

Article

AST-PG: Attention-Based Spatial–Temporal Point-of-Interest-Group Model for Real-Time Point-of-Interest Recommendation

Huarui Yu  and Zesheng Cheng *

College of Computer Science and Technology, Qingdao University, Qingdao 266071, China; yuhuarui@qdu.edu.cn

* Correspondence: czs_110@hotmail.com

Abstract: Research on next-point-of-interest (POI) recommendation has become a new focus in the field of POI recommendation in recent years. The goal of POI recommendation tasks is to predict a user's future movement trajectory based on their current state and historical behavioral information. Recent studies have shown the effectiveness of neural network-based next-POI recommendation engines. However, most existing models only consider the correlation between consecutive visits, neglecting the complex dependencies of the POIs in the area and category features, as well as the processing of unstructured time series. This paper presents a new Attention-Based Spatial–Temporal Point-of-Interest-Group (AST-PG) model for POI recommendation. The model consists of a spatial module and a temporal module combined with each other by a multiple-attention mechanism. The spatial module in the proposed model groups the POIs based on geographic and category features, while the temporal module develops a uniform-length time trajectory vector for the unstructured temporal features. Comprehensive experimental results on two real datasets demonstrate that the proposed model of this study is superior to the state-of-the-art POI recommendation models in terms of performance.

Keywords: real-time POI recommendation; category–regional features; time trajectory vectors



Citation: Yu, H.; Cheng, Z. AST-PG: Attention-Based Spatial–Temporal Point-of-Interest-Group Model for Real-Time Point-of-Interest Recommendation. *Appl. Sci.* **2024**, *14*, 5337. <https://doi.org/10.3390/app14125337>

Academic Editor: Juan A. Gómez-Pulido

Received: 17 April 2024

Revised: 19 May 2024

Accepted: 3 June 2024

Published: 20 June 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Location-based social networks (LBSNs) [1], such as Foursquare, Gowalla, and Yelp, have experienced rapid growth in recent years due to the continuous improvement in mobile-device technologies [2–4]. These platforms are popular for users to share their experiences and explore their surroundings [5,6]. They have accumulated a large amount of user check-in data that reflect users' activity trajectories and preferences. These data provide an important source for the next-point-of-interest (POI) recommendation service [7,8].

It is important to note that users' behaviors are not only influenced by their historical preferences but also by their real-time preferences and time constraints. For example, Figure 1 illustrates that the user left home at 7:00 a.m., went shopping at 9:00 a.m., and then visited a cafe. When recommending the next point of interest (POI) to the user between 1:00 p.m. and 2:00 p.m., it is important to consider their real-time needs and preferences. Therefore, the user's trajectory should be used to mine their preferences at a specific time period for real-time POI recommendations. However, applying these trajectories to real-time POI recommendations is not simple due to two issues. Firstly, the time series is unstructured in terms of users' trajectories. Users have irregular time intervals between visits to different POIs, or they visit different POIs for inconsistent lengths of time in different time periods. This situation may be caused by the user's personal habits, practical needs, or environment, which makes it difficult for structured models to extract relevant temporal features. Additionally, the harvest data from LBSNs often contain information about the regions and categories to which the POIs belong [9]. The utilization of these features is insufficient in most commonly used POI recommendation engines, thereby limiting further improvements in the performances of these models.

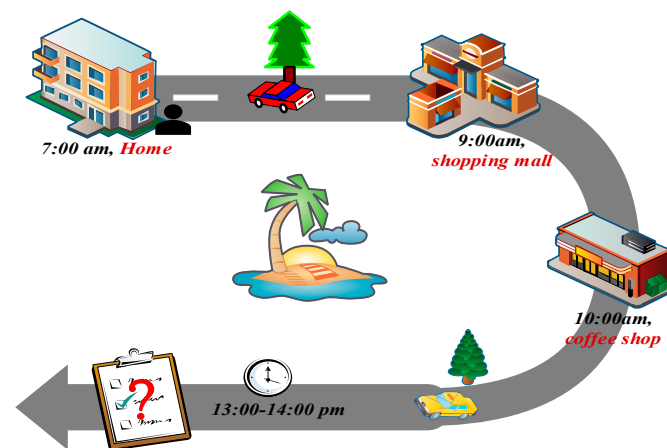


Figure 1. Example of time-series trajectories with varying lengths.

To address the above issues, this paper proposes a new real-time POI recommendation model, namely, the Attention-Based Spatial–Temporal Point-of-Interest-Group (AST-PG) model. The proposed model mines the transfer patterns of check-in behaviors in the sequence of user check-in records by a temporal module and a spatial module and then couples them together by a multi-head attention mechanism [10]. The temporal module converts the unstructured temporal trajectory information into a fixed-length vector.

Relevant experiments on large public datasets show that the model developed in this study has a state-of-the-art predictive performance. Moreover, the ablation experiments show that both the temporal and spatial modules developed in this study are effective.

2. Related Work

2.1. Preliminaries

In this paper, let $U = \{u_1, u_2, \dots, u_\mu\}$, $T = \{t_1, t_2, \dots, t_\tau\}$, and $P = \{p_1, p_2, \dots, p_\rho\}$ denote the set of users, timestamps, and candidate POIs, where μ , τ , and ρ denote the number of users, timestamps, and number of POIs, respectively.

Let $R = \{r_1, r_2, \dots, r_\gamma\}$ and $C = \{c_1, c_2, \dots, c_\zeta\}$ denote the sets of region and category levels of the POIs, where γ and ζ denote the number of regions and number of category levels of the POIs, respectively.

Let $G = \{(r_i, c_j), r_i \in R, c_j \in C\}$ denote the set of groups. Each POI ($p_k \in P$) is represented by a unique group (r_i, c_j) .

Based on the above symbols, the following concepts are introduced:

Category levels of the POIs: an integer representing the proportion of the check-in frequency by the target user for different categories relative to the frequency of the total check-ins by the target user;

Region of POIs: a two-tuple vector, consisting of the latitude and longitude of the POI;

Group of POIs: A bunch of POIs with the same region and category levels. Each POI is assigned to one and only one group;

Check-in record: A check-in record is represented as $ck = (u, p, t) \in U \times P \times T$, which represents a check-in record of the user (u) on the POI (p) at the timestamp (t). Since each POI is assigned to a group, the check-in can also be expressed with a tuple (u, r, c, t) . All check-in activities created by the user ($u \in U$) form a check-in sequence $(CK_u = \{ck_u^1, ck_u^2, ck_u^3, \dots\})$, where ck_u^l represents the l -th check-in record of the user (u).

2.2. POI Recommendation without Real-Time Performance

POI recommendation has emerged as a new research hot spot, attracting extensive attention from researchers in recent years. The early POI recommendation methods relied heavily on machine learning methods [8], especially stochastic models based on

Markov chains. Additionally, inspired by the successful application of matrix factorization (MF) [11,12] methods in other domains, researchers began integrating MF methods into the modeling of POI recommendations. The factorizing personalized Markov chain (FPMC) model [13] combines matrix factorization with Markov chains to model users' long-term preferences and sequential behaviors, aiming to achieve more accurate recommendation results.

Currently, Recurrent Neural Networks (RNNs) [14] have become the mainstream model for modeling user check-in sequences. The spatial-temporal recurrent neural network (ST-RNN) model [15] extends the RNN model by introducing a transformation matrix of temporal and distance features, which effectively captures the cyclic effects of time and space. In addition, variants of RNNs, such as LSTM [16], show good performances. The stochastic tailor-made gradient noise (STGN) model [17] modifies the basic LSTM by adding four new gating mechanisms, including two long-term gates and two short-term gates. The Spatial-Temporal-Aware Graph Convolutional Neural Network (STGCN) model [18] extends the STGN model by adding coupled input gates and forgetting gates to reduce the computational complexity. In addition, several studies have explored the attention mechanism to better model spatial data and rich contextual information. The Personalized Long-Short-Term Preference Learning (PLSPL) model [19] combines the attention mechanism with the LSTM to recommend the next POI. In this model, the attention mechanism is used to learn the user's long-term preferences, while the LSTM focuses on capturing the user's short-term preferences. By employing personalized linear weights, the PLSPL model comprehensively takes into account the user's overall preferences, thereby providing more accurate results. The Spatiotemporal Attention Network (STAN) model [20] uses a self-attention layer to mine the relationship between non-adjacent check-in records in a check-in record sequence of the user. However, these POI recommendation models tend to perform poorly in real-time POI recommendation scenarios because they lack attention to real-time user preferences and cannot effectively model preferences over time.

2.3. Real-Time POI Recommendation

Real-time POI recommendation is different from traditional POI recommendation in that it pays more attention to the user's possible real-time needs and preferences at a given time. The location recommendation model (LRT) [21] demonstrates its ability to improve the performance of the POI recommendation by learning the user behavior through the temporal patterns of check-in records and introduces four temporal aggregation strategies to integrate the user's check-in preferences for different time states. The real-time preference-mining (RTPM) model [22] uses the LSTM to mine the real-time preferences from the user's long-term and short-term preferences. For long-term preferences, the RTPM analyzes the cyclical trend of the user behavior between weeks. For short-term preferences, trainable time-shift vectors are introduced to model the user's publicly influenced preferences at the current time. The Timestamp Cross-Attention Network (TSCAN) model [23] uses a two-layer cross-attention network to predict the most relevant next POIs by focusing on the cross-over information between the timestamps through a timestamp cross-attention module; secondly, a cross-interval-aware module adjusts the POI sequences by using the time intervals to enhance the similarity of the neighboring POIs. The exploring user's preferences and real-time demand simultaneously (DSPR) model [24] utilizes various types of contextual information, including the absolute time, POI-POI transition time, and POI type, combined with user preferences, which are further modeled and automatically learned by an attention-based recurrent neural network model to support the final next-POI recommendation. However, these POI recommendation methods do not fully exploit the spatial correlation features and ignore the fact that users have different choices between different regions and different categories of POIs. Therefore, we propose a real-time POI recommendation model (AST-PG) that fully captures both the spatial and real-time information of the POIs, aiming to achieve better recommendation results.

2.4. POI Recommendation Based on Multi-Head Attention Mechanism

Multi-head attention is a prevalent attention mechanism in the domain of deep learning, particularly in the fields of natural language processing and computer vision. The fundamental concept is to parallelize the attention computation of the input information in multiple distinct projection spaces (or “heads”) in order to capture a more comprehensive range of features and contextual information. In [25], a novel long- and short-term preference learning (LSPL) model is proposed that combines an attention mechanism and a long- and short-term memory (LSTM) network to capture and analyze users’ long-term and short-term preferences, respectively. Specifically, the attention mechanism is used to focus and extract the user’s long-term stable interests, while the LSTM is responsible for capturing the user’s dynamically changing short-term preferences. By combining these two preferences, the model is able to provide users with more accurate and personalized POI (point-of-interest) recommendations. DeepMove [26] introduces an attentional framework that effectively captures multi-level periodic patterns. To model the sequential transitions for the final recommendations, the framework uses a recurrent neural network, which allows it to incorporate temporal dependencies and dynamics. The authors of [27] presented a multitasking transformer model dubbed the TLR-M, which incorporates a multi-head attention mechanism. This model not only recommends the next POI to the target user but also forecasts the estimated waiting time for concurrent visits to these POIs. In [28], a spatiotemporal long- and short-term memory (ST-LSTM) network is proposed. By feeding spatiotemporal context information into the LSTM network at each step, the ST-LSTM can better model the spatiotemporal information. In addition, an attention-based spatiotemporal LSTM (ATST-LSTM) network is developed for the next-POI recommendation. By using the attention mechanism, the ATST-LSTM can use spatiotemporal contextual information to focus selectively on the relevant historical check-in records in the check-in sequence.

3. Methodology

This section describes the structure of the proposed AST-PG model. As shown in Figure 2, the model consists of four modules: the time-embedding module, POI-group-embedding module, encoding–decoding module, and prediction module.

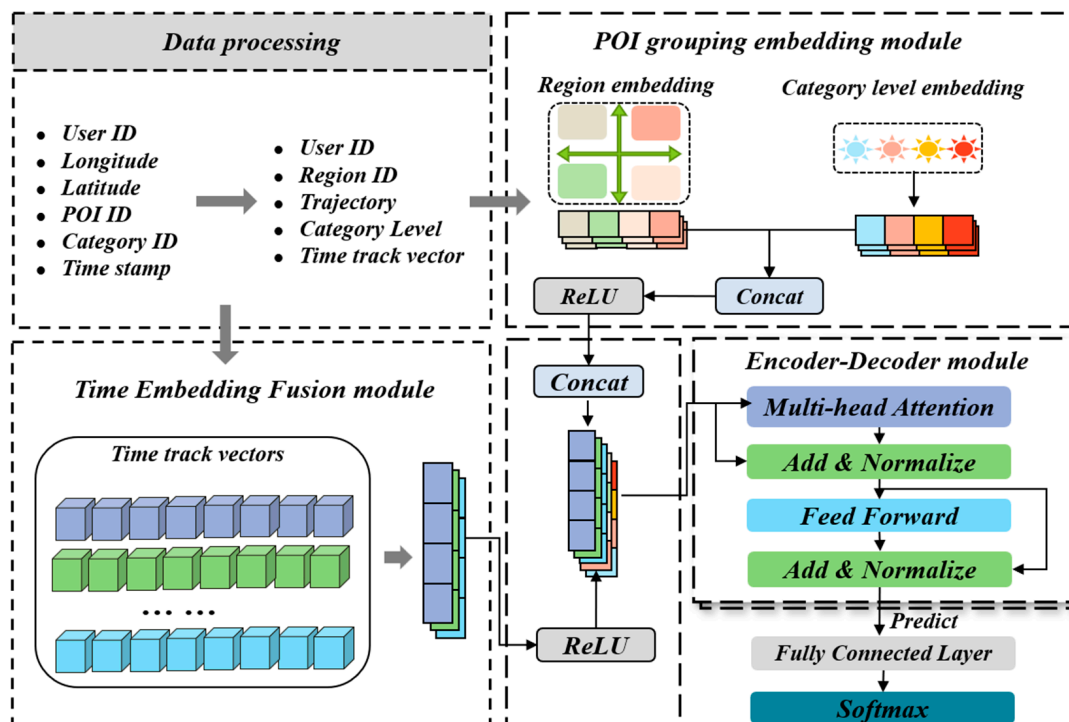


Figure 2. The architecture of the proposed AST-PG model.

3.1. Time-Embedding Module

This section describes the construction of the temporal-embedding module. To solve the problem of temporal disorder in the user trajectories, we convert the user’s time trajectory data into fixed-length time trajectory vectors (TTVs). We utilize these vectors to predict whether a user made a POI check-in at a specific time interval. First, we determine the time of the user’s first check-in point and then compute the check-in time for each subsequent point. These time points are then mapped to the corresponding locations in the time trace vector. The position associated with a check-in timestamp is assigned a value of 1, indicating that a check-in timestamp exists for that period. Conversely, the remaining locations are assigned a value of 0, indicating that there is no timestamp for that period. Presumably, if a location in the time track vector is 0, then the user will not access the POI in the period after that.

To further improve the accuracy of the time embedding, we add two dimensions after the 48-dimensional time track vector to indicate whether the track occurs mainly on weekdays or weekends. First, we iterate through all check-in events and count the number of check-ins on weekdays and weekends separately. If the number of weekday check-ins is more than half of the total number of check-ins in the trajectory, the first extra dimension (weekday dimension) should be set to 1; otherwise, it should be set to 0. Similarly, if the number of weekend check-ins is more than half of the total number of check-ins in the trajectory, the second extra dimension (weekend dimension) should be set to 1; otherwise, it should be set to 0. Figure 3 illustrates an example of temporal-data mapping. With this approach, we construct a 50-dimensional final trajectory vector that contains information about the temporal distribution and global information about whether the trajectory mainly occurs on weekdays or weekends.

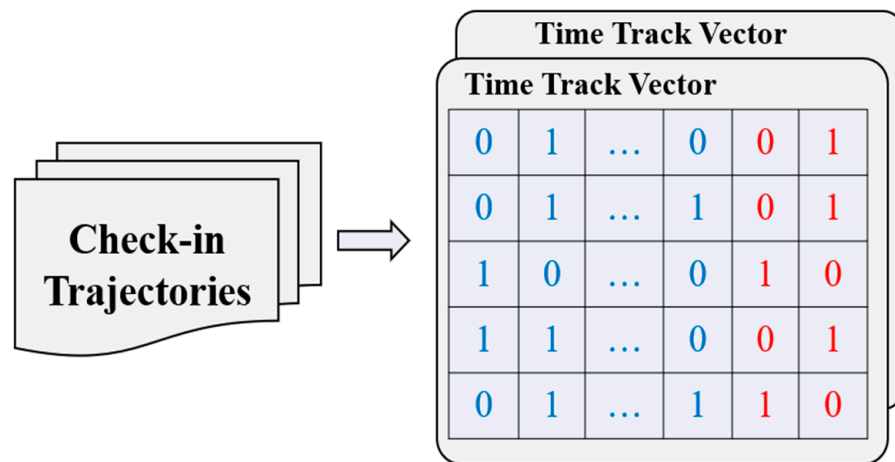


Figure 3. Example of forming time trajectory vector.

In order to capture the general patterns of the TTVs, we trained an embedding layer to map each TTV to a low-dimensional vector. The embedding for each TTV is learned from its historical check-in sequences. The embedding of the time trajectory vector (t), denoted as e_t , serves as the input data and is defined by Equation (1), as follows:

$$e_t = f^{embed}(t) \in R^\tau \tag{1}$$

where τ is the time trajectory vector-embedding dimension.

3.2. Geo-Categorical-Based POI-Group-Embedding Module

This section presents a strategy for categorizing points of interest (POIs) based on their geographic and categorical information. Initially, the POIs are divided into regions based on their latitude and longitude characteristics. Then, they are grouped into different geographic regions. As noted in [29], in reality, users are more likely to revisit a region than

a specific POI. When users do visit POIs, it is not necessarily because they are attracted to the POIs themselves, but rather because they are interested in the region.

To provide more insight into the importance of grouping POIs based on the region and category hierarchy, we are aided by two charts. Figure 4a shows in detail the distribution of the numbers of POIs across the 30 regions. It is evident from the chart that there are significant differences in the numbers of POIs in different regions. For example, the region with ID 20 tends to have many POI check-ins due to intensive user activity, while the region with ID 24 has relatively few. This distribution pattern reflects the degree of prosperity and user activity in different regions, which is essential for POI grouping. Figure 4b reveals the distribution of the numbers of POIs in different category classes within a given area. The category rank is calculated based on the ratio of the frequency of user check-ins in that category to the total number of check-ins. We can see from the chart that even within the same region, POIs in different categories may have different category ranks.

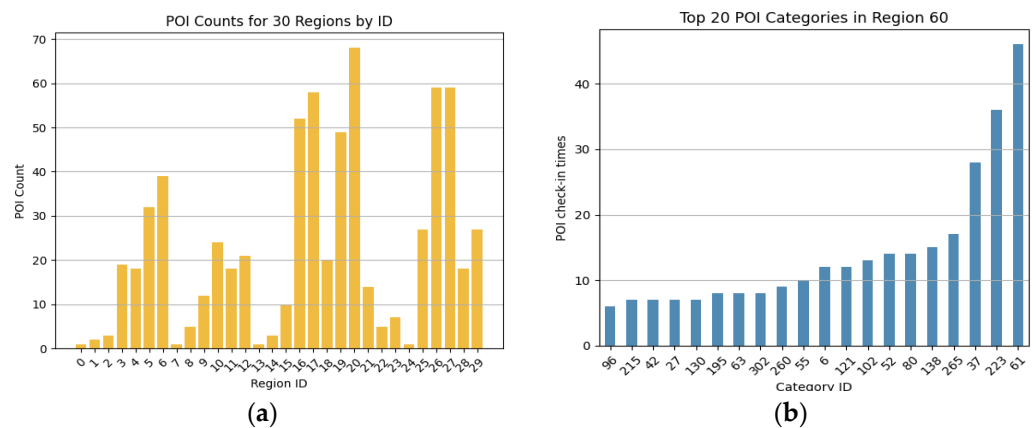


Figure 4. (a) Distribution of the numbers of POIs in different regions and (b) different category levels in the same region.

Accordingly, this study developed a geo-categorical-based POI-group-embedding module, which uses the embedding vectors of the regions and categories to which the target POI belongs to represent the spatial features. Figure 5 below illustrates the grouping process of the POIs. The POIs are divided into different geographic regions and graded according to the characteristics of their categories in each region. This helps to understand the interest in and importance of the different categories of POIs in each region in a more effective way. The grading is based on the obvious quantitative differences in the categories of POIs in each region. In short, with this grouping approach, we can better capture the attributes of these POIs and the access patterns of users to them. Even POIs that are accessed less frequently, as long as they represent a certain domain region and category rank value, will still be included in the corresponding POI group and have a chance to be recommended to the target users. Our prediction task is to predict the probability of a target user visiting a specific group from the POI-group level to generate the final POI recommendation list. This method not only improves the prediction efficiency but also improves the recommendation accuracy by aggregating similar POIs. At the same time, infrequently visited POIs may have higher representation in some specific POI groups, and thus they also have a chance to be prioritized higher in the recommendation list. We model spatial features using POI groups that consist of region and category features.

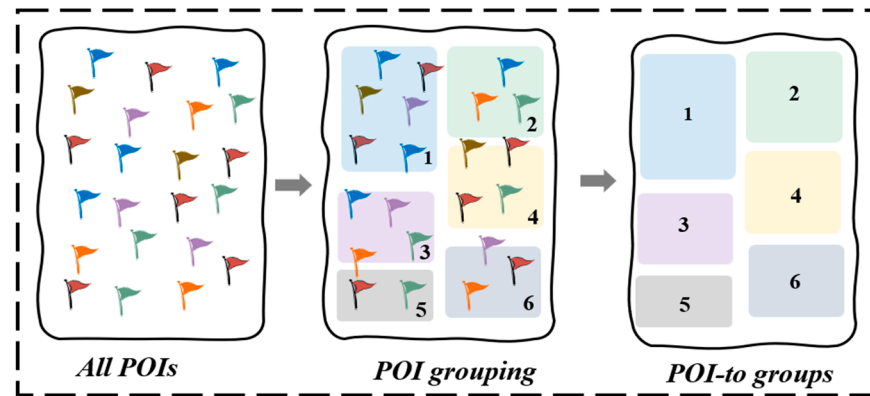


Figure 5. Example of geo-categorical-based POI grouping.

The region division of the POIs based on latitude and longitude information is calculated by the following Equations (2)–(4):

$$Lat^p = \left\lfloor \frac{Lat_0^p \times \alpha}{\alpha} \right\rfloor \quad (2)$$

$$Lon^p = \left\lfloor \frac{Lon_0^p \times \alpha}{\alpha} \right\rfloor \quad (3)$$

$$r^p = Cat(Lat^p, Lon^p) \quad (4)$$

where α is a pre-determined parameter employed to regulate the precision of the decimal places; Lat_0^p and Lon_0^p represent the original latitude and longitude of the POI (p), respectively; Lat^p and Lon^p represent the rounded latitude and longitude of the POI (p), respectively; r^p represents a specific region.

The category-level division of the POIs based on check-in time information is calculated by the following Equation (5):

$$CatL_u^r(c) = \left\lfloor \frac{Fre_u^r(c)}{Total_{Fre_u}^r} \times \beta \right\rfloor \quad (5)$$

where β is a pre-determined leveling parameter employed to regulate the precision of the decimal places; $CatL_u^r(c)$ represents the category level of the user (u) in region r , where c denotes the category; $Fre_u^r(c)$ indicates the frequency of check-ins for category c by the user (u) in region r (i.e., the number of times the user checks into the POIs of that specific category in the region); $Total_{Fre_u}^r$ denotes the total check-in frequency of the user (u) in region r (i.e., the total number of check-ins made by the user in that region).

Next, we fuse the region-index vector and category-level vector by a dense layer using Equation (6) below:

$$e_{r,CatL} = ReLU(Cat(e_r, e_{CatL})) \in R^{\gamma+\zeta} \quad (6)$$

where e_r and e_{CatL} are the embeddings of the region-index vector and category-level vector, respectively. The dimension of the output embedding ($e_{r,CatL}$) is the sum of the embeddings of the region- and category-level embeddings.

3.3. Encoder–Decoder Module

The embedding of each check-in record is obtained using the methods described in Sections 3.1 and 3.2. The embedding vectors of the input are concatenated to generate the inputs for layer 0 in the encoder. The input is represented as $\xi^0 \in R^{k \times d}$, where d is the

dimension of the check-in record embedding. The output of the first attention header in layer h is denoted by Equations (7)–(10) below:

$$H^h = \zeta^{h-1} w_Q^h \left(\zeta^{h-1} w_K^h \right)^T \quad (7)$$

$$H^{h'} = \sum_{i=1}^k \frac{\exp(H^{i,j})}{\sum_{j=1}^k \exp(H^{i,j})} \quad (8)$$

$$\text{head}_1^h = H^{h'} \zeta^{h-1} w_V^h \quad (9)$$

$$\text{MultiHead}(\zeta^h) = \text{Cat}(\text{head}_1^h; \dots; \text{head}_n^h) \times w_O \quad (10)$$

where w_Q^h , w_K^h , and w_V^h represent the learnable weight matrices corresponding to the query, key, and value of the h -th attention head, respectively; $H^{h'}$ represents the output of the h' -th attention head in the multi-head attention mechanism; w_O is the learnable weight of the multi-headed attention output in the linear layer. The units of the decoder model are essentially the same as those of the encoding module.

3.4. Prediction Module

The prediction model consists of several possible check-in locations for the target user at the next time slot, ranked in the order of probability after the fully connected layers. This study takes the most probable location as the prediction result and provides POI recommendations. More details about the experiment results and analysis can be found in Section 4.

4. Experiments

This section introduces the control and ablation experiments designed on two different datasets to validate the proposed data and model. It then analyzes the experimental results.

4.1. Experiment Settings

4.1.1. Problem Definition

We preprocess the check-in sequence (CK_u) of each user (u) by segmenting it into a set of consecutive trajectories, denoted as $CK_u = S_u^1 \oplus S_u^2 \oplus \dots$, where \oplus represents the concatenation. Each trajectory spans 48 h and contains a maximum of 10 and a minimum of 3 unique check-in locations that the user visits. To simplify the temporal analysis, we approximate the check-in times in the trajectories to hourly intervals. Specifically, for each hour within the 48 h, we retain only the earliest check-in location, as multiple check-ins within the same hour are rare in our dataset. This approximation helps maintain data conciseness while capturing the user's overall mobility patterns. If a user visits more than ten unique locations within 48 h, we segment the check-in sequence into multiple trajectories, each containing ten locations. In this way, a user can have multiple trajectories in the dataset, representing different periods.

In addition to the preprocessing steps mentioned above, we also introduce handling for weekdays and weekends in the trajectory data. When approximating the check-in times in the trajectory as hourly intervals, not only is the check-in time recorded but also whether each check-in point occurs on a weekday or weekend. Moreover, a label is added for each check-in point using binary coding (0 for weekdays and 1 for weekends). Suppose a user visits more than ten unique locations within 48 h, and these locations span both weekdays and weekends. In this case, the trajectories are segmented by weekday/weekend boundaries to ensure the accuracy of the data labels.

The task of our model is to predict the n -th check-in location of a trajectory containing n check-ins based solely on the previous $n-1$ check-in locations within that trajectory. This prediction capability enables us to capture the sequential nature of the user mobility and infer future check-in locations based on past behavior.

4.1.2. Experiment Environment

The model was implemented in PyTorch 1.7.1, and all programs were run in the Python 3.8 environment. This choice of Python version ensured compatibility with the latest libraries and tools, facilitating flexibility and efficiency in both the model development and data analysis. To facilitate the training and evaluation of the model, we used several key software libraries. These included PyTorch for deep learning operations, NumPy for numerical calculations, Pandas for data manipulation, and Matplotlib for data visualization. This combination of libraries, along with Python 3.8, provided a robust and reliable setup for running our experiments.

4.1.3. Datasets

We conducted experiments on two public datasets: Foursquare-NYC and Four-square-TKY. The Foursquare-NYC dataset was gathered in New York City between April 2012 and February 2013, and the Foursquare-TKY dataset was collected during the same timeframe in Tokyo. For each dataset, we sorted the check-in records by time and divided them into training, validation, and test datasets in an 8:1:1 ratio. Basic dataset statistics are shown in Table 1. The stability and reliability of these datasets have been extensively validated and used as benchmark data in several studies. These datasets also provide a reasonable exploration and validation of the effectiveness of the AST-PG model in the next-POI recommendation task. Moreover, it can be of great value in terms of rationally evaluating and comparing different recommender system approaches.

Table 1. Basic dataset statistics.

	Users	POIs	Categories	Check-Ins	Trajectories
NYK	1065	5031	305	112180	21643
TKY	2287	7863	297	396237	52823

4.1.4. Baseline

We compared our proposed method with several baseline models to evaluate its performance and effectiveness in next-POI recommendation. The following seven models were used as the baseline models in this study: the MF [11], FPMC [13], ST-RNN [15], LSTM [16], ATST-LSTM [28], STGN [17], STGCN [18], and STAN [20] models. The underlying theories associated with these models are briefly described in the Related Work Section. Except for the necessary modifications to the hyper-parameters, such as the dimensions of input and output that ensure program operation, the rest of the hyper-parameters, such as the network composition, learning rate, masks, etc., use the optimal values provided in the open-source codes of the methods.

4.1.5. Evaluation Matrix

We introduce the Accuracy (@K) (denoted by Acc @K) and Mean Reciprocal Rank (denoted by MRR) as the evaluation metrics for assessing the performances of our proposed method and the above-mentioned baseline models. Acc @K emphasizes the accuracy of the top K recommendations provided by the model, where K takes values of 1, 5, 10, and 20. The MRR is a metric used to evaluate the average ranking of the first correct next POI in the recommendation list provided by the model for a given user. Given a dataset with Q samples, the two metrics are calculated by Equations (11) and (12):

$$Acc@K = \frac{1}{Q} \sum_{i=1}^Q 1(rank \leq K) \quad (11)$$

$$MRR = \frac{1}{Q} \sum_{i=1}^Q \frac{1}{rank} \quad (12)$$

4.2. Experiment Results

As shown in Table 2, the model proposed in this study performed optimally in all five evaluation indicators compared to the baseline models. The evaluation indicators improved approximately 7–10%. In the baseline model, self-attention models like the STAN demonstrated superior performances compared to RNN-based models, such as the ST-RNN. This is attributed to the self-attention mechanism's ability to compute attention weights for all positions in the input sequence simultaneously, facilitating the efficient utilization of computational resources, particularly when processing lengthy sequences.

Table 2. Acc @K and MRR performance comparison of two datasets.

NYC					
	Acc@1	Acc@5	Acc@10	Acc@20	MRR
MF	0.0368	0.0961	0.1522	0.2375	0.0672
FPMC	0.1003	0.2126	0.2970	0.3323	0.1701
ST-RNN	0.1483	0.2923	0.3622	0.4502	0.2198
LSTM	0.1305	0.2719	0.3283	0.3568	0.1857
ATST-LSTM	0.1525	0.3208	0.3985	0.4816	0.2363
STGN	0.1716	0.3381	0.4122	0.5017	0.2598
STGCN	0.1799	0.3425	0.4279	0.5214	0.2788
STAN	0.2231	0.4582	0.5734	0.6328	0.3253
Ours	0.2378	0.5284	0.6304	0.6883	0.3665
TKY					
	Acc@1	Acc@5	Acc@10	Acc@20	MRR
MF	0.0241	0.0701	0.1267	0.1845	0.0861
FPMC	0.0814	0.2045	0.2746	0.3450	0.1344
ST-RNN	0.1409	0.3022	0.3577	0.4753	0.2212
LSTM	0.1335	0.2728	0.3277	0.3598	0.1834
ATST-LSTM	0.1596	0.3283	0.3642	0.4473	0.2408
STGN	0.1689	0.3391	0.3848	0.4514	0.2422
STGCN	0.1716	0.3453	0.3927	0.4763	0.2504
STAN	0.1963	0.3798	0.4464	0.5119	0.2852
Ours	0.1992	0.4371	0.5260	0.5981	0.3092

The traditional MF model is limited in terms of capturing the temporal pattern and temporal dynamics in the sequence of interest points, so it cannot effectively deal with the time-dependent relationship and spatial-temporal information in the sequence. While the FPMC model takes into account user preferences and contextual information in the sequence, its Markov assumption-based structure may not fully capture long-term time dependencies, resulting in an inadequate long-term predictive performance. ST-RNNs and LSTMs may face computational and storage challenges when dealing with large-scale spatial and temporal data, and modeling long-term dependencies may also be inadequate, resulting in degraded predictive performances over long time series.

The introduction of the multi-head attention mechanism in the ATST-LSTM model represents a significant advancement, as it enables the model to automatically assess the relevance of various inputs to the network at each step and adjust the attention weights of the inputs accordingly. This mechanism is highly beneficial to enhancing the model's comprehension and utilization of spatiotemporal contextual information, thereby improving the accuracy of the subsequent POI prediction. However, in contrast to our model, the ATST-LSTM does not appear to fully consider the impact of additional information, such as categories, on the POI prediction accuracy. Category information plays a pivotal role in recommender systems, as it can provide valuable insights into user preferences and POI characteristics.

Models such as the STGN, STGCN, and STAN may encounter computational and storage bottlenecks when dealing with large amounts of data and may be limited by the model complexity and training efficiency when considering complex spatial and temporal dependencies. Therefore, while these models have achieved some results in next-POI recommendation, there is still room for improvement. The comprehensive experimental results conducted on two real datasets demonstrate that our approach surpasses the existing state-of-the-art POI recommendation models in terms of performance.

4.3. Ablation Study

We conducted ablation experiments on the TKY dataset to evaluate the individual impact of each component on the overall performance of the model. Specifically, four experiments were conducted: (1) the remove geo-categorical-based POI-group-embedding model (denoted by “w/o GCP”); (2) the remove time and category information model (denoted by “w/o Time & Cat”); (3) the remove the multi-head attention mechanism module (denoted by “w/o multi-head attention”); (4) the use only the fully connected layer model (denoted by “Fully Connected”); (5) the full model. The results are shown in Table 3.

Table 3. Ablation study: comparing the full model with the three variants.

	Acc@1	Acc@5	Acc@10	Acc@20	MRR
w/o GCP	0.1829	0.4022	0.4596	0.5768	0.2984
w/o Time & Cat	0.1962	0.4251	0.4849	0.5859	0.3118
w/o multi-head attention	0.1884	0.4107	0.4638	0.5809	0.3043
Fully Connected	0.1873	0.4052	0.4721	0.5957	0.3067
Full Model	0.1992	0.4371	0.5260	0.5981	0.3092

As shown in Table 3, the full model has the best performance. For the rest of the components, the results show that the GCP plays a larger role in the overall performance than the other components. For example, without GCP embedding, the TOP-1 accuracy dropped from 19.92% to 18.29%. Other components also contributed to the final recommendation outcome. For example, with no time or category embeddings, the TOP-1 accuracy dropped from 19.92% to 19.62%. Multi-head attention mechanisms also play an important role in the model, enhancing the representation of the model by performing multiple independent attention operations in parallel and then merging the outputs. Without multi-head attention mechanisms, the TOP-1 accuracy dropped from 19.92% to 18.84%. To sum up, the ablation experiment proves the effectiveness of the attention-based time-embedding module and geo-categorical-based POI-group-embedding module developed in this study.

5. Conclusions and Future Work

In conclusion, the present study successfully demonstrated a real-time POI (point-of-interest) recommendation system leveraging social-media check-in data. The AST-PG model, which integrates a spatial module grouping POIs by geography and category, and a temporal module creating a uniform time trajectory vector, linked by a multi-attention mechanism, achieved a significant improvement in the POI recommendation accuracy, surpassing the state-of-the-art models by over 10%. This achievement demonstrates the efficacy of the AST-PG model in POI prediction and recommendation. Accurate POI recommendations have the potential to be utilized in a multitude of applications, particularly in the transportation sector. They can facilitate in-depth analyses of the travel demand, transportation modes, and user behaviors, ultimately contributing to the rational planning and optimization of transportation networks.

As we look towards the future, we intend to explore the integration of large models into our POI recommendation system. To accomplish this, we will investigate techniques for efficiently integrating large models into our current framework while maintaining its computational efficiency and scalability. This will involve optimizing the model architecture, input representations, and training procedures to ensure that the integrated

system can effectively handle large volumes of social-media data and provide real-time POI recommendations. Additionally, we plan to enhance the algorithmic efficiency of our POI recommendation system by exploring techniques such as parallel processing and distributed computing. Furthermore, we will continue to refine the input and output structures of our model to further improve the prediction speed and accuracy. Lastly, we aim to incorporate a dynamic adaptation mechanism into our system to respond to real-time changes in user preferences and spatial dynamics. This mechanism will leverage updates from social-media data to continually refine the POI recommendations, ultimately leading to an improved recommendation quality over time.

Author Contributions: Conceptualization, H.Y. and Z.C.; methodology, H.Y. and Z.C.; software, H.Y.; validation, Z.C.; writing—original draft preparation, H.Y.; visualization, H.Y. and Z.C.; supervision, Z.C.; project administration, Z.C.; funding acquisition, Z.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Natural Science Foundation of Shandong Province, Grant No. ZR2022QF144.

Data Availability Statement: Data is contained within the article.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Bao, J.; Zheng, Y.; Wilkie, D.; Mokbel, M. Recommendations in location-based social networks: A survey. *GeoInformatica* **2015**, *19*, 525–565. [[CrossRef](#)]
- Ye, M.; Yin, P.; Lee, W.C.; Lee, D.L. Exploiting geographical influence for collaborative point-of-interest recommendation. In Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, Beijing, China, 25–29 July 2011; pp. 24–28, 325–334.
- Zhang, J.D.; Chow, C.Y. Geosoca: Exploiting geographical, social and categorical correlations for point-of-interest recommendations. In Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, Santiago, Chile, 9–13 August 2015; pp. 9–13, 443–452.
- Han, P.; Li, Z.; Liu, Y.; Zhao, P.; Li, J.; Wang, H.; Shang, S. Contextualized point-of-interest recommendation. In Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, Yokohama, Japan, 7–15 January 2021; pp. 7–15.
- Carr, C.T.; Hayes, R.A. Social media: Defining, developing, and divining. *Atl. J. Commun.* **2015**, *23*, 46–65. [[CrossRef](#)]
- Chen, Y.-C.; Li, C.-T. Finding Potential Propagators and Customers in Location-Based Social Networks: An Embedding-Based Approach. *Appl. Sci.* **2020**, *10*, 8003. [[CrossRef](#)]
- Islam, M.A.; Mohammad, M.M.; Das, S.S.S.; Ali, M.E. A survey on deep learning based Point-of-Interest (POI) recommendations. *Neurocomputing* **2022**, *472*, 306–325. [[CrossRef](#)]
- Werneck, H.; Silva, N.; Viana, M.; Pereira, A.C.; Mourão, F.; Rocha, L. Points of interest recommendations: Methods, evaluation, and future directions. *Inf. Syst.* **2021**, *101*, 101789. [[CrossRef](#)]
- Narayanan, M.; Cherukuri, A.K. A study and analysis of recommendation systems for location-based social network (LBSN) with big data. *IIMB Manag. Rev.* **2016**, *28*, 25–30. [[CrossRef](#)]
- Niu, Z.; Zhong, G.; Yu, H. A review on the attention mechanism of deep learning. *Neurocomputing* **2021**, *452*, 48–62. [[CrossRef](#)]
- Mahesh, B. Machine learning algorithms—A review. *Int. J. Sci. Res.* **2020**, *9*, 381–386.
- Koren, Y.; Bell, R.; Volinsky, C. Matrix factorization techniques for recommender systems. *Computer* **2009**, *42*, 30–37. [[CrossRef](#)]
- Rendle, S.; Freudenthaler, C.; Schmidt-Thieme, L. Factorizing personalized Markov chains for next-basket recommendation. In Proceedings of the 19th International Conference on World Wide Web, Raleigh, NC, USA, 26–30 April 2010; pp. 26–30, 811–820.
- Koutník, J.; Greff, K.; Gomez, F.; Schmidhuber, J. A clockwork rnn. In Proceedings of the International Conference on Machine Learning, PMLR, Beijing, China, 21–26 June 2014; pp. 22–24, 1863–1871.
- Cheng, Y.; Wu, B.; Song, L.; Shi, C. Spatial-temporal recurrent neural network for anomalous trajectories detection. In Proceedings of the Advanced Data Mining and Applications: 15th International Conference, ADMA 2019, Dalian, China, 21–23 November 2019; Proceedings 15. Springer: Cham, Switzerland, 2019; pp. 21–23, 565–578.
- Greff, K.; Srivastava, R.K.; Koutník, J.; Steunebrink, B.R.; Schmidhuber, J. LSTM: A search space odyssey. *IEEE Trans. Neural Netw. Learn. Syst.* **2016**, *28*, 2222–2232. [[CrossRef](#)] [[PubMed](#)]
- Wu, T.; Ding, X.; Tang, M.; Zhang, H.; Qin, B.; Liu, T. Stgn: An implicit regularization method for learning with noisy labels in natural language processing. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, Abu Dhabi, United Arab Emirates, 7–11 December 2022; pp. 7–11, 7587–7598.
- Han, H.; Zhang, M.; Hou, M.; Zhang, F.; Wang, Z.; Chen, E.; Wang, H.; Ma, J.; Liu, Q. STGCN: A spatial-temporal aware graph learning method for POI recommendation. In Proceedings of the 2020 IEEE International Conference on Data Mining (ICDM), Sorrento, Italy, 17–20 November 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 17–20, 1052–1057.

19. Wu, Y.; Li, K.; Zhao, G.; Qian, X. Personalized long-and short-term preference learning for next POI recommendation. *IEEE Trans. Knowl. Data Eng.* **2020**, *34*, 1944–1957. [[CrossRef](#)]
20. Luo, Y.; Liu, Q.; Liu, Z. Stan: Spatio-temporal attention network for next location recommendation. In Proceedings of the Web Conference 2021, Ljubljana, Slovenia, 19–23 April 2021; pp. 19–23, 2177–2185.
21. Gao, H.; Tang, J.; Hu, X.; Liu, H. Exploring temporal effects for location recommendation on location-based social networks. In Proceedings of the 7th ACM Conference on Recommender Systems, Hong Kong, China, 12–16 October 2013; pp. 12–16, 93–100.
22. Liu, X.; Yang, Y.; Xu, Y.; Yang, F.; Huang, Q.; Wang, H. Real-time POI recommendation via modeling long-and short-term user preferences. *Neurocomputing* **2022**, *467*, 454–464. [[CrossRef](#)]
23. Duan, J.; Meng, X.; Liu, G. Where To Go at the Next Timestamp. *Data Sci. Eng.* **2024**, *9*, 88–101. [[CrossRef](#)]
24. Wang, H.; Li, P.; Liu, Y.; Shao, J. Towards real-time demand-aware sequential POI recommendation. *Inf. Sci.* **2021**, *547*, 482–497. [[CrossRef](#)]
25. Wu, Y.; Li, K.; Zhao, G.; Qian, X. Long- and short-term preference learning for next POI recommendation. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management, Beijing, China, 3–7 November 2019; pp. 3–7, 2301–2304.
26. Feng, J.; Li, Y.; Zhang, C.; Sun, F.; Meng, F.; Guo, A.; Jin, D. Deepmove: Predicting human mobility with attentional recurrent networks. In Proceedings of the 2018 World Wide Web Conference, Lyon, France, 23–27 April 2018; pp. 23–27, 1459–1468.
27. Halder, S.; Lim, K.H.; Chan, J.; Zhang, X. Transformer-based multi-task learning for queuing time aware next poi recommendation. In Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining, Virtual Event, 11–14 May 2021; pp. 11–14, 510–523.
28. Huang, L.; Ma, Y.; Wang, S.; Liu, Y. An attention-based spatiotemporal lstm network for next poi recommendation. *IEEE Trans. Serv. Comput.* **2019**, *14*, 1585–1597. [[CrossRef](#)]
29. He, Y.; Zhou, W.; Luo, F.; Gao, M.; Wen, J. Feature-based POI grouping with transformer for next point of interest recommendation. *Appl. Soft Comput.* **2023**, *147*, 110754. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.