Dynamic Personalized POI Sequence Recommendation with Fine-Grained Contexts

JING CHEN and WENJUN JIANG, Hunan University, China
JIE WU, Temple University, USA
KENLI LI, Hunan University, China
KEQIN LI, State University of New York, USA

The Point Of Interest (POI) sequence recommendation is the key task in itinerary and travel route planning. Existing works usually consider the temporal and spatial factors in travel planning. However, the external environment, such as the weather, is usually overlooked. In fact, the weather is an important factor because it can affect a user's check-in behaviors. Furthermore, most of the existing research is based on a static environment for POI sequence recommendation. While the external environment (e.g., the weather) may change during travel, it is difficult for existing works to adjust the POI sequence in time. What's more, people usually prefer the attractive routes when traveling. To address these issues, we first conduct comprehensive data analysis on two real-world check-in datasets to study the effects of weather and time, as well as the features of the POI sequence. Based on this, we propose a model of Dynamic Personalized POI Sequence Recommendation with fine-grained contexts (*DPSR* for short). It extracts user interest and POI popularity with fine-grained contexts and captures the attractiveness of the POI sequence. Next, we apply the Monte Carlo Tree Search model (MCTS for short) to simulate the process of recommending POI sequence in the dynamic environment, i.e., the weather and time change after visiting a POI. What's more, we consider different speeds to reflect the fact that people may take different transportation to transfer between POIs. To validate the efficacy of *DPSR*, we conduct extensive experiments. The results show that our model can improve the accuracy of the recommendation significantly. Furthermore, it can better meet user preferences and enhance experiences.

CCS Concepts: • **Computing methodologies** → **Recommendation**; • **Information systems** → **Travel Route recommendation**;

Additional Key Words and Phrases: Context-aware, dynamic, fine-grained, personalized POI sequence recommendation

ACM Reference format:
Jing Chen, Wenjun Jiang, Jie Wu, Kenli Li, and Keqin Li. 2023. Dynamic Personalized POI Sequence Recommendation with Fine-Grained Contexts. *ACM Trans. Internet Technol.* 23, 2, Article 32 (May 2023), 28 pages.
https://doi.org/10.1145/3583687

This work was supported by National Natural Science Foundation of China Grants no. 62172149 and no. 61632009, the open project of Zhejiang Lab (2019KE0AB02), and the Natural Science Foundation of Hunan Province of China (2021JJ30137).
Authors' addresses: J. Chen, W. Jiang (corresponding author), and K. Li, Hunan University, College of Information Science and Electronic Engineering, South Lushan Road N. 2, Changsha, Hunan 410082, China; emails: {jessicachan, jiangwenjun, lkl}@hnu.edu.cn; J. Wu, Temple University, Department of Computer and Information Sciences Temple University SERC 362 1925 N. 12th Street Philadelphia, PA 19122; email: jiewu@temple.edu; K. Li, Department of Computer Science State University of New York Science Hall 249 1 Hawk Drive New Paltz, New York 12561, USA; email: lik@newpaltz.edu.

1533-5399/2023/05-ART32 $15.00
https://doi.org/10.1145/3583687

1 INTRODUCTION

In recent years, **Location-Based Social Networking** (**LBSN**) platforms, such as Foursquare [34] and Yelp [22], are becoming more and more popular. Personalized **Point Of Interest** (**POI**) sequence recommendation is one of the core functions in many LBSN applications, e.g., travel route recommendation and itinerary plan. It can recommend interesting travel routes for people in an unfamiliar city, as well as provide an affable itinerary plan for busy people. In fact, it is very challenging to perform effective pathfinding in a large and complex road network [5]. Many researchers consider the temporal and spatial factors in the next POI recommendation task (e.g., [38, 49, 51]). However, the external environment like the weather has rarely been applied to the dynamic POI sequence recommendation tasks. In addition, people will consider more attractive routes when going out [13]. To address the above issues, in this article, we propose a model of Dynamic Personalized POI Sequence Recommendation with fine-grained contexts.

Some researchers are devoted to recommending the POI sequence considering the contexts, e.g., time [10, 11], geographical location [43, 53], POI category [8], and social relationships [40, 42, 46]. A few works handle the itinerary recommendation by avoiding queue time and crowds [25, 27, 44]. However, existing works usually neglect the weather information, e.g., if the recommended POI does not consider the weather during the recommended travel process, it may recommend outdoor POIs for people on a rainy day. In real life, the external environment [16, 50] is one of the most important factors that affect users' check-in behaviors. For example, people tend to visit ice cream shops when the temperature is high; while the bar is more popular at night.

Different features have been studied for POI sequence recommendation. Chen et al. [7] considered the POI ranking and the POI to POI transitions. Baral et al. [2] applied RNN and LSTM to recommend POI sequence incorporating social influence. The LSTM model is also used to learn a user's visiting behavior for the next POI recommendation [52]. What's more, some works consider the geography-aware sequential recommendation based on the self-attention network, to capture long-term sequential dependence [24]. However, they neglect the fine-grained features of the POI sequence and they usually overlook the context information. Additionally, they mainly consider static environments for recommendation [26]. As the environment may change during a tour, the POI sequence should adapt to the dynamic contexts changing. For instance, a tourist has completed part of a recommended POI sequence while it starts to rain. Then the recommendation model should adjust a proper next POI, e.g., to recommend indoor POIs.

It is a complex task to recommend the POI sequence dynamically because (1) it is an NP-hard problem [3] since POI sequence recommendation needs to compute all permutations from many POIs of a city in a dynamic environment; (2) it needs to consider various contextual information, and fine-grained user interest and POI popularity; and (3) the POI sequence recommendation needs to consider more fine-grained features, e.g., the transfer probability from one POI to another.

Our motivations. Keeping the tasks of personalized POI sequence recommendation and the above analysis in mind, our motivations are threefold: (1) to effectively explore the impact of weather and time on user interest and POI popularity; (2) to capture the features of POI sequence for improving the attraction of recommendation; and (3) to simulate the process of POI sequence recommendation in dynamic environments.

We propose a model of **Dynamic Personalized POI Sequence Recommendation** with fine-grained contexts (*DPSR* for short). It exploits the impact of context on user check-in behaviors and considers the context-aware user interest, context-aware POI popularity, and the attractive sub-sequence. The main contributions are as follows:

(1) We identify the problem of *DPSR* with fine-grained contexts, and analyze its NP complexity. We conduct comprehensive data analysis based on two real LBSN check-in datasets,

verifying that weather and time play an important role in the user's check-in behaviors. We also analyze the features of POI sequences, including the sequence length distribution, the transfer probability between POIs, and the distance and time interval distribution between consecutive check-in POIs.

(2) We propose a *DPSR* model with fine-grained contexts. First, we model user interest and POI popularity with fine-grained contexts. Then we propose a POI transfer probability model to capture popular POI sub-sequences. Based on this, the *DPSR* model applies the Monte Carlo Tree Search to simulate the process of POI selection in a dynamically changing environment within the time budgets. It also takes into account the transfer speed of users between different POIs.

(3) We conduct extensive experiments on two real LBSN check-in datasets, for which we crawled the weather information. Besides commonly used metrics for recommendation, we also propose three metrics ($pair\_Pre$, $pair\_Rec$, $pair\_F1$) to measure the accuracy of the recommended POI sequences. The results show that the *DPSR* model can significantly improve the accuracy of the POI sequence recommendation. Furthermore, the recommended POI sequences can better meet users' interests and have higher popularity.

2 RELATED WORK

We briefly review the literature and point out how they connect with or differ from our work.

2.1 Context-Aware POI Recommendation

Adomavicius et al. [1] first integrated contextual information into the recommendation system and proved contexts can improve the efficiency of recommendation. Yuan et al. [49] constructed a time-aware POI recommendation model and proved that the temporal factor is an important role in POI recommendation. Trattner et al. [40] conducted a lot of data analysis that concluded that different weather features (e.g., temperature, rainfall intensity, humidity, and so on) can affect the POI popularity. However, these existing works only consider the impact of a single contextual factor (e.g., weather or time) on a single POI recommendation, and have not fully applied it to the POI sequence recommendation task. If these contextual factors are to be considered comprehensively in the POI sequence recommendation, it will be a very complicated problem. Shi and Jiang [38] considered the geographical influence from the distance between every pair of locations and developed the concept of local exploration and local trajectories. Pei et al. [33] proposed a value-aware recommendation, and it exploits the reinforcement learning model to maximize the profit in e-commerce systems. Similarly, we try to maximize user satisfaction in the POI sequence recommendation scenarios. Xin et al. [47] developed a novel context-aware recommendation algorithm that seamlessly combines automatic feature interaction modeling of Factorization Machines and Convolutional Neural Networks. Rendle et al. [35] applied the FM model to the context-aware recommendation. Similarly, our context-aware user interest model is mainly related to Chen et al. [47] to refine user preference under the context.

Some researchers focus on investigating where people go and conduct the next POI recommendation [4, 19, 32, 39]. Chang et al. [4] presented a content-aware successive POI recommendation method, which exploited the text content to capture the characteristics of POIs. Sun et al. [39] proposed **Long- and Short-Term Preference Modeling** (**LSTPM**) for the next POI recommendation, in which the geo-dilated RNN and nonlocal network are used to model the short-term and long-term preference, respectively. Huang et al. [19] proposed a **spatio-temporal long and short-term memory** (**ST-LSTM**) network and developed an **attention-based spatiotemporal LSTM** (**ATST-LSTM**) network for the next POI recommendation. Manotumruksa [32] employed a **Deep Recurrent Collaborative Filtering** (**CF**) (**DRCF**) framework with a pairwise ranking function. It exploits the Multi-Layer Perception and RNN architectures to capture user-venue interactions in a CF manner. The above-mentioned studies aim to recommend the next POI based on the successive POIs. Different from that, we consider recommending the POI sequence for a period of time, which can be applied to real-life travel scenarios.

2.2 POI Sequence Recommendation

Travel route recommendation systems are popular in both academia and industry [23]. Among them, the POI sequence recommendation is the key task of the itinerary plan and travel route recommendation. Choudhury et al. [9] were one of the earlier researchers and regarded the travel route recommendation as the Orientation Problem. They recommended the user's most popular travel route when determining start and end locations. Chen et al. [7] propose a probability model to learn the POIs ranking and transition. Some researchers combined various scenarios to recommend travel routes [6, 25, 44], such as recommending comfortable travel routes by sensing traffic conditions [6] and considering queuing time [25]. Hu et al. [17] exploited the user interest to plan a personalized travel route for the user. Chen et al. [8] developed novel search space pruning techniques for route recommendation by effectively reusing past travel behaviors. Although many techniques have been proposed, the context information is usually overlooked by existing works, while in real life, the travel routes are closely related to context. It incites us to deeply study the fine-grained context for improving travel routes recommendation. We mainly consider the weather, time, and user preference to recommend attractive routes for the user.

Xu et al. [48] conducted topic-level travel recommendation by mining the travel histories in different seasons. Debnath et al. [10] presented an a priori-based travel route generation algorithm that considered the users' preferences varying with the time. Lim et al. proposed the personalized itineraries that take into consideration some contexts, such as POI popularity, user interests, queuing times, crowds, and duration time [25, 27, 44]. Baral et al. [2] adopted the capability of RNNs to incorporate various types of context including the temporal, categorical, and spatial information; however, it didn't consider the user's interest. As far as we know, few works pay attention to the impact of weather on users' POI visiting behaviors. Therefore, our work tries to comprehensively consider the context in real-life scenarios, e.g., weather, time, and social influence will be explored to recommend users more satisfactory POI sequences.

Different from existing works, our work tries to (1) combine the weather and temporal influence to refine the fine-grained user preference and POI popularity; (2) capture the features of POI sequence to extract the attractive sub-routes; and (3) simulate the process of POI sequence selection that can cater to user preference and POI attraction in the dynamic context.

3 DATA DESCRIPTION AND ANALYSIS

In this section, we describe the datasets, analyze the impact of weather and time on user check-in behaviors, and explore the POI sequence features.

3.1 Dataset

Since most of the existing POI sequence recommendations do not use the weather information, there is no suitable dataset (e.g., Weeplaces, Foursquare, and so on) that contains weather context. Therefore, we make use of the check-in records from the Weeplaces [29] dataset to crawl the weather data. The datasets are widely applied to the research of POI recommender systems [2, 28, 30]. Furthermore, since we don't change any information in the original data, we believe it will not introduce any bias. A check-in record in the Weeplaces dataset includes the user_id, place_id, the check-in time, the latitude *lat* and longitude *lon*, and the category of a POI. We mainly crawl the weather data of two cities, i.e., New York and San Francisco, through the Dark Sky weather

Table 1. Statistics of the Two Datasets in New York and San Francisco

Weeplaces Dataset				
City	Check-ins	Users	POIs	POI Sequences
New York	273,524	2,723	12,487	111,644
San Francisco	115,251	1,459	4,055	26,406

Table 2. An Example POI Sequence in the Dataset (user_id=2, seq_id=2888)

placeid	datetime	lat	lon	Category	Summary	Temperature	humidity
1,430	01-31T12:28:16	40.32	−73.98	Food:Asian	Cloudy	18.66	0.46
6,981	01-30T13:50:49	40.74	−73.98	Food:Coffee Shop	Rainy	17.66	0.48
6,981	1-30T14:20:49	40.75	−73.98	Arts Entertainment:Music Venue	Rainy	15.98	0.49
12,951	01-31T17:40:35	40.73	−73.98	Food:Diner	Cloudy	15.85	0.49
137	01-30T01:50:57	40.75	−73.98	Nightlife:Bar	Cloudy	15.31	0.51

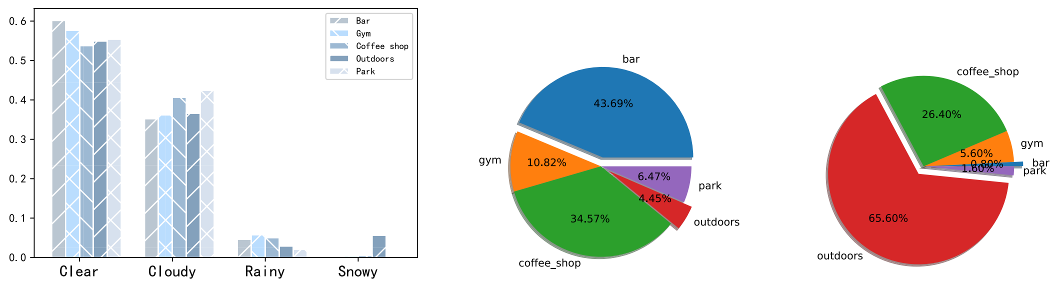
API [45]. The weather characteristics consist of the summary (e.g., clear, cloudy, rainy, and snowy), and the temperature and humidity of the current hour. The scale of the integrated dataset is shown in Table 1. We empirically split the travel sequence into small sub-sequences if the interval between two consecutive POIs is more than 8 hours. Other researchers have adopted a similar approach to divide such POI sequences [9]. Table 2 shows a specific example of user check-in in the dataset, which mainly includes user name, POI, the check-in time, the latitude and longitude of the POI, the POI category, and the weather information.

3.2 Data Analysis

In this section, we study the user check-in behaviors on POIs. To be specific, we explore the user check-in frequency when varying the weather and time, so as to validate their effects. Furthermore, we investigate the fine-grained features of the POI sequence, including sequence length, transfer frequency, transfer speed, and so on.

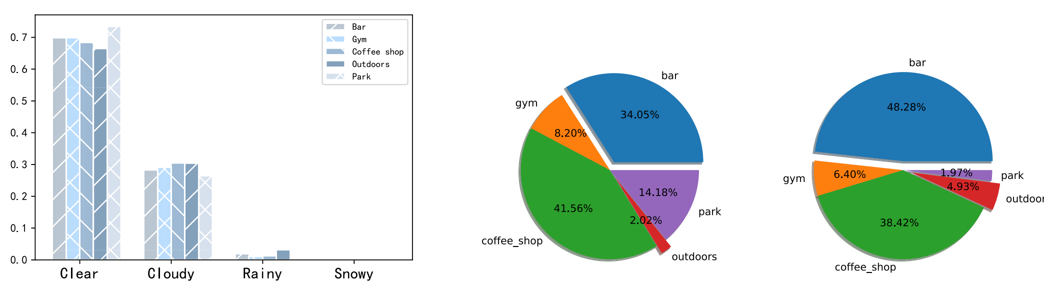
3.2.1 Weather Analysis. The weather is inseparable from people's daily life. People will look at the weather forecast in advance to make plans for the travel routes. The popularity of a POI is also affected by the weather. For example, as the temperature rises, the frequency of visits to an ice cream shop will increase; meanwhile, people tend to visit indoor POIs on rainy days. In order to verify that the weather can affect the user's check-in behaviors, we select five types of POIs in the two datasets, i.e., bars, gyms, cafes, outdoors, and parks. Furthermore, we analyze the check-in frequency of the five types of POIs in different weather in the New York and San Francisco datasets, respectively.

Figures 1 and 2 show the user's check-in frequency of each POI in the New York dataset under four different weather characteristics (i.e., clear, cloudy, rainy, and snowy). The abscissa of Figures 1(a) and 2(a) represent the weather feature, and the ordinate represents the check-in frequency. The frequency is calculated by the number of visits of each POI in the current weather feature divided by the total number of visits of the POI in the four weather categories. Figure 1(a) shows that when the weather is clear, users prefer to go out. When the weather conditions are bad, that is, on rainy and snowy days, the number of check-ins for the five types of POIs is significantly decreased. But for outdoor activities, the user's check-in frequency on a snowy day is greater than that on a rainy day. This phenomenon shows that people living in New York may like to go outdoors on snowy days. Similarly, Figure 2(a) shows that when the weather is clear, the overall check-in frequency is the highest, and when it rains or snows, the overall check-in frequency is low. Note that the user's check-in frequency is 0 on a snowy day since San Francisco has a special climate. That is, although it is cold in winter, it rarely snows.



(a) Visit frequency in four weather categories (b) Rainy (c) Snowy

Fig. 1. Check-in distributions of five types of POIs over four weather categories (New York).



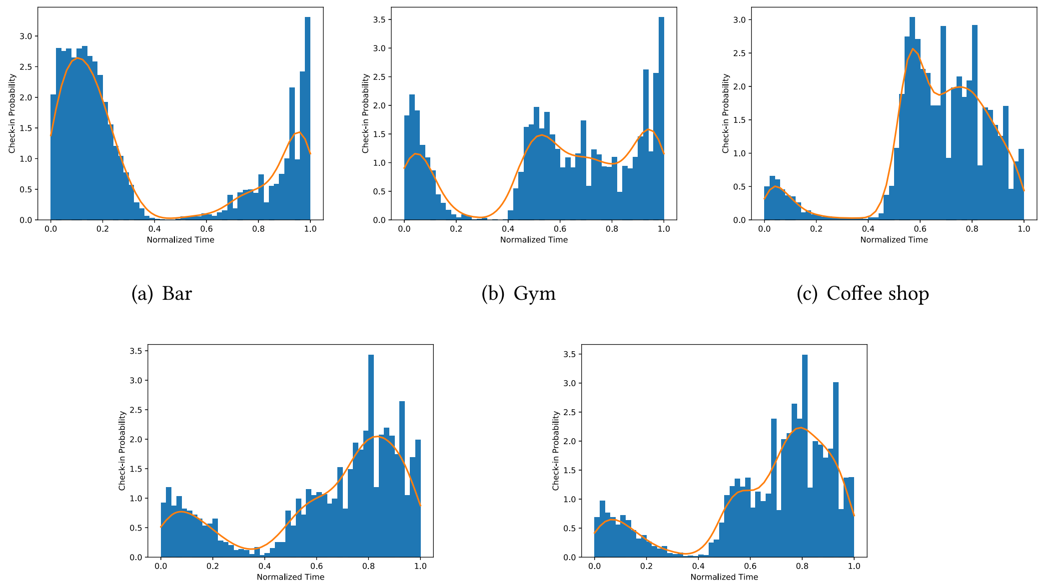
(a) Visit frequency in four weather categories (b) Clear (c) Rainy

Fig. 2. Check-in distributions of five types of POIs over four weather categories (San Francisco).

Figures 1(b) and (c) further show that different POIs are visited by users on rainy days and snowy days. It can be seen that on a rainy day, bars are the most popular in New York. Cafes are followed by the least number of visitors to outdoor activities. Unlike rainy days, on snowy days, outdoor activities are more popular than bars, parks, coffee shops, and gyms; bars are the least popular. Figures 2(b) and (c) show the popularity of each type of POI on sunny and rainy days, respectively, for San Francisco. It can be found that, unlike in New York, people living in San Francisco like to visit cafes when the weather is clear. On rainy days, the bar has the highest check-in frequency.

It should be noted that the datasets may not fully cover all check-in situations; hence, the check-in frequency of each category of POI in various kinds of weather only represents the situation of the dataset and may not conform to the actual local situation exactly. However, the main findings are suitable for most cases. Through the above data analysis and visualization, it can be seen that weather characteristics can affect the user's check-in behaviors, thereby affecting the POI popularity.

3.2.2 Temporal Analysis. According to Wikipedia, we mainly divide a day into six intervals (i.e., morning, forenoon, noon, afternoon, evening, and early morning). To be specific, we define 5:00 am to 9:00 am, 9:00 am to 11:00 am, 11:00 am to 2:00 pm, 2:00 pm to 5:00 pm, 5:00 pm to 8:00 pm, and 8:00 pm to 5:00 am as the morning, forenoon, noon, afternoon, evening, and night, respectively. It is worth noting that, we designate midnight as 12:00 am and noon as 12:00 pm, according to timeanddate.com. The weather and time play an important role in the user's check-in behaviors. For example, users are more inclined to visit bars at night. In order to investigate the impact of time on user check-in behaviors, we explore the check-in distribution of five types of POIs over



(a) Bar (b) Gym (c) Coffee shop

(d) Outdoors (e) Park

Fig. 3. The check-in distribution of five types of POIs over time in New York.

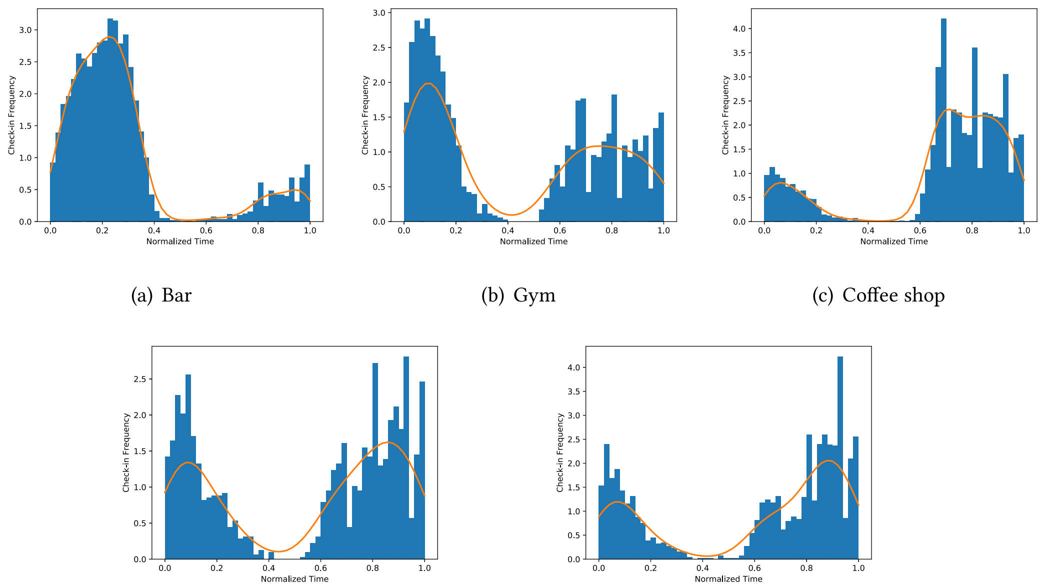
time. Figures 3 and 4 show the frequency distributions based on the New York and San Francisco datasets. The abscissa represents the normalized check-in time, that is, 0.0, 0.2, 0.4, 0.6, 0.8, and 1.0 represent 12:00 am, 4:48 am, 9:36 am, 2:24 pm, 7:12 pm, and 11:59 pm, respectively. The ordinate is the relative frequency (i.e., the actual frequency is equal to the relative frequency multiplied by the class interval).

Figure 3(a) shows that in New York, bars are frequently visited by people from 12:00 am to 5:00 am. Figure 3(b) shows that the number of visits to the gym is evenly distributed from 9:00 am to 12:00 am, and its most frequently visited time is around 12 o'clock at night. Figure 3(c) shows that people living in New York generally like to visit cafes around 2:00 pm or 7:00 pm. As shown in Figure 3(d) and (e), outdoor activities and parks have a similar frequency distribution. They are the most popular POIs around 7:00 pm.

Figure 4(a) shows that bars are frequently visited in San Francisco from 12:00 am to 8:00 am. Unlike New York, Figure 4(b) shows that San Francisco's gyms are generally the most popular POIs in the afternoon and evening. Figure 4(c) shows that people living in San Francisco generally like to visit cafes in the afternoon and evening. As shown in Figure 4(d), outdoor activities are more popular during the night, which may indicate that most people are busy at work during the day. Figure 4(e) shows that parks are the most popular around 8:00 pm. People may be inclined to visit parks in the evening.

Figures 3 and 4 reveal that users are used to visiting specific POIs at specific times, and the popularity of a POI is closely related to time. Moreover, POIs in different places (e.g., cities) may have different check-in or visiting patterns.

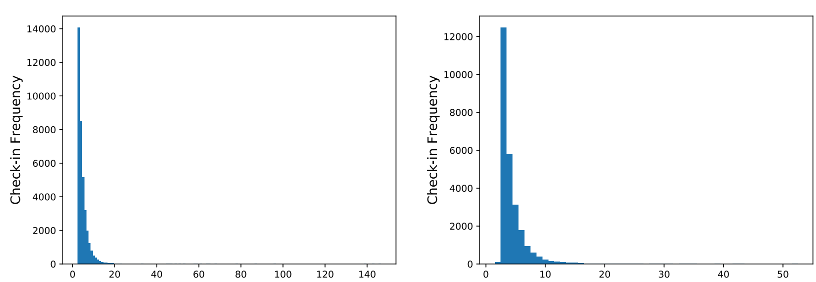
3.2.3 POI Sequence Feature Analysis. In this section, we explore the fine-grained POI sequence features. We first explore the distribution of sequence lengths in the two datasets of New York and San Francisco. We then analyze the transfer characteristics from one POI to another. Finally, we analyze the distribution of distance and time from one POI to the next.



(a) Bar (b) Gym (c) Coffee shop

(d) Outdoors (e) Park

Fig. 4. The check-in distribution of five types of POIs over time in San Francisco.

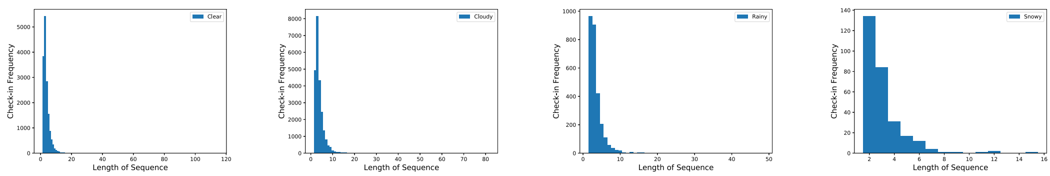


(a) New York (b) San Francisco

Fig. 5. The distribution of sequence length within 8 hours.

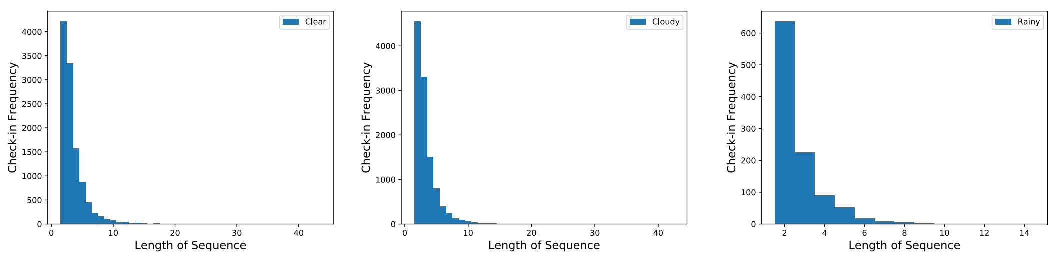
Sequence length distribution. In order to mine the relevant features of the sequence, we calculate the length distribution of the POI sequences, as shown in Figure 5. As mentioned in Section 3.1, we set the sequence interval as 8 hours. It can be seen from Figure 5(a) that, the lengths of about 30% sequences in the New York dataset are within 3; while that of 60% sequences are within [3, 6], and about 10% are larger than 6. Figure 5(b) shows the results in the San Francisco dataset. The distribution of sequence length is similar to that of the New York dataset. It means that people usually visit about three to six POIs within 8 hours. However, there are also some abnormal data. The sequence length is as high as 140 within 8 hours. Considering people usually will not visit too many POIs, we regard the POI sequence length over 20 as abnormal data.

In addition, we calculate the distribution of POI sequence lengths on the two datasets under different weather, as shown in Figures 6 and 7, respectively. From Figure 6, we can see that in New York, when the weather is clear, the sequence length is longer than that of other weather. Since it rarely snows in San Francisco, we only analyze the sequence length distribution under clear,



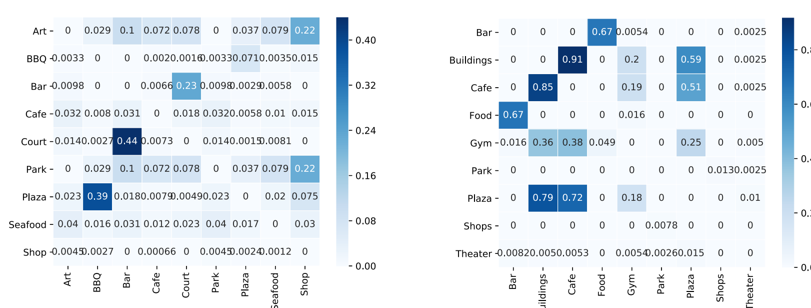
(a) clear (b) cloudy (c) rainy (d) snowy

Fig. 6. The distribution of sequence length over different weather in New York.



(a) clear (b) cloudy (c) rainy

Fig. 7. The distribution of sequence length over different weather in San Francisco.



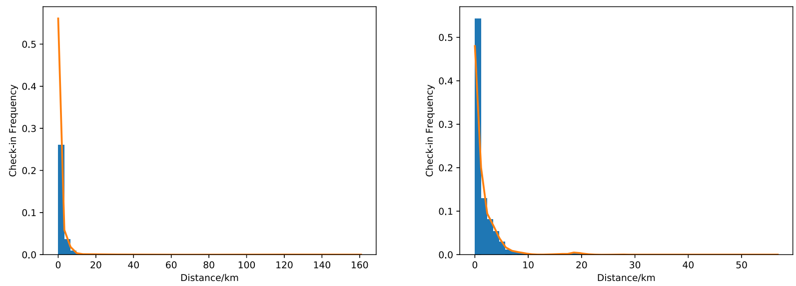
(a) New York (b) San Francisco

Fig. 8. The Probability from one POI to another.

cloudy, and rainy weather conditions (Figure 7). Similar to New York, the length of the sequence in San Francisco is shorter when the weather condition is not good. For example, when it is sunny, the length of the sequence is between 2 and 40, while that of rainy days is between 2 and 14.

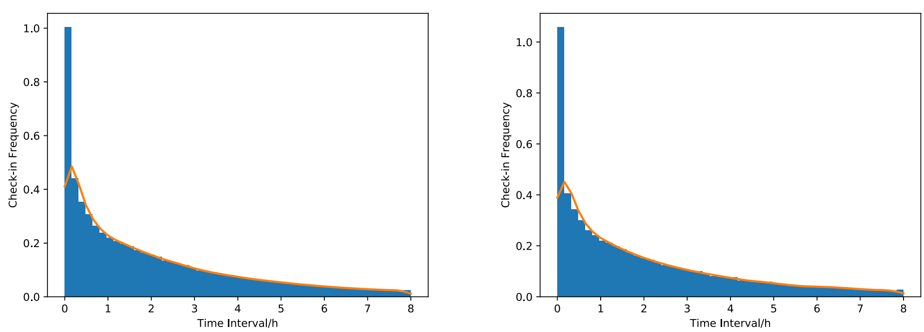
Through the above analysis, we can see that the weather is an important factor that affects the number of POI visits. Therefore, it should be considered for POI sequences recommendation.

Transfer characteristics of adjacent POIs. In order to find out the characteristics of POI transfer in the sequence, we calculate the transfer probability between nine adjacent POIs, as shown in Figure 8. It can be seen from Figure 8(a) that in New York, after visiting the Art Gallery, people tend to go shopping. Similarly, when people ask about the Plaza, they prefer to take BBQ. From Figure 8(b), it can be seen that after visiting the Buildings, people living in San Francisco go to the cafe to enjoy the coffee. Meanwhile, after visiting the Plaza, people will visit the nearby Buildings. It indicates that the probability of occurrence from one POI to another is different. Therefore, when recommending the POI sequence, we should consider the POI transition probability.



(a) New York (b) San Francisco

Fig. 9. The distribution of distance between successive check-in POIs.



(a) New York (b) San Francisco

Fig. 10. The distribution of time interval between successive check-in POIs.

We sort the check-in behaviors by time for each user. For the check-in sequence within 8 hours, we calculate the distance between every two adjacent POIs, as shown in Figure 9. It can be seen that when the adjacent distance of POIs is larger, the user's check-in probability is lower. It indicates that users are more likely to visit a POI closer to their current locations.

Similarly, we also statistically analyze the time interval distribution between two consecutive check-in POIs, as shown in Figure 10. It can be seen that the frequency distribution curves of the consecutive check-in intervals on the two datasets are similar. In the two datasets, about 50% of the time intervals between consecutive check-in POIs are within 1 hour, about 30% of them are distributed within (1,5] hours, and about 20% of them are distributed in (5,8] hours. As the time interval becomes larger, the probability of its occurrence becomes smaller.

From the above analysis, we can see that a comprehensive POI sequence recommendation model needs to consider the appropriate length of POI sequence, the residence time of users in different POIs, the user transfer speed, and so on.

3.2.4 Summary of Findings. Through the above data analysis, we gain the following main findings. First, weather can affect users' interest preferences, POI popularity, and the number of POI visits. What's more, the user's check-in preference is related to time. It should be noted that the transfer probability of adjacent POIs also reveals some popular sub-routes. Therefore, our POI sequence recommendation needs to consider the impact of weather and time on the user's interest and POI popularity, as well as the user's stay and transfer time in different POIs. Finally, when recommending POI sequences, we need to consider the transfer probability between POIs.

Table 3. Symbol Definitions

Symbol	Definition
U	set of users: $u_1, u_2, \ldots, u_{\|U\|}$
P	set of POIs: $p_1, p_2, \ldots, p_{\|P\|}$
S	a POI sequence
W	the weather context: {clear, cloudy, rainy, snowy}
T	the time set: {morning, forenoon, noon, afternoon, night, early morning}
$d(p_i, p_j)$	the distance between p_i and p_j
$Int(p, w, t)$	user interest for the POI p in current contexts, i.e., w and t
$Pop(p, w, t)$	the popularity of the POI p in current contexts, i.e., w and t
v	the speed that the user taken: {by walking, driving, taking bus}
$l_{dua}(p_i)$	the duration in p_i
$l_{tra}(p_i, p_j)$	the time taken from p_i to p_j

4 PROBLEM DEFINITION

In this section, we describe the concepts, the problem we solve, and our solution overview.

4.1 System Settings and Concepts

The notations and key concepts are described in Table 3. We denote $U : \{u_1, u_2, \ldots, u_{|U|}\}$ and $P : \{p_1, p_2, \ldots, p_{|P|}\}$ as the set of users and POIs, respectively. C denotes the category set of POIs: $\{c_1, c_2, \ldots, c_{|C|}\}$. Each POI is classified into a single category. We define the weather context as a variable $w \in W$, e.g., $W = \{sunny,\ overcast,\ rainy,\ snowy\}$. Similarly, the time context is denoted by $t \in T$. For example, Alice went to the coffee shop p on a sunny afternoon, where the longitude and latitude are 78 and 65, respectively. Then, the check-in record can be denoted as H : {*Alice, p, coffee shop, afternoon, clear, 78, 65*}.

Definition 1 (POI Sequence). A POI sequence consists of a set of POIs that are composed of the user's historical check-in records. We define the check-ins history as an ordered sequence, $S_{u:}\ \{H_1, H_2, \ldots, H_n\}$.

Definition 2 (Transfer Speed). Let v refer to the speed of a user transferring from a POI to the next. We provide a variety of transportation, including walking, cycling, taking a car, and taking the subway. According to some investigation, the range of walking speed is set as [3–7]. The range of speed by public transport is [30, 45]. That of a car is [35–50], and that of highway is [50–65]. All are in miles. In order to simulate the fact that different users may take different ways, in this article we randomly select transportation for the target users, and then randomly set the speed v of user transfer within the speed range of the selected transportation.

Definition 3 (Transfer Time). Transfer time refers to the length of time for which a user transfers from a place to another, which is denoted by l_{tra}. It is calculated by $d(p_i, p_j)/v$, where $d(p_i, p_j)$ is calculated by the Haversine formula [36], which exploits the latitude and longitude to measure the spherical distance between two POIs; and v is the transfer speed.

Definition 4 (Personalized Duration). Duration refers to the time that the user stayed at some POI. Similar to the number of visits, user's preference for different categories of POI can be reflected by the duration. In other words, user's duration in different POI categories can be different. For each user, we define the average time he has spent at that type of POI as his Personalized Duration on this POI. Since it is difficult to differentiate the actual Duration and the transfer time in two adjacent POIs, we propose to appropriately measure the personalized duration of the user u in POI p_i.

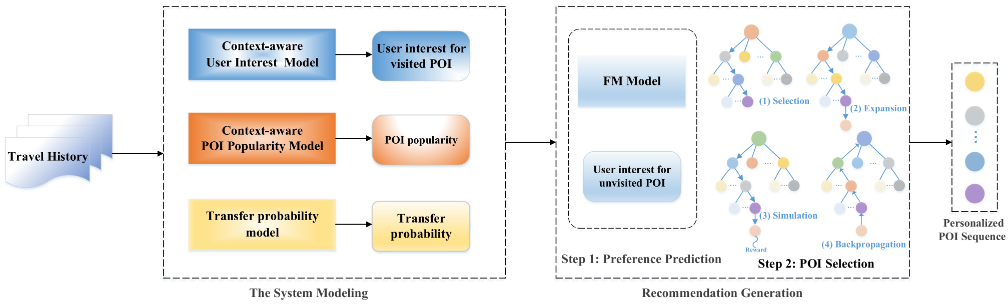


Fig. 11. The overview framework of *DPSR*.

4.2 The Problem of Dynamic Personalized POI Sequence Recommendation

Given the starting POI, p_0, the ending POI, p_n, and the total time budget for completing the POI sequence, the current weather w_0 and time t_0, the tasks of POI sequence recommendation for the target user are threefold: (1) to refine context-aware user interest and POI popularity with fine-grained approach, (2) to capture the feature of POI sequence and consider its attraction, and (3) to simulate the process of POI sequence recommendation in dynamic contexts. We recommend a POI sequence such that the following objective function is maximized:

$$\max\left(\sum_{p_i}^{|P|-1}\sum_{p_{i+1}}^{|P|} Pro(p_j|p_i) * \left[\alpha * Int_u(p_j, w, t) + (1-\alpha) * Pop(p_j, w, t))\right]\right), \tag{1}$$

where $Int_u\,(p, w, t)$ denotes u's preference on POI p, under the context, i.e., weather w and time t, and $Pop(p, w, t)$ denotes the POI popularity under the context that will be introduced in detail. $Pro(p_j|p_i)$ is the transfer probability from p_i to p_j. $|P|$ refers to the number of all POIs.

It is worth noting that, the input starting and ending POIs can be obtained in different ways. For instance, we can provide choices for users to select. We can also obtain the user's starting location according to the positioning system. If the user is not sure about his end point, we can also recommend a POI sequence that maximizes the sum of user interest and POI popularity.

The POI sequence recommendation problem is a variant of the **Orienteering Problem** (**OP**) [12]. OP originated from a sport, which determines the start and end points. If a participant visits a point, he will get a pre-arranged score. Participants reach the end from the starting point at the specified time and get the most points to win. OP has been proven to be an NP-hard problem [41]. Similar to OP, the dynamic personalized POI sequence recommendation with fine-grained contexts problem that we proposed also requires the input of the starting and ending POIs, and the user's time budget. The goal of this article is to determine the sequence between the starting POI and ending POI and recommend a POI sequence within a specified time that maximizes user interest and POI popularity while considering the transfer probability. Being more difficult than that OP, we consider the complex external environment, such as weather and time. Therefore, the proposed problem is also NP-hard.

4.3 Solution Overview

We propose a dynamic personalized POI sequence recommendation framework as shown in Figure 11. It has two components: system modeling and recommendation generation, as follows.

System Modeling: It has three parts, including the context-aware user interest model, the context-aware POI popularity model, and the transfer probability model.

— **Context-Aware User Interest Model:** This module captures user interests with fine-grained context. Specifically, for the POI that the user has visited, we measure his interest preference by the number of his visiting times under different weather and time.
— **Context-Aware POI Popularity Model:** The module is to enhance the popularity of POI sequence recommendation, which utilizes the number of POI visits under different weather and time to measure the popularity of a POI.
— **Transfer Probability Model:** This module is to mine the features of POI sequences and extract the attractive sub-routes from check-in histories, so as to ensure the rationality of the recommended POI sequences.

Recommendation Generation: It consists of preference prediction and dynamic POI selection. For POIs that users have not visited, we exploit the **Factorization Machine** (**FM**) models to predict users' interest in different contexts. Next, we exploit the **Monte Carlo Tree Search** (**MCTS**) algorithm to simulate the process of POI sequence selection in a dynamic environment. It incorporates user interest, POI popularity and POI transfer probability, as well as the stay time and transfer speed.

5 DPSR: THE ALGORITHM DETAILS

In this section, we describe the details of the *DPSR* model. We first introduce the system modeling of user preference, POI popularity, and transfer probability. Then, we conduct POI sequence recommendation, including preference prediction and dynamic POI selection.

5.1 The System Modeling

5.1.1 Context-Aware User Interest Model. We extract and refine the user interest preference with fine-grained context. It is difficult to directly measure user preferences, due to the lack of users' ratings on the visited POIs. Meanwhile, if users often go to a certain place or a certain type of place, it usually indicates that they like the place. Therefore, we exploit the number of check-ins on a POI to measure the user preference, i.e., the more frequently users visit, the more they like it.

We construct a user-POI-weather-time matrix by counting the number of times a user has visited a POI in different weather and time. The preference a user u gives to POI p_i in different weather w and time t, $Int_u(p_i, w, t)$, is calculated as follows:

$$Int_u(p_i, w, t) = \frac{n_u\ (p_i, w, t)}{\max(n_u(p_j, w, t))}, \tag{2}$$

where $n_u(p_i, w, t)$ denotes the visiting times of u on p_i in the current weather w and time t.

5.1.2 Context-Aware POI Popularity Model. Generally, the more people visit a POI, the more popular it is. People may first consider some local attractions when traveling, such as the Great Wall and the Forbidden City in China, the Great Barrier Reef in Australia, and the Golden Temple in India. Therefore, we propose a POI popularity model with fine-grained contexts. The popularity of a POI is measured by the number of visiting times on it in different weather and time by all users. It is calculated as follows:

$$Pop(p_i, w, t) = \frac{n(p_i, w, t)}{\max(n(p_j, w, t))}, \tag{3}$$

where $n\ (p_i, w, t)$ denotes the visiting times of the POI under the weather w and time t by all users.

5.1.3 Transfer Probability Model. The transfer probability model is to measure the probability of transferring from one POI to the next. However, in the current datasets, there isn't enough fine-grained context for the transfer probability model. Therefore, we resort to mining the attractive POI sub-sequences from check-in history, so as to enhance the recommendation performance.

$$\underbrace{[0, 1, 0, 0, \dots, 0]}_{\text{User ID}}\ \underbrace{[1, 0, 0, 0, \dots, 0]}_{\text{POI ID}}\ \underbrace{[0, 1, 0, 0]}_{\text{Weather ID}}\ \underbrace{[12]}_{\text{Time ID}}$$

Fig. 12. The example of a feature $x^{(i)}$.

In general, the user check-in records can reflect user preference and POI popularity. Furthermore, the POI sequence popularity can reflect the attractions of some sub-sequences. For example, people frequently visit some sub-sequences, which indicates they are very popular and classic.

We use $Pro(p_j|p_i)$ to denote the occurrence probability from p_i to p_j. We make use of the conditional probability to measure the probability of an event occurring given that another event has occurred in probability theory [14]. It is calculated as follows:

$$Pro(p_j|p_i) = \frac{Pro(p_i, p_j)}{Pro(p_i)}, \tag{4}$$

where $Pro(p_i, p_j)$ denotes the probability that p_i and p_j that appear in the same sequence S, which is calculated by dividing the number of p_i and p_j appearing in the same sequence S by the total number of sequences. $Pro(p_i)$ denotes the occurrence probability of p_i in the whole sequence.

5.2 Recommendation Generation

In this section, we construct the context-aware dynamic personalized POI recommendation model based on MCTS. Generally, a POI sequence may contain some POIs that the target user hasn't visited. In order to recommend the proper sequence that the user is interested in, it is necessary to conduct user interest prediction before recommendation. The details are as follows.

5.2.1 User Interest Prediction. For the POI that the user has not visited, we implement the FM model to predict the user's preference, which can easily model user preference with fine-grained context, and it can estimate parameters under huge sparsity and have good performance in POI recommendation [35]. We show in detail how FM can be applied to predict the preference user u on POI p_i, that has not been visited by u, in weather w and time t.

Given feature matrix $\mathbf{x}$ that each row represents a feature vector $x^{(i)}$ with its corresponding user preference $Int^{(i)}$, $x^{(i)}$ is composed of user ID, POI ID, weather, and time. Figure 12 shows an example. It utilizes one-hot encoding to depict contextual information, e.g., weather, time, and so on. To be specific, there are four types of weather, i.e., 1000, 0100, 0010, and 0001. Among them, 0 represents that this weather type does not occur and 1 means it occurs. Next, the time is represented with a real number and we can directly use the real number as a feature, e.g., 12 o'clock equals 12.

FM estimates the user preference on the POI by modeling all interaction between each pair of features via factorized interaction parameters:

$$\hat{Int}_u(\mathbf{x}) := z_0 + \sum_{i=1}^{n} z_i x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \hat{z}_{i,j} x_i x_j, \tag{5}$$

where z_0 is the global bias, z_i models the interaction of the i-th variable to the target, and $\hat{z}_{i,j}$ represents the factorized interaction parameters between pairs. It is calculated as follows.

$$\hat{z}_{i,j} := \langle \mathbf{a}_i, \mathbf{a}_j \rangle = \sum_{f=1}^{k} a_{i,f} \cdot a_{j,f}, \tag{6}$$

where $<,>$ is the dot product of two vectors of size k. a_i describes the i-th variable with k factors. k is a hyper-parameter that represents the dimension of the factorization. We use the gradient descent method to train the parameters [35].

5.2.2 POI Sequence Selection. We exploit the MCTS algorithm to dynamically recommend personalized POI sequences. MCTS is a probabilistic and heuristic-driven search algorithm, which is combined with the **Upper Confidence Bound Apply to Tree** (**UCT** for short) algorithm. It can make real-time decision-making in a dynamically changing environment. It starts with randomly selecting and exploring nodes, and then records the results of these selections. With several iterations, MCTS gradually deviates from the random result, shifts to the direction guided by the reward mechanism, and finally achieves the optimal result through convergence.

We extend the MCTS algorithm and propose a new reward mechanism to adapt to the dynamically personalized POI sequence recommendation scenario. After ensuring the user's starting and ending POIs, starting time, and time budget, the *DPSR* model will select the POI with the highest reward under the dynamically changing weather and time. It considers the user's interests, POI popularity, and transfer probability. Meanwhile, the model will reserve the stay time of users in the POI, taking into account the different transfer speeds. Finally, within a given time budget, it dynamically recommends a user-satisfied POI sequence.

Given the set of POIs (P) as the search space, MCTS aims to search a POI sequence that meets the user's dynamic preference with the dynamic context, while ensuring the popularity of the POIs. Furthermore, we consider recommendations according to the attractiveness of the POI subsequence. The POI sequence recommendation algorithm is divided into four levels: (1) Selection: using the selection algorithm to select the optimal POIs under different contexts. (2) Expansion: randomly expanding the next POI to avoid the local optimum. (3) Simulation: making use of the reward mechanism to evaluate the quality of the selected POI sequence. (4) Backpropagation: using the result of the playout to update information in the POIs on the path. The details are as follows.

Selection (Level one). In the selection process, we adopt the UCT algorithm to realize the POI selection under different weather and time. First, the current tree is traversed from the root node (i.e., the initial POI). Each node represents a POI. The branches of the tree are a sequence of POIs searched by the algorithm through multiple iterations. The UCT algorithm aims to balance exploration and exploitation. It retains POIs with high rewards in the current state, as well as explores POIs that are rarely visited. The POI with the highest estimated reward for the task is optimally selected so that the tree expands in the optimal direction. The UCT algorithm better simulates the process of user selection of a POI. During the traversal of the tree, a POI will be selected based on the rewards of some nodes, the number of visits, and the number of visits of its parent node. The principle is shown in the following formula.

$$exploit_{p_j} = Pro(p_j \mid p_i) \times \left[\alpha \times Int_u(p_j, w, t) + (1-\alpha) \times Pop(p_j, w, t)\right] + T_{R_{p_j}}/V_{p_j}, \tag{7}$$

$$explore_{p_j} = \sqrt{2 \ln V_{p_i}/V_{p_j}}, \tag{8}$$

$$UCT_{p_j} = exploit_{p_j} + C * explore_{p_j}, \tag{9}$$

where V_{p_j} and $T_{R_{p_j}}$ denote the total number and rewards of the p_j, respectively. p_i is the parent node of p_j. The higher the value of $exploit_{p_j}$, the more the POI can be selected for use. The higher the value of $explore_{p_j}$, it indicates that the node is rarely selected, so its exploration value is greater. When V_{p_j} is less, it is more likely to be explored. UCT_{p_j} denotes the value of POI p_j in the dynamic context, i.e., w and t. C is a constant and we set $C = 1/\sqrt{2}$, which Kocsis et al. [21] have proved to be the best value as it satisfies Hoeffding's inequality.

In the current iteration process, the model traverses all POIs P, and selects POIs by the selection algorithm. When the cut-off condition is reached, the POI selection is completed.

Expansion (Level Two). In the POI sequence recommendation, we first judge whether the current node is fully expanded, i.e., removing the POIs selected in the iteration process, whether the remaining POIs in P are all expanded to the child nodes of the current node. If so, it indicates it has all been expanded, and there is no need to expand new nodes, then use UCT calculation to select the best child node. If it is not, then a new node is randomly expanded from the remaining nodes as the child node of the current node.

Simulation (Level Three). The process is to evaluate the reward for the current round. For example, in the game, the reward is 1 if winning, otherwise 0. We propose a new reward mechanism to evaluate the quality of the POI sequence, which integrates the user's preference and the popularity of POI, as well as the transfer probability. The reward is calculated as follows:

$$Reward = Pro(p_j|p_i) * [\alpha * Int_u(p_j, w, t) + (1 - \alpha) * Pop(p_j, w, t)]. \tag{10}$$

α is a parameter that balances user interest and POI popularity. $Int_u(p_j, w, t)$ represents user interest in POI p_j under weather w and time t. If u has visited p_j, then Int_u can be obtained directly; otherwise, it can be predicted through the FM model. $Pro(p_j|p_i)$ refers to the probability of visiting p_j after p_i. $Pop(p_j, w, t)$ represents the popularity of p_j under weather w and time t.

Backpropagation (Level Four). It records the number of times each node in the tree was selected and the total reward value. When the game is completed in this round, it will return the result from the leaf node to the root node and update the number of selects and the reward value.

5.3 The Integrated Algorithm DPSR

We integrate all components and generate the final POI sequence recommendation (Algorithm 1).

In Algorithm 1, line 1 initializes two trees T_{visit} and T_{reward} that will be used to record the total reward and the visiting times of nodes. Within the time budget, the algorithm converges after many iterations (lines 2–18). Line 3 initializes an empty list P_{tem} to collect the selected POI according to the *SelectNode* algorithm (Algorithm 2) within a given time budget. Line 4 initializes *cost* and R, which is used to record the time cost and the reward of selecting a POI, respectively. Line 5 updates the current weather, time, and POI p_i. Next, line 6 adds the user's current POI p_0 to P_{tem}.

The algorithm will select as many nodes as possible within the time budget (lines 7–15). Line 8 conducts the *Selection* and *Expansion* of MCTS, which selects a node through the *SelectNode* algorithm (Algorithm 2). Then line 10 randomly selects the user's travel speed v. According to the speed, we can estimate the user's transfer time from p_i to p_j, i.e., l_{tra} (line 11). Finally, line 12 updates the time cost. Line 13 describes the *Simulation* step of MCTS, which calculates the reward of selecting a node with Equation (10). Next, lines 14 and 15 update the current weather, time, and POI.

When the time budget is exceeded, the algorithm conducts the *Backpropagation* step of MCTS (lines 16–18). It returns the node selection and rewards of each iteration, which are recorded in two trees T_{visit} and T_{reward}. Line 16 backtracks from the selected leaf node to the root node, and the visiting time increases by 1 of each selected node. When the leaf node is the end POI p_n and the length of P_{tem} doesn't equal 2, the reward increases by R. Otherwise, this selection has no reward (lines 17 and 18). If the recommended sequence length is 2, it only includes the start POI and the end POI, which is not a proper sequence. Algorithm 1 dynamically recommends a personalized POI sequence to the user (line 19).

The *SelectNode* algorithm (Algorithm 2) describes the process of node expansion. It has two stages. If the number of child nodes of the current POI is equal to the length of the optional POIs,

ALGORITHM 1: DPSR: Dynamic Personalized POI Sequence Recommendation

Input: The candidate POIs P; user interest Int; POI popularity Pop; transfer probability $Pro(p_i, p_j)$; the beginning POI p_0 and the end POI p_n; the travel time budget B; the number of iterations N; the current weather w_0 and time t_0; POI duration dur; the transfer time tra; parameter α

Output: A personalized POI sequence

1: Initialize two trees T_{visit} and T_{reward} to record the number of visit times and the total reward during the iterations;
2: **for** $iterations \leftarrow 1 \ldots N$ **do**
3:     Initialize P_{tem} to reserve the selected POI at each iteration;
4:     Initialize $cost$ and R to record the cumulative time and reward of each iteration, respectively: $cost \leftarrow 0$ and $R \leftarrow 0$
5:     The current node: $p_i \leftarrow p_0$, $w \leftarrow w_0$, and $t \leftarrow t_0$
6:     $P_{tem} \leftarrow p_0$
7:     **while** $cost < B$ **do**
8:         $p_j \leftarrow SelectNode(p_i, w, t, P_{tem}, T_{visit}, T_{reward}, P)$
9:         $P_{tem} \leftarrow p_j$
10:         $v \leftarrow Choose\_speed()$
11:         $l_{tra}(p_i, p_j) \leftarrow d(p_i, p_j)/v$
12:         $cost \leftarrow cost + l_{tra}(p_i, p_j) + l_{dur}(p_j)$
13:         $R \leftarrow R + reward$ (using Equation (10))
14:         Update the current time t and weather w
15:         Update current node: $p_i \leftarrow p_j$
16:     $BackPropC(P_{tem}, T_{visit})$
17:     **if** $p_j = p_n$ and $len(P_{tem})! = 2$ **then**
18:         $BackPropR(P_{tem}, T_{reward}, R)$
19: **return** the last POI sequence: P_{tem}

the algorithm will choose the next node according to Equation (9), and select the "best" child node, that is, letting the game tree expand toward the most promising moves. Otherwise, it will randomly expand the POI from the nodes that have not been selected. The details are as follows.

Algorithm 2 first determines whether p_i is fully expanded. That is, whether the rest selected set (i.e., P-P_{tem}) has been expanded to the child nodes of the current node p_i (line 3). If so, it will traverse all the child nodes of p_i (lines 5 and 6). Then it will select the "best" child node that has the highest UCT value under the current weather and time as the next visited POI p_j (lines 8–14). This step indicates that the *DPSR* model can select the optimal POI in the dynamic environment. If not, it will randomly select the next node p_j among the non-expanded child nodes (line 16). Finally, the last selected node p_j will be used in the *DPSR* algorithm (Algorithm 1, line 9).

Finally, within the time budget, *DPSR* recommends a personalized POI sequence that considers the changing contexts to the user. If the user has not set a destination or the last POI in the recommended sequence is not the one set by the user, our model will recommend the POI sequence obtained from the last iteration to the user.

Time complexity. In each iteration, the time complexity of the node selection algorithm is determined by the time budget and the length of the candidate POI, which is $O(k * |P|)$, where k is the length of POI sequence recommended and $|P|$ is the total number of candidate POIs. After N iterations, the total time complexity of *DPSR* is $O(N * k * |P|)$.

ALGORITHM 2: SelectNode

Input: the current selected POI p_i; the current weather w and time t; the two trees T_{visit} and T_{reward}; the selected POIs set P_{tem}; the candidate POIs P
Output: The selected next POI

$best\_score \leftarrow 0$
$best\_node \leftarrow Node$
if {the children node of p_i} $= P - P_{tem}$ **then**
  for p_j in {the children node of p_i} **do**
    if p_j in P_{tem} **then**
      continue
    else
      $exploit_{p_j} \leftarrow$ compute the value of $exploit\ p_j$ (using Equation (7))
      $explore_{p_j} \leftarrow$ compute the value of $explore\ p_j$ (using Equation (8))
      $UCT_{p_j} \leftarrow$ compute the value of $UCT\ p_j$ (using Equation (9))
      if $UCT_{p_j} > best\_score$ **then**
        $best\_score \leftarrow UCT_{p_j}$
        $best\_node \leftarrow p_j$
  return *best_node*
else
  Randomly expand a node from the rest set $P - P_{tem} -$ {the children node of p_i}
  return *node*

6 EXPERIMENTAL EVALUATION

In this section, we conduct extensive experiments on two real-world datasets to validate the performance of *DPSR*. The two datasets are described in Section 3.1.

In this article, 80% of POI sequences of each user is randomly selected as the training data, and the other 20% is used as test data. As default values, we set the number of iterations in the *DPSR* model to 1,000, the user interest weight α to 0.7, and the constant value C in the Algorithm 1 to $1/\sqrt{2}$. We select 100 POIs as candidate POIs P, including 50 user-interested POIs and 50 popular POIs. The time interval between a user's check-in time in the end POI and that of the start POI is taken as his time budget, and the weather at the beginning of the user's visit of the POI sequence is regarded as the current weather. Each time it selects a POI, the model updates the weather and time with current contexts.

6.1 Metrics

To verify the performance, we adopt the widely used evaluation metrics, i.e., **Precision**, **Recall**, and **F1 score**. Suppose p_r is the recommended POI sequence; P_t is the actual check-in sequence visited by the user. The metrics are calculated as follows.

Precision: it is calculated by $\frac{|P_r \cap P_t|}{|P_r|}$. $|P_r|$ denotes the length of recommended POI sequence. $|P_r \cap P_t|$ denotes the number of the same POIs of P_r and P_t.

Recall: it is calculated by $\frac{P_r \cap P_t}{P_t}$.

F1 score: $F1$ is defined as the harmonic mean of the accuracy and recall, as follows:

$$F1 = \frac{2 \times precision \times \text{recall}}{\text{precision} + \text{recall}}. \tag{11}$$

While being good at measuring whether POIs are correctly recommended, the above metrics on points ignore the visiting order between POIs. To address this drawback, Chen et al. [7] propose a metric that considers the relative visiting order. For instance, suppose the test sequence P_t is $\{p_1, p_2, p_3, p_4, p_5\}$, and the recommended POI sequence P_r is $\{p_1, p_2, p_5, p_3\}$. Then, there are five POI pairs that can be regarded as correctly recommended in [7], i.e., $\{p_1, p_2\}$, $\{p_1, p_5\}$, $\{p_1, p_3\}$, $\{p_2, p_3\}$, and $\{p_2, p_5\}$. However, this metric neglects the adjacent relation between two POIs, which may lead to unsuitable sequences.

Based on the above analysis, we propose three new metrics that consider both the visiting order and the adjacent relation, i.e., *pair_Pre*, *pair_Rec*, and *pair_F*1. The range of these three metrics is [0,1]. The details are as follows: first, we calculate the number of identical sub-sequences (i.e., adjacent POI pairs) contained in P_r and P_t, denoted as Q_r and Q_t, respectively. Then, *pair_Pre* is calculated by $\frac{|Q_r \cap Q_t|}{|Q_r|}$. *pair_Rec* is calculated by $\frac{|Q_r \cap Q_t|}{|Q_t|}$. *pair_F*1 is calculated by $2 \times pair\_Pre \times pair\_Rec/(pair\_Pre + pair\_Rec)$. Again, taking the above example, for instance, there is only one correctly recommended POI pair, i.e., p_1 to p_2. Then, the *pair_Pre* is 1/3, the *pair_Rec* is 1/4, and the *pair_F*1 is 2/7.

6.2 Baselines

We compare our method with the state-of-the-art models, as follows:

(1) *PopRank* is a basic method for recommending popular POIs to users based on POI popularity rankings. In order to ensure the fair comparison with *DPSR*, we implement Poprank with the same settings. That is, the start and end POIs, the same calculation method to get the stay and transfer time of POIs. Finally, we recommend the most popular POIs on sequence to users within the same time range, i.e., the check-in time difference from the start POI to the end POI by the target user.

(2) *Rand* is a random selection algorithm. Similarly, we use the same start and end POIs, randomly select the next POI that the user visits, use the same calculation method to get the stay and transfer time, and then recommend a POI sequence to the user within the specified time range.

(3) *Rank-MarkovPath* [7] is a probability model, which combines POI ranking and the POI to POI transitions as the feature in machine learning algorithms, and then recommends probable routes.

(4) *HGN* [31] is a **hierarchical gating network** (**HGN**) based sequence recommendation model. It exploits the **Bayesian Personalized Ranking** (**BPR**) model to capture both the long-term and short-term user interests. In order to suit the HGN model in the POI sequence recommendation scenarios and ensure the fairness of comparison, we use the same start and end POIs, and then select the top five POIs as the generated POI sequence for the user.

(5) *PersQ* [25] recommends personalized itineraries that take into consideration the attraction, popularity, user interests, and queuing times.

We also implement ablation models to test the effects of weather, time, transfer probability, and transfer speed, as in the following.

(6) *DPSR-No-probability*. When recommending the POI sequence, we consider transfer probability from one POI to the next. We ablate the transfer probability model to explore its effectiveness.

(7) *DPSR-No-context*. This method does not consider any context, which aims at investigating the impact of contexts on POI sequence recommendation task.

(8) *DPSR-No-temporal*. This method does not consider temporal influence on user interest and POI popularity. It only captures the weather-aware user interest and POI popularity.

(9) *DPSR-No-weather*. This method does not consider the impact of weather on user interest and POI popularity. In other words, It only captures the time-aware user interest and POI popularity.

Table 4. Comparison Results on Common Metrics

Method	New York			San Francisco		
	precision	recall	F1	precision	recall	F1
PopRank	0.274	0.533	0.340	0.238	0.511	0.305
Rand	0.277	0.535	0.345	0.266	0.513	0.327
Rank-MarkovPath	0.591	0.356	0.428	0.496	0.264	0.344
HGN	0.336	**0.709**	0.451	0.361	**0.677**	0.490
PersQ	0.535	0.558	0.507	0.523	0.526	0.484
DPSR-No-probability	0.621	0.579	0.601	0.578	0.536	0.507
DPSR-No-context	0.677	0.571	0.575	0.606	0.535	0.525
DPSR-No-temporal	0.691	0.580	0.591	0.609	0.530	0.518
DPSR-No-weather	0.694	0.579	0.594	0.604	0.536	0.513
DPSR-Constant-speed	0.761	0.577	0.588	0.538	0.522	0.53
DPSR	**0.772**	0.581	**0.665**	**0.644**	0.541	**0.544**

(10) *DPSR-Constant-speed*. Unlike *DPSR*, this method regards the users' transfer speed as a constant, e.g., the walking speed, 4 km/h.

6.3 Experimental Results

In this section, we validate the performance of our work in the two datasets, i.e., New York and San Francisco. We compare it with the state-of-the-art baselines in six metrics, i.e., precision, recall, F1, $pair\_Pre$, $pair\_Rec$, and $pair\_F1$. We also conduct ablation experiments to verify the impact of different modules.

6.3.1 Comparative Studies. To verify the performance of our model, we compare *DPSR* with the strong basic and advanced models on six metrics, i.e., *PopRank*, *HGN*, *Rank-MarkovPath*, and *PersQ*. The results are shown in Tables 4 and 5.

Comparison on common metrics. It can be seen from Table 4 that the precision and F1 score of the *DPSR* model are better than that of all baselines. *Rank-MarkovPath* and *PersQ* are the two best baselines. Compared with *Rank-MarkovPath* in the New York dataset, *DPSR* improves precision, recall, and F1 score by 31%, 63%, and 43%, respectively. While the improvement over *PersQ* is 43%, 4%, and 31%, respectively. Similarly, on the San Francisco dataset, *DPSR* also outperforms *PopRank*, *Rand*, *Rank-MarkovPath*, and *PersQ*. The only exception is *HGN*. Compared with *HGN*, *DPSR* shows better performance in precision and F1 score, but it is not as good as *HGN* in recall. For instance, compared with *HGN* in the New York dataset, *DPSR* improves precision and F1 score by 1.3 times and 47%, but it decreases the recall by 22%. We analyze the reason and find that *HGN* is more suitable for Top-N sequential recommendation. Its precision is lower and the recall is higher, indicating that *HGN* can recommend accurate POIs, rather than a suitable POI sequence. Compared with *HGN*, our *DPSR* model can flexibly recommend a suitable POI sequence according to the user's time budget and contexts, thus *DPSR* can recommend more accurate POI sequences to users.

Comparison on new metrics. Table 5 shows the performance on the new metrics. Since the values of *PopRank* on these metrics are infinitely close to 0, it is not shown in the table. On the two datasets of New York and San Francisco, our *DPSR* model performs significantly better than other baselines in $pair\_Pre$, $pair\_Rec$, and $pair\_F1$. For example, on the New York dataset, compared with *Rank-MarkovPath*, *DPSR* improves the $pair\_Pre$, $pair\_Rec$, and $pair\_F1$ by 39%, 5%, and 21%, respectively. Compared with *PersQ*, the improvements are 1.37 times, 1.32 times, and 1.37 times,

Table 5. Comparison Results on New Metrics

Method	New York			San Francisco		
	pair-Pre	pair-Rec	pair-F1	pair-Pre	pair-Rec	pair-F1
Rand	0.001	0.001	0.002	0.002	0.002	0.002
Rank-MarkovPath	0.041	0.055	0.047	0.029	0.035	0.032
HGN	0.017	0.016	0.02	0.029	0.024	0.035
PersQ	0.024	0.025	0.024	0.015	0.014	0.015
DPSR-No-probability	0.024	0.033	0.030	0.027	0.030	0.028
DPSR-No-context	0.051	0.050	0.051	0.037	0.037	0.037
DPSR-No-temporal	0.0541	0.054	0.055	0.035	0.036	0.036
DPSR-No-weather	0.054	0.055	0.055	0.039	0.040	0.039
DPSR-Constant-speed	0.056	0.055	0.056	0.047	0.046	0.046
DPSR	**0.057**	**0.058**	**0.057**	**0.049**	**0.049**	**0.049**

respectively. While compared with *HGN*, *DPSR* improves the $pair\_Pre$, $pair\_Rec$, and $pair\_F1$ by 2.3 times, 2.6 times, and 1.85 times, respectively. The performance of the *Rand* method on the three metrics is extremely poor. The reason may be that the baselines don't consider the sequence characteristics between POIs, and cannot capture attractive POI sub-sequences.

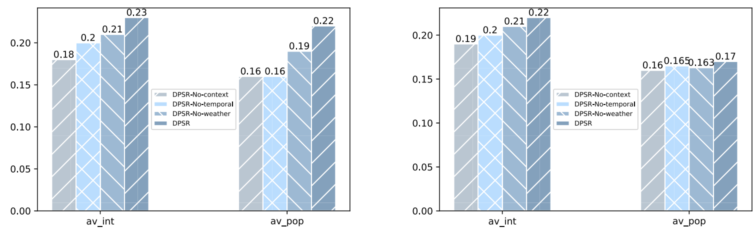
6.3.2 Variants of DPSR. We conduct extensive ablation experiments to investigate the effect of weather, time, transfer probability model, and transfer speed on POI sequence recommendation. Tables 4 and 5 show the comparison results of *DPSR* and its ablation models.

Variants performance on common metrics. In Table 4, compared with *DPSR-No-weather*, *DPSR-No-temporal*, *DPSR-No-context*, *DPSR-No-probability*, and *DPSR-Constant-speed*, *DPSR* performs the best in precision, recall, and F1 score. For example, on the New York dataset, compared with *DPSR-No-probability*, *DPSR* improves precision, recall, and F1 score by 24%, 0.3%, and 16%, respectively. The reason may be that the transfer probability model can capture the transfer characteristics of POI and popular POI sub-sequences, while introducing popular sub-sequences can improve the effectiveness of the system. Note that all ablation models perform better than *PopRank*, *Rand*, *Rank-MarkovPath*, and *PersQ* in precision and F1 score. For example, compared with *PersQ* on the New York dataset, *DPSR-No-weather* improves precision, recall, and F1 scores by 15%, 2%, and 6%, respectively. This further validates the effectiveness of the proposed *DPSR* model.

On the San Francisco dataset, compared with *DPSR-No-context*, *DPSR* improves precision, recall, and F1 score by 6%, 1%, and 4%, respectively. It indicates that the fine-grained user interest and POI popularity can improve the performance. Furthermore, the performance of *DPSR* in three metrics is better than that of *DPSR-Constant-speed*. The reason may be that *DPSR* considers different transfer speeds for users, which can better simulate the transfer behaviors between POIs.

Variants performance on new metrics. Similarly, it can be seen from Table 5 that on the two datasets, the performance of *DPSR* on $pair\_Pre$, $pair\_Rec$, and $pair\_F1$ is better than that of all ablation models. For example, in the New York dataset, compared with the *DPSR-No-probability*, *DPSR* increases $pair\_Pre$, $pair\_Rec$, and $pair\_F1$ by 1.38 times, 76%, and 90%, respectively. It indicates that *DPSR* can recommend correct POIs, as well as ensure the orderliness of POI sequences. What's more, we can see that all four ablation models perform better than baselines on the three new metrics. For example, on the San Francisco dataset, compared with *Rank-MarkovPath*, *DPSR-No-weather* increases $pair\_Pre$ and $pair\_F1$ by 31% and 17%, respectively.

However, the performance of *DPSR-No-probability* on the three metrics is worse than that of *Rank-MarkovPath*, and slightly better than *PersQ*. For example, compared with *PersQ* on the New



(a) New York (b) San Francisco

Fig. 13. The score of User Interest and POI Popularity.

York dataset, *DPSR-No-probability* increases *pair_Rec* and *pair_F*1 by 32% and 25%, respectively, and it has the same *pair_Pre* with *PersQ*. Compared with *Rank-MarkovPath*, *DPSR-No-probability* decreases *pair_Pre*, *pair_Rec*, and *pair_F*1 by 41%, 40%, and 36%, respectively. The results validate the importance of our transfer probability module, which can capture the transfer characteristics between POIs, as well as guarantee the order of the recommended POI sequence.

The quality of recommended sequence. We further investigate the quality of recommended sequence by *DPSR* and its variants by checking their interest degree and sequence popularity. To this end, we define two new metrics, the average interest (*av_int*) and popularity of sequence (*av_pop*). *av_int* is calculated by the total interest (*Int*) dividing the length of the recommended POI sequence, where the total interest denotes the sum of *Int* (see Equations (2) and (5)) for each POI in different contexts. Similarly, the average popularity (*av_pop*) is calculated by the total *Pop* (see Equation (3)) dividing the length of the recommended POI sequence, where the total *Pop* denotes the sum of popularity for each POI on a different context. The results are shown in Figure 13.

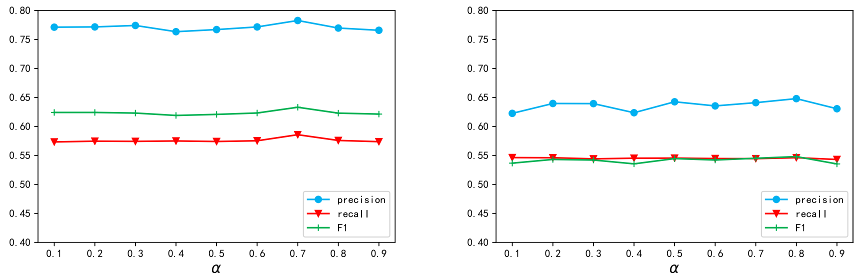
Compared with *DPSR-No-weather*, *DPSR-No-temporal*, and *DPSR-No-context*, the *DPSR* model performs better in terms of *av_int* and *av_pop*. For example, on the New York dataset, compared with *DPSR-No-context*, *DPSR* increases *av_int* and *av_pop* by 28% and 38%, respectively. Compared with *DPSR-No-temporal*, *DPSR* increases *av_int* and *av_pop* by 15% and 38%, respectively. The results indicate that by introducing popular POI sub-sequences, *DPSR* can recommend POI sequences that meet the user's interest, as well as ensure the popularity of the recommended sequences.

6.4 Parameters Sensitivity Analysis

We conduct sensitivity analysis on different parameters, including the user interest weight α in Equation (8), the number of algorithm iterations, and the time budget and transfer speed.

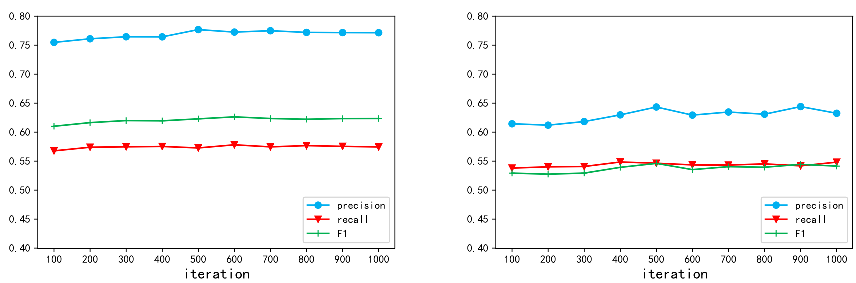
The effect of the interest weight α. α is used to explore whether users will focus on selecting POI sequences based on their interests, or whether they are more inclined to choose local popular POIs. Figure 14 shows the effect of α on the precision, recall, and F1 score of *DPSR*. It can be seen that, as α increases, the precision, recall, and F1 score of the *DPSR* model will fluctuate. In the New York datasets, when $\alpha = 0.7$, the overall performance of the *DPSR* model is the best. In the San Francisco dataset, when $\alpha = 0.8$, the *DPSR* model performs the best. Overall, the performance is relatively stable with the change of α. We analyze the reason and find that the average interest of all users and the average POI popularity are quite close to each other (e.g., 0.16–0.23 in Figure 13). Through analysis, it can be seen that users are more inclined to visit POI according to their preferences. Meanwhile, POI popularity also takes an important role.

The effect of iteration times. We check the influence of the number of iterations on the convergence of the *DPSR* model, as shown in Figure 15. Figure 15(a) shows that on the New York



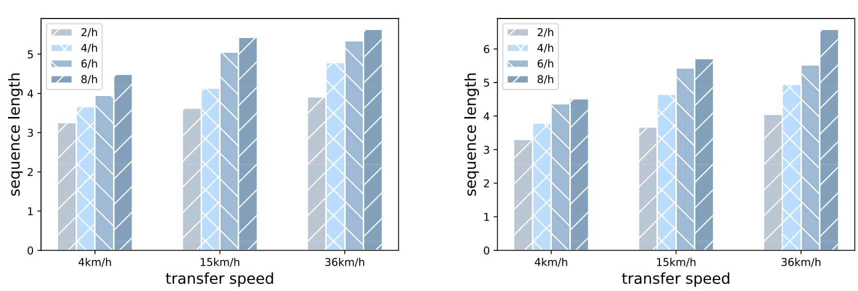
(a) New York (b) San Francisco

Fig. 14. Vary α for precision, recall, and F1.



(a) New York (b) San Francisco

Fig. 15. The number of iterations for the recommendation.

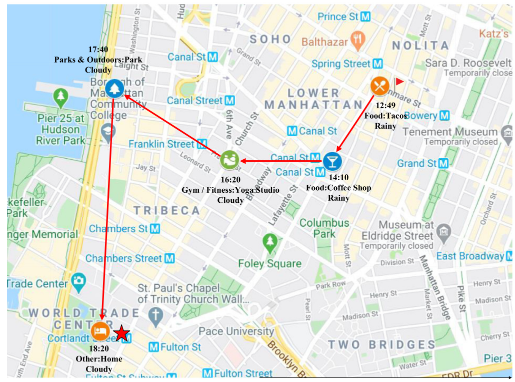


(a) New York (b) San Francisco

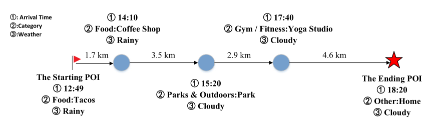
Fig. 16. The distribution of sequence length and check-in probability.

dataset, as the number of iterations increases, the precision, recall, and F1 score of the *DPSR* model first increase, and then remain relatively stable. When the number of iterations is 600, the overall performance of the *DPSR* model is the best, indicating that the algorithm has reached convergence. Similarly, Figure 15(b) shows the performance on the San Francisco dataset. It can be seen that as the number of iterations increases, the accuracy of the model will improve, and they tend to stabilize; the recall and F1 score remain stable. When the number of iterations reaches 500, the overall performance is the best.

The effect of time intervals on sequence length. We separately check the effect of different time budgets and different transfer speeds on the distribution of sequence length. We select 2,



(a) Recommended POI sequence on the map



(b) Intuitive Display of the POI sequence

Fig. 17. A case study showing the POI sequence recommended using *DPSR*.

4, 6, and 8 hours as the time budgets and 4 km/h, 15 km/h, and 36 km/h as the transfer speed, respectively. The results are shown in Figure 16. It shows that as the time budget increases, the length of the sequence also increases gradually. For example, in the New York dataset, when the speed is 4 km/h, the average sequence length in the 2-hour interval is approximately 3; and that in the 4-hour, 6-hour, and 8-hour intervals is 3.6, 4, and 4.5, respectively. Furthermore, as the transfer speed increases, the average sequence length also increases. For example, at a speed of 15 km/h, the average length in an 8-hour interval is about 5.5. When the speed is 36 km/h, the average length of the sequence is close to 6. This means that the larger the transfer speed, the longer the POI sequence. Meanwhile, the more transfer time a user spends, the more POIs can be visited. This also reveals that our model can recommend different POI sequence lengths according to user needs.

6.5 Case Study

In order to better illustrate the effect of POI sequence recommendation, we conduct a case study on the *DPSR* and its ablation models. We randomly select 20 different POIs and set the user's time budget to 5 hours. We set the starting POI to a restaurant called "the-stanton-social," and the destination is "75-wall-street." Figure 17(a) is the visualization result of the POI sequence recommended by the *DPSR* model on the real map. Figure 17(b) shows the recommended POI sequence, as well as the weather and time in each POI.

As can be seen from Figure 17, the POI sequences recommended by *DPSR* are "the-stanton-social," "the-coffee-shop," "herald-square," "crunch-fitness," and "75-wall-street." The distances between POI sequences are 1.7 km, 3.5 km, 2.9 km, and 4.6 km, respectively. When recommending the POI sequence, the current weather is raining, and the *DPSR* model recommends an indoor POI, that is, the cafe. After the user visits the cafe, at this time, the weather changes from rainy to cloudy, and the *DPSR* model recommends the user an outdoor park. Then, the model continues to recommend the gym to the user and finally recommends the destination given by the user. Through this case, it can be found that the *DPSR* model fully considers the dynamically changing context, and it can dynamically select the POI in combination with the current weather and time. Furthermore, the recommended POI sequence is more consistent with the user's real check-in behaviors.

What's more, our model recommends a POI sequence in about 30 s. The system will not display the dynamic process during the selection process, and the user will only get the result. We control the number of candidate POIs and the number of iterations to achieve real-time recommendations.

7 CONCLUSION

We deeply study the dynamic POI sequence recommendation in this article, considering the fine-grained context, particularly the external contexts (e.g., weather and time). We comprehensively study the impacts of weather and time on check-in behaviors, as well as the features of users' POI sequences. Based on the findings, we propose the *DPSR* model that incorporates fine-grained context and considers different transfer speeds and the attractiveness of sequences. It can be applied in a dynamically changing environment, to simulate the process of the user's POI selection in real-life scenarios. Experimental results on two real-world datasets validate the effectiveness of our work. In addition, *DPSR* can recommend POI sequences that meet the user's interest well, while ensuring the popularity of the recommended POI sequences. In future work, we are interested in enhancing the current transfer probability module in *DPSR* with fine-grained context, including mining user opinions [15, 20], interest [37], and sentiment [18] from review texts or tags. We would also like to combine the model with the front-end platform and apply it to real-life scenarios.

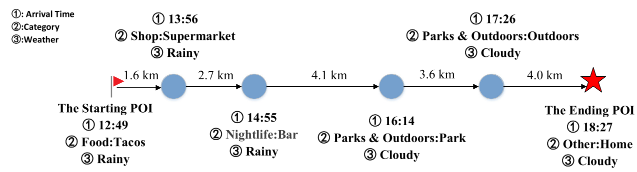
APPENDIX

A CASE STUDY OF ABLATION MODELS

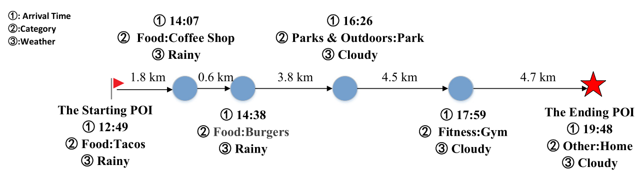
We also conduct the ablation experiments to treat the case study, as shown in Figure 18.

Figure 18(a) shows the result after the ablation of transfer probability model, and it is "la-esquina," "whole-foods," "lilys," "mccarren-park," "union-square," "75- wall-street." The categories are Food: Tacos, Supermarket, Bar, Park, Outdoors, and Home, respectively. It can be seen that the model recommended POI in the park at 16:14, and then recommended outdoor activities similar to the previous category at 17:26. It indicates that neglecting the POI transfer probability may lead to an unreasonable sequence.

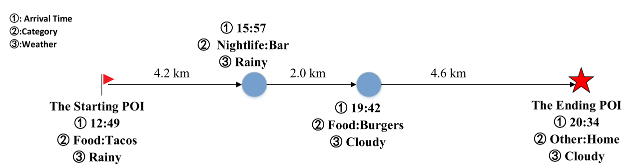
Figures 18(b), (c), and (d) show the recommended POI sequences after ablation of different contextual features (i.e., the weather and time). It can be seen from Figure 18(b) that the model that ablated weather and time features recommends users to visit POI ("shake-shack") at 14:38, and the category is also Food: Burgers, while the start POI category is Food. It means that the model cannot fully perceive time changes and recommends POIs that conform to the user's check-in behavior at a suitable time. Similarly, Figure 18(c) shows the result after ablating the time. It can be seen that the model is not sensitive to time changes. From the data analysis in Section 3, bars are usually more popular at night. However, the ablation model recommends the bar at 15:57. Figure 18(d) shows the result after the weather is ablated. We can see the model still recommends outdoor POIs to the user even when it rains, indicating that the ablation model cannot perceive changes in the weather. The above analysis shows that, compared to models that do not consider weather, time, or transfer probability, the *DPSR* model can recommend a POI sequence that is more reasonable and is met with the user's actual check-in behaviors.



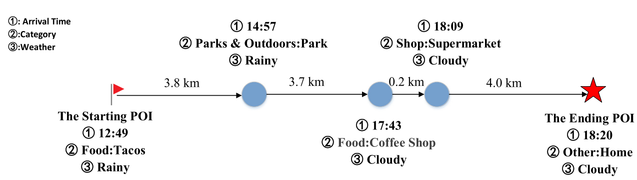
(a) Recommended POI sequence after ablation of transfer probability (DPSR-No-probability)



(b) Recommended POI sequence after ablation of weather and time (DPSR-No-context)



(c) Recommended POI sequence after ablation of time (DPSR-No-temporal)



(d) Recommended POI sequence after ablation of weather (DPSR-No-weather)

Fig. 18. POI sequences recommended by ablation models.

REFERENCES

[1] Gediminas Adomavicius, Ramesh Sankaranarayanan, Shahana Sen, and Alexander Tuzhilin. 2005. Incorporating contextual information in recommender systems using a multidimensional approach. *ACM Transactions on Information Systems (TOIS)* 23, 1 (2005), 103–145.

[2] Ramesh Baral, S. S. Iyengar, Tao Li, and N. Balakrishnan. 2018. CLoSe: Contextualized location sequence recommender. In *Proceedings of the 12th ACM Conference on Recommender Systems*. ACM, 470–474.

[3] Daniel Bienstock, Michel X. Goemans, David Simchi-Levi, and David Williamson. 1993. A note on the prize collecting traveling salesman problem. *Mathematical Programming* 59, 1-3 (1993), 413–420.

[4] Buru Chang, Yonggyu Park, Donghyeon Park, Seongsoon Kim, and Jaewoo Kang. 2018. Content-Aware hierarchical Point-of-Interest embedding model for successive POI recommendation. In *IJCAI*. 3301–3307.

[5] Chandra Chekuri, Nitish Korula, and Martin Pál. 2012. Improved algorithms for orienteering and related problems. *ACM Transactions on Algorithms (TALG)* 8, 3 (2012), 1–27.

[6] Chao Chen, Daqing Zhang, Bin Guo, Xiaojuan Ma, Gang Pan, and Zhaohui Wu. 2014. TripPlanner: Personalized trip planning leveraging heterogeneous crowdsourced digital footprints. *IEEE Transactions on Intelligent Transportation Systems* 16, 3 (2014), 1259–1273.

[7] Dawei Chen, Cheng Soon Ong, and Lexing Xie. 2016. Learning points and routes to recommend trajectories. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. ACM, 2227–2232.

[8] Lisi Chen, Shuo Shang, Christian S. Jensen, Bin Yao, Zhiwei Zhang, and Ling Shao. 2019. Effective and efficient reuse of past travel behavior for route recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 488–498.

[9] Munmun De Choudhury, Moran Feldman, Sihem Amer-Yahia, Nadav Golbandi, Ronny Lempel, and Cong Yu. 2010. Automatic construction of travel itineraries using social breadcrumbs. In *Proceedings of the 21st ACM Conference on Hypertext and Hypermedia*. ACM, 35–44.

[10] Madhuri Debnath, Praveen Kumar Tripathi, Ashis Kumer Biswas, and Ramez Elmasri. 2018. Preference aware travel route recommendation with temporal influence. In *Proceedings of the 2nd ACM SIGSPATIAL Workshop on Recommendations for Location-Based Services and Social Networks*. ACM, 2.

[11] Yunqi Dong and Wenjun Jiang. 2019. Brand purchase prediction based on time-evolving user behaviors in e-commerce. *Concurrency and Computation: Practice and Experience* 31, 1 (2019), e4882.1–e4882.15.

[12] Damianos Gavalas, Charalampos Konstantopoulos, Konstantinos Mastakas, and Grammati Pantziou. 2014. A survey on algorithmic approaches for solving tourist trip design problems. *Journal of Heuristics* 20, 3 (2014), 291–328.

[13] Jiqing Gu, Chao Song, Wenjun Jiang, Xiaomin Wang, and Ming Liu. 2020. Enhancing personalized trip recommendation with attractive routes. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI'20)*, Vol. 34. 662–669.

[14] Allan Gut. 2013. *Probability: A Graduate Course*. Vol. 75. Springer Science & Business Media.

[15] Jia-Wei He, Wen-Jun Jiang, Guo-Bang Chen, Yu-Quan Le, and Xiao-Fei Ding. 2022. Enhancing N-Gram based metrics with semantics for better evaluation of abstractive text summarization. *Journal of Computer Science and Technology* 37, 5 (Oct. 2022), 1118–1133. https://doi.org/10.1007/s11390-022-2125-6

[16] Liang He and Faqing Wu. 2009. A time-context-based collaborative filtering algorithm. In *2009 IEEE International Conference on Granular Computing*. IEEE, 209–213.

[17] Gang Hu, Jie Shao, Fumin Shen, Zi Huang, and Heng Tao Shen. 2017. Unifying multi-source social media data for personalized travel route planning. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 893–896.

[18] Chunli Huang, Wenjun Jiang, Jie Wu, and Guojun Wang. 2020. Personalized review recommendation based on users' aspect sentiment. *ACM Transactions on Internet Technology* 20, 4 (Oct. 2020), Article 42, 26 pages. https://doi.org/10.1145/3414841

[19] Liwei Huang, Yutao Ma, Shibo Wang, and Yanbo Liu. 2019. An attention-based spatiotemporal LSTM network for next POI recommendation. *IEEE Transactions on Services Computing* 14, 6 (2019), 1585–1597.

[20] Wenjun Jiang, Jing Chen, Xiaofei Ding, Jie Wu, Jiawei He, and Guojun Wang. 2021. Review summary generation in online systems: Frameworks for supervised and unsupervised scenarios. *ACM Transactions on the Web* 15, 3 (May 2021), Article 13, 33 pages. https://doi.org/10.1145/3448015

[21] Levente Kocsis, Csaba Szepesvári, and Jan Willemson. 2006. Improved Monte-Carlo search. University of Tartu, Estonia, Technical Report 1.

[22] The largest review site. 12th December, 2022. www.yelp.com.

[23] Mengxian Li, Wenjun Jiang, and Kenli Li. 2016. Recommendation systems in real applications: Algorithm and parallel architecture. In *International Conference on Security, Privacy and Anonymity in Computation, Communication and Storage*. Springer, 45–58.

[24] Defu Lian, Yongji Wu, Yong Ge, Xing Xie, and Enhong Chen. 2020. Geography-aware sequential location recommendation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2009–2019.

[25] Kwan Hui Lim, Jeffrey Chan, Shanika Karunasekera, and Christopher Leckie. 2017. Personalized itinerary recommendation with queuing time awareness. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 325–334.

[26] Kwan Hui Lim, Jeffrey Chan, Shanika Karunasekera, and Christopher Leckie. 2019. Tour recommendation and trip planning using location-based social media: A survey. *Knowledge and Information Systems* 60, 3 (2019), 1247–1275.

[27] Kwan Hui Lim, Jeffrey Chan, Christopher Leckie, and Shanika Karunasekera. 2018. Personalized trip recommendation for tourists based on user interests, points of interest visit durations and visit recency. *Knowledge and Information Systems* 54, 2 (2018), 375–406.

[28] X. Liu, Y. Liu, K. Aberer, and C. Miao. 2013. Personalized point-of-interest recommendation by mining users' preference transition. In *ACM International Conference on Conference on Information & Knowledge Management*.

[29] Yong Liu. 2016. https://www.yongliu.org/datasets/.

[30] Y. Liu, W. Wei, A. Sun, and C. Miao. 2014. Exploiting geographical neighborhood characteristics for location recommendation. In *ACM International Conference on Conference on Information & Knowledge Management*.

[31] Chen Ma, Peng Kang, and Xue Liu. 2019. Hierarchical gating networks for sequential recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 825–833.

[32] Jarana Manotumruksa, Craig Macdonald, and Iadh Ounis. 2017. A deep recurrent collaborative filtering framework for venue recommendation. In *Proceedings of the 2017 ACM Conference on Information and Knowledge Management*. 1429–1438.

[33] Changhua Pei, Xinru Yang, Qing Cui, Xiao Lin, Fei Sun, Peng Jiang, Wenwu Ou, and Yongfeng Zhang. 2019. Value-aware recommendation based on reinforced profit maximization in E-commerce systems. arXiv:1902.00851. http://arxiv.org/abs/1902.00851

[34] Mobile phone service website based on user location information (LBS). 2009. www.foursquare.com.

[35] Steffen Rendle. 2010. Factorization machines. In *2010 IEEE International Conference on Data Mining*. IEEE, 995–1000.

[36] C. Carl Robusto. 1957. The cosine-Haversine formula. *The American Mathematical Monthly* 64, 1 (1957), 38–40.

[37] Ming-Min Shao, Wen-Jun Jiang, Jie Wu, Yu-Qing Shi, TakShing Yum, and Ji Zhang. 2022. Improving friend recommendation for online learning with fine-grained evolving interest. *Journal of Computer Science and Technology* 37, 6 (Nov. 2022), 1444–1463. https://doi.org/10.1007/s11390-021-2124-z

[38] Yangkai Shi and Wenjun Jiang. 2017. Point-of-Interest recommendations: Capturing the geographical influence from local trajectories. In *2017 IEEE International Symposium on Parallel and Distributed Processing with Applications and 2017 IEEE International Conference on Ubiquitous Computing and Communications (ISPA'17/IUCC'17)*. IEEE, 1122–1129.

[39] Ke Sun, Tieyun Qian, Tong Chen, Yile Liang, Quoc Viet Hung Nguyen, and Hongzhi Yin. 2020. Where to go next: Modeling long-and short-term user preferences for point-of-interest recommendation. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI'20)*.

[40] Christoph Trattner, Alexander Oberegger, Lukas Eberhard, Denis Parra, and Leandro Balby Marinho. 2016. Understanding the impact of weather for POI recommendations. In *RecTour@ RecSys*. 16–23.

[41] Pieter Vansteenwegen, Wouter Souffriau, and Dirk Van Oudheusden. 2011. The orienteering problem: A survey. *European Journal of Operational Research* 209, 1 (2011), 1–10.

[42] Guojun Wang, Wenjun Jiang, Jie Wu, and Zhengli Xiong. 2014. Fine-grained feature-based social influence evaluation in online social networks. *IEEE Transactions on Parallel and Distributed Systems* 25, 9 (2014), 2286–2296.

[43] Jingyuan Wang, Ning Wu, Wayne Xin Zhao, Fanzhang Peng, and Xin Lin. 2019. Empowering A* search algorithms with neural networks for personalized route recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 539–547.

[44] Xiaoting Wang, Christopher Leckie, Jeffrey Chan, Kwan Hui Lim, and Tharshan Vaithianathan. 2016. Improving personalized trip recommendation by avoiding crowds. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. ACM, 25–34.

[45] Dark Sky weather API. 2016. https://api.darksky.net/forecast/.

[46] Peike Xia, Wenjun Jiang, Jie Wu, Surong Xiao, and Guojun Wang. 2021. Exploiting temporal dynamics in product reviews for dynamic sentiment prediction at the aspect level. *ACM Transactions on Knowledge Discovery from Data* 15, 4 (2021), 68:1–68:29.

[47] Xin Xin, Bo Chen, Xiangnan He, Dong Wang, Yue Ding, and Joemon Jose. 2019. CFM: Convolutional factorization machines for context-aware recommendation. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*. AAAI Press, 3926–3932.

[48] Zhenxing Xu, Ling Chen, and Gencai Chen. 2015. Topic based context-aware travel recommendation method exploiting geotagged photos. *Neurocomputing* 155 (2015), 99–107.

[49] Quan Yuan, Gao Cong, Zongyang Ma, Aixin Sun, and Nadia Magnenat Thalmann. 2013. Time-aware point-of-interest recommendation. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 363–372.

[50] Jifeng Zhang, Wenjun Jiang, Jinrui Zhang, Jie Wu, and Guojun Wang. 2021. Exploring weather data to predict activity attendance in event-based social network. *ACM Transactions on the Web (TWEB)* 15, 2 (2021) 10:1–10:25.

[51] Kangzhi Zhao, Yong Zhang, Hongzhi Yin, Jin Wang, Kai Zheng, Xiaofang Zhou, and Chunxiao Xing. 2020. Discovering subsequence patterns for next POI pecommendation. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence (IJCAI'20*), Christian Bessiere (Ed.). ijcai.org, 3216–3222.

[52] Pengpeng Zhao, Anjing Luo, Yanchi Liu, Fuzhen Zhuang, Jiajie Xu, Zhixu Li, Victor S. Sheng, and Xiaofang Zhou. 2020. Where to go next: A spatio-temporal gated network for next POI recommendation. *IEEE Transactions on Knowledge and Data Engineering* 34, 5 (2020), 2512–2524.

[53] Pengpeng Zhao, Chengfeng Xu, Yanchi Liu, Victor S. Sheng, Kai Zheng, Hui Xiong, and Xiaofang Zhou. 2021. Photo2Trip: Exploiting visual contents in geo-tagged photos for personalized tour recommendation. *IEEE Transactions on Knowledge and Data Engineering* 33, 4 (Apr. 2021), 1708–1721. https://doi.org/10.1109/TKDE.2019.2943854

Received 1 February 2022; revised 11 December 2022; accepted 31 January 2023