 37–60, Oct. 2013.

[12] Z. Xu, L. Chen, and G. Chen, "Topic based context-aware travel recommendation method exploiting geotagged photos," *Neurocomputing*, vol. 155, pp. 99–107, May 2015.

[13] L. Chen, Z. Wu, J. Cao, G. Zhu, and Y. Ge, "Travel recommendation via fusing multi-auxiliary information into matrix factorization," *ACM Trans. Intell. Syst. Technol.*, vol. 11, no. 2, pp. 1–24, Mar. 2020.

[14] S. Jiang, X. Qian, T. Mei, and Y. Fu, "Personalized travel sequence recommendation on multi-source big social media," *IEEE Trans. Big Data*, vol. 2, no. 1, pp. 43–56, Mar. 2016.

[15] I. Brilhante, J. A. Macedo, F. M. Nardini, R. Perego, and C. Renso, "Where shall we go today?: Planning touristic tours with tripbuilder," in *Proc. 22nd ACM Int. Conf. Conf. Inf. Knowl. Manage. (CIKM)*, San Francisco, CA, USA, 2013, pp. 757–762.

[16] K. H. Lim, "Recommending tours and places-of-interest based on user interests from geo-tagged photos," in *Proc. ACM SIGMOD PhD Symp.-SIGMOD PhD Symp.*, Melbourne, VIC, Australia, Jun. 2015, pp. 33–38.

[17] Z. Yu, H. Xu, Z. Yang, and B. Guo, "Personalized travel package with multi-point-of-interest recommendation based on crowdsourced user footprints," *IEEE Trans. Human-Mach. Syst.*, vol. 46, no. 1, pp. 151–158, Feb. 2016.

[18] Y.-T. Wen, J. Yeo, W.-C. Peng, and S.-W. Hwang, "Efficient keyword-aware representative travel route recommendation," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 8, pp. 1639–1652, Aug. 2017.

[19] C. Zhang, H. Liang, K. Wang, and J. Sun, "Personalized trip recommendation with POI availability and uncertain traveling time," in *Proc. 24th ACM Int. Conf. Inf. Knowl. Manage. (CIKM)*, Melbourne, VIC, Australia, Oct. 2015, pp. 911–920.

[20] M. Chen, X. Yu, and Y. Liu, "Mining moving patterns for predicting next location," *Inf. Syst.*, vol. 54, pp. 156–168, Dec. 2015.

[21] J. He, X. Li, L. Liao, D. Song, and W. K. Cheung, "Inferring a personalized next point-of-interest recommendation model with latent behavior patterns," in *Proc. 13th AAAI Conf. Artif. Intell.* Phoenix, AZ, USA: AAAI Press, Feb. 2016, pp. 137–143.

[22] J. He, X. Li, and L. Liao, "Category-aware next point-of-interest recommendation via listwise Bayesian personalized ranking," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Melbourne, VIC, Australia, Aug. 2017, pp. 1837–1843.

[23] R. Li, Y. Shen, and Y. Zhu, "Next point-of-interest recommendation with temporal and multi-level context attention," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*. Singapore: IEEE Computer Society, Nov. 2018, pp. 1110–1115.

[24] H.-P. Hsieh, C.-T. Li, and S.-D. Lin, "Exploiting large-scale check-in data to recommend time-sensitive routes," in *Proc. ACM SIGKDD Int. Workshop Urban Comput. (UrbComp)*, Beijing, China, Aug. 2012, pp. 55–62.

[25] D. Ashbrook and T. Starner, "Using GPS to learn significant locations and predict movement across multiple users," *Pers. Ubiquitous Comput.*, vol. 7, no. 5, pp. 275–286, 2003.

[26] T. Hofmann, "Collaborative filtering via Gaussian probabilistic latent semantic analysis," in *Proc. 26th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (SIGIR)*, Toronto, ON, Canada, 2003, pp. 259–266.

[27] T. Iwata, S. Watanabe, T. Yamada, and N. Ueda, "Topic tracking model for analyzing consumer purchase behavior," in *Proc. 21st Int. Joint Conf. Artif. Intell.*, Pasadena, CA, USA, Jul. 2009, pp. 1427–1432.

[28] J. Xu *et al.*, "DTRP: A flexible deep framework for travel route planning," in *Proc. 18th Int. Conf. Web Inf. Syst. Eng.*, in Lecture Notes in Computer Science, vol. 10569. Puschino, Russia: Springer, Oct. 2017, pp. 359–375.

[29] M. V. Moreno *et al.*, "Applicability of big data techniques to smart cities deployments," *IEEE Trans. Ind. Informat.*, vol. 13, no. 2, pp. 800–809, Apr. 2017.

[30] L. Zhang, J. Wu, S. Mumtaz, J. Li, H. Gacanin, and J. J. P. C. Rodrigues, "Edge-to-edge cooperative artificial intelligence in smart cities with on-demand learning offloading," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Waikoloa, HI, USA, Dec. 2019, pp. 1–6.

[31] Z. Wu, L. Kong, G. Chen, M. K. Khan, S. Mumtaz, and J. J. P. C. Rodrigues, "CrowdSwitch: Crowdsensing based switch between multiple cellular operators in subways," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Waikoloa, HI, USA, Dec. 2019, pp. 1–6.

[32] L. Tang, Y. He, Z. Zhou, Y. Ren, S. Mumtaz, and J. Rodriguez, "A distance-sensitive distributed repulsive sleeping approach for dependable coverage in heterogeneous cellular networks," *Trans. Emerg. Telecommun. Technol.*, vol. 30, no. 11, p. e3784, Nov. 2019.

[33] R. Matai, S, P. Singh, and M. L. Mittal, "Traveling salesman problem: An overview of applications, formulations, and solution approaches," in *Traveling Salesman Problem, Theory and Applications*. 2010, p. 1.

[34] J. Faigl, R. Penicka, and G. Best, "Self-organizing map-based solution for the orienteering problem with neighborhoods," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Budapest, Hungary, Oct. 2016, pp. 1315–1321.

[35] B. Rodríguez, J. Molina, F. Pérez, and R. Caballero, "Interactive design of personalised tourism routes," *Tourism Manage.*, vol. 33, no. 4, pp. 926–940, Aug. 2012.

[36] Y. Dong, N. V. Chawla, and A. Swami, "Metapath2vec: Scalable representation learning for heterogeneous networks," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Halifax, NS, Canada, Aug. 2017, pp. 135–144.

[37] C. Shi, B. Hu, W. X. Zhao, and P. S. Yu, "Heterogeneous information network embedding for recommendation," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 2, pp. 357–370, Feb. 2019.

[38] B. Hu, C. Shi, W. X. Zhao, and P. S. Yu, "Leveraging meta-path based context for top-n recommendation with a neural co-attention model," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, London, U.K., Jul. 2018, pp. 1531–1540.

[39] H. Chen, H. Yin, W. Wang, H. Wang, Q. V. H. Nguyen, and X. Li, "PME: Projected metric embedding on heterogeneous networks for link prediction," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, London, U.K., Jul. 2018, pp. 1177–1186.

[40] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. Adv. Neural Inf. Process. Syst.*, Lake Tahoe, NV, USA, Dec. 2013, pp. 3111–3119.

[41] A. Grover and J. Leskovec, "Node2vec: Scalable feature learning for networks," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, San Francisco, CA, USA, Aug. 2016, pp. 855–864.

[42] B. Perozzi, R. Al-Rfou, and S. Skiena, "DeepWalk: Online learning of social representations," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, New York, NY, USA, 2014, pp. 701–710.

[43] X. Rong, "Word2vec parameter learning explained," 2014, *arXiv:1411.2738*. [Online]. Available: http://arxiv.org/abs/1411.2738

[44] M. Sundermeyer, R. Schlüter, and H. Ney, "LSTM neural networks for language modeling," in *Proc. 13th Annu. Conf. Int. Speech Commun. Assoc. (INTERSPEECH)*. Portland, OR, USA: ISCA, Sep. 2012, pp. 194–197.

[45] T. Mikolov, M. Karafiá, L. Burget, J. Cernocký, and S. Khudanpur, "Recurrent neural network based language model," in *Proc. 11th Annu. Conf. Int. Speech Commun. Assoc. (INTERSPEECH)*. Makuhari, Japan: ISCA, Sep. 2010, pp. 1045–1048.

[46] A. Graves, "Generating sequences with recurrent neural networks," 2013, *arXiv:1308.0850*. [Online]. Available: http://arxiv.org/abs/1308.0850

[47] K. Xu *et al.*, "Show, attend and tell: Neural image caption generation with visual attention," in *Proc. 32nd Int. Conf. Mach. Learn.*, Lille, France, vol. 37, Jul. 2015, pp. 2048–2057.

[48] Z. Yang, X. He, J. Gao, L. Deng, and A. Smola, "Stacked attention networks for image question answering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*. Las Vegas, NV, USA: IEEE Computer Society, Jun. 2016, pp. 21–29.

[49] A. Vaswani *et al.*, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, Long Beach, CA, USA, Dec. 2017, pp. 5998–6008.

[50] F. Huang, X. Zhang, Z. Zhao, and Z. Li, "Bi-directional spatial-semantic attention networks for image-text matching," *IEEE Trans. Image Process.*, vol. 28, no. 4, pp. 2008–2020, Apr. 2019.

[51] Y. Zhang, J. Wu, Z. Cai, and P. S. Yu, "Multi-view multi-label learning with sparse feature selection for image annotation," *IEEE Trans. Multimedia*, early access, Jan. 15, 2020, doi: 10.1109/TMM.2020.2966887.

[52] F. Huang, X. Zhang, J. Xu, Z. Zhao, and Z. Li, "Multimodal learning of social image representation by exploiting social relations," *IEEE Trans. Cybern.*, early access, Mar. 6, 2019, doi: 10.1109/TCYB.2019.2896100.

[53] V. Vanhoucke, A. Senior, and M. Z. Mao, "Improving the speed of neural networks on cpus," in *Proc. Deep Learn. Unsupervised Feature Learn. Workshop (NIPS)*, Princeton, NJ, USA: Citeseer, 2011, pp. 1–8.

[54] Y. Liu, C. Sun, L. Lin, and X. Wang, "Learning natural language inference using bidirectional LSTM model and inner-attention," 2016, *arXiv:1605.09090*. [Online]. Available: http://arxiv.org/abs/1605.09090

[55] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR)*, San Diego, CA, USA, May 2015, pp. 1–15.

[56] C. Tillmann and H. Ney, "Word reordering and a dynamic programming beam search algorithm for statistical machine translation," *Comput. Linguistics*, vol. 29, no. 1, pp. 97–133, Mar. 2003.

[57] K. H. Lim, J. Chan, C. Leckie, and S. Karunasekera, "Personalized tour recommendation based on user interests and points of interest visit durations," in *Proc. 24th Int. Joint Conf. Artif. Intell. (IJCAI)*. Buenos Aires, Argentina: AAAI Press, Jul. 2015, pp. 1778–1784.

[58] B. Thomee *et al.*, "YFCC100M: The new data in multimedia research," 2015, *arXiv:1503.01817*. [Online]. Available: http://arxiv.org/abs/1503.01817

[59] H. Gao, J. Tang, and H. Liu, "Exploring social-historical ties on location-based social networks," in *Proc. 6th Int. Conf. Weblogs Social Media*. Dublin, Ireland: AAAI Press, Jun. 2012, pp. 1–8.

[60] Q. Liu, S. Wu, L. Wang, and T. Tan, "Predicting the next location: A recurrent model with spatial and temporal contexts," in *Proc. 13th AAAI Conf. Artif. Intell.* Phoenix, AZ, USA: AAAI Press, Feb. 2016, pp. 194–200.

Feiran Huang received the B.Sc. degree from Central South University, Changsha, China, in 2011, and the Ph.D. degree in computer software and theory from the School of Computer Science and Engineering, Beihang University, Beijing, China, in 2019. He is currently a Lecturer with the School of Information Science and Technology and the College of Cyber Security, Jinan University, Guangzhou, China. He has published over ten articles, such as TIP, TCYB, KBS, ACM MM, CIKM, and ICMR. His research interests include social media analysis and multimodal learning.

Jie Xu received the M.Sc. degree from the School of Computer Science, Beijing University of Posts and Telecommunications, Beijing, China, in 2014. She is currently pursuing the Ph.D. degree with the State Key Laboratory of Software Development Environment, School of Computer Science and Engineering, Beihang Univeristy, Beijing. Her research interests include data mining and social media analysis.

Jian Weng (Member, IEEE) received the B.S. and M.S. degrees in computer science from the South China University of Technology, Guangzhou, China, in 2001 and 2004, respectively, and the Ph.D. degree in computer science from Shanghai Jiao Tong University, Shanghai, China, in 2008.

He is currently a Professor and the Executive Dean with the College of Information Science and Technology, Jinan University, Guangzhou. He has authored/coauthored 100 papers in international conferences and journals, such as CRYPTO, EUROCRYPT, ASIACRYPT, TCC, PKC, CT-RSA, the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, the IEEE TRANSACTIONS ON INFORMATION THEORY, and the IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING. His research areas include public key cryptography, cloud security, and blockchain. He was a recipient of the Young Scientists Fund of the National Natural Science Foundation of China in 2018 and the Cryptography Innovation Award from the Chinese Association for Cryptologic Research (CACR) in 2015. He has served as the General Co-Chair for SecureComm 2016, the TPC Co-Chair for RFIDsec13 Asia, and ISPEC 2011, and a Program Committee Member for more than 40 international cryptography and information security conferences. He also serves as an Associate Editor for the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY.