Article

A Privacy-Preserving Time-Aware Method for Next POI Recommendation

Jianyong Fan 1, Chenxi Pan 2, Yue Geng 2 and Shuyu Li 2,*

1 School of Computer Science, Northwestern Polytechnical University, Xi’an 710072, China; fans@snnu.edu.cn
2 School of Computer Science, Shaanxi Normal University, Xi’an 710119, China; panchenxi@snnu.edu.cn (C.P.); gengyue@snnu.edu.cn (Y.G.)
* Correspondence: lishuyu@snnu.edu.cn

Abstract: Compared with traditional point-of-interest (POI) recommendation, next POI recommendation is more difficult and requires comprehensive consideration of users’ behavior patterns, spatial–temporal context, and other information. In addition, unreliable service providers may disclose the privacy of users when providing recommendation services. For next POI recommendation, a privacy-preserving time-aware recommendation method (PPTA-RM) is proposed in this paper. The PPTA-RM method is based on centralized differential privacy and combines coarse-grained recommendation with fine-grained recommendation. At the coarse-grained level, the users’ POI category preference is modeled by improved matrix factorization and predicted by singular spectrum analysis (SSA), and gradient perturbation is carried out during the matrix factorization process to protect the POI category preference of users. At the fine-grained level, the users’ POI preference is modeled and predicted by an improved hyperlink-induced topic search (HITS) algorithm, which considers the weighted combination of users’ social attributes and POI geographic distance attributes, and a privacy budget allocation strategy considering the visit count of POIs is designed to add Laplace noise to the check-in data of users. The experimental analysis on two real datasets shows that the proposed method improves F1-Score@10 by 0.4–21.8% under different privacy budgets, which validates that the proposed method holds the next POI recommendation accuracy better while preserving the user’s privacy.

Keywords: next POI recommendation; privacy preservation; differential privacy; time aware; check-in data

1. Introduction

Currently, many POI recommendation methods focus on learning the preferences of users based on their check-in data to predict the locations a user may visit. It is difficult for these methods to accurately predict locations that a user may be interested in during the next time interval, and flexible personalized POI recommendation under the spatial–temporal context of users is hard to achieve. In view of the above shortcomings, researchers have conducted various studies to address the problem of next POI recommendation. Among them, a typical method attempts to learn the behavior patterns (such as a sequence of behavior actions) of users, which can be used to discover the location transition pattern of users and predict their next new action (i.e., check-in behavior). However, due to the sparsity of check-in data, learning the location transitions of users is not a simple process. In addition, exploring the static preference of users from check-in data while ignoring the current spatial–temporal context may generate useless recommendations. For instance, it may be inappropriate to recommend a restaurant far away from a user’s current location or a bar during a user’s working hours. In reality, the preference of a user for POIs dynamically changes according to different contexts. A user’s check-in behavior is not only driven by behavior preference but also influenced by spatial–temporal information and other factors.
Therefore, next POI recommendation takes into account many contextual factors that are closely associated with the user’s behavior, including behavior patterns, time and geographical influences, etc. These factors are either discrete values (such as location transition patterns) or continuous values (such as geographical distance). Hence, next POI recommendation methods should be carefully designed to use these heterogeneous factors to predict user behavior in the next time interval. Furthermore, next POI recommendation methods are constructed on the basis of collecting user data, such as their relationships and the association between POIs and users. Third-party recommendation systems can extract sensitive user information and disclose their privacy for certain purposes. In addition, most recommendation models imply the personal information of users; for example, the latent feature matrix of the matrix factorization technique can directly infer users’ ratings on POIs. Therefore, next POI recommendation systems also face a high risk of privacy disclosure [1,2].

Based on the above analysis of current research, two main problems emerge: (1) how to integrate users’ behavior patterns and spatial–temporal attributes to learn their next POI preferences; and (2) how to provide users with personalized POI recommendation services without compromising user privacy.

To address the above problems of the next POI recommendation, a privacy-preserving time-aware recommendation method (PPTA-RM) is proposed in this paper. The PPTA-RM method assumes that the third-party data aggregator is credible and integrates both coarse-grained recommendation and fine-grained recommendation to predict the locations that users may visit in the next time slot, according to the users’ historical check-in data and current spatial–temporal context. At the coarse-grained level, the PPTA-RM method adopts matrix factorization to learn users’ behavior transition patterns and historical preferences and predict their POI category preferences in the next time slot by employing singular spectrum analysis (SSA). At the fine-grained level, the PPTA-RM method takes the weighted combination of users’ social attributes and POI geographic distance attributes into consideration and designs the improved hyperlink-induced topic search (HITS) model to provide POI recommendations for users. The quality of the next POI recommendation can be improved by combining coarse-grained recommendations with fine-grained recommendation results.

Briefly, the main contributions of our study are summarized as follows:

(1) To alleviate the sparsity problem of check-in data and improve the recommendation accuracy, the PPTA-RM method integrates coarse-grained recommendations with fine-grained recommendations. At the coarse-grained level, the users’ POI category preference is modeled by an improved matrix factorization and predicted by SSA. At the fine-grained level, the users’ POI preference is modeled and predicted by an improved HITS algorithm that considers the weighted combination of users’ social attributes and POI geographic distance attributes.

(2) The PPTA-RM method adopts double perturbation to preserve user privacy. Firstly, gradient perturbation is carried out during the matrix factorization process of coarse-grained recommendation to protect the POI category preference of users. Secondly, a privacy budget allocation strategy based on the visit count of POIs is designed and Laplace noise is added to the check-in data during the fine-grained recommendation stage.

(3) Experiments are conducted on both Gowalla and Foursquare datasets, and the results show the effectiveness of the PPTA-RM method.

The rest of the paper is arranged as follows: Section 2 discusses related studies. Section 3 reviews preliminary knowledge about differential privacy, matrix factorization, and the HITS model. Section 4 presents the design of the PPTA-RM method. Section 5 presents the experiments conducted on two real datasets and analyzes the results. Section 6 includes the conclusion and prospects of our study.
2. Related Work

2.1. POI Recommendation Methods

Compared with general recommendation tasks (such as movie recommendation), POI recommendation is highly dependent on geographical locations. Cai et al. [3] believed that users’ check-in frequency for a certain location was beneficial for better recommendation. The data of check-in frequency were treated as the weight attribute, and a weighted kernel density estimation algorithm considering spatial distance was designed to forecast the user’s POI preference. Zhao et al. [4] first designed a POI mining method, which integrated both geographical and sentimental factors of locations to obtain the proper POIs. Then, they presented a personalized recommendation model based on probabilistic matrix factorization. POI sentiment similarity and geographical distance between POIs and the user’s activity centers were used for POI rating evaluation. Lian et al. [5] proposed a collaborative recommendation approach for location-aware recommendation. Location information, user activities, and relationships were taken into consideration in the given approach. To solve the problems of data sparsity and cold start, Li et al. [6] classified the categories according to the friend relationship, based on which they calculated the corresponding POI ratings to improve the recommendation result. Using weekly intervals, Hosseini et al. [7] presented a probability model which adopted the time slot alignment between users and POIs to detect the time orientation of users.

Deep learning methods have also been successfully applied in the field of POI recommendation. Chen et al. [8] designed a POI recommendation framework to learn sophisticated sequential transformations. Factors of time and distance irregularity were used for the learning process, and dynamic weight was adopted for the prediction process. Qi et al. [9] proposed a POI category recommendation model named PPCM. The PPCM model first tried to discover similar users on different IoT platforms by locality-sensitive hashing in the time dimension and then considered group preferences (group similarity) to predict POI categories. Jeong and Kim [10] proposed a context-aware recommender method that considered the change in users’ preferences over time. The proposed method divided data into time units to account for temporality and adopted a preference transition matrix to detect preference change. A combination of an autoencoder and a neural network was integrated with a context-aware recommender system to solve the problem of data sparsity. For next POI recommendation in a traveling scenario, Liu et al. [11] presented an improved graph convolution network (GCN) model named ITGCN. The ITGCN model could find the implicit features between users and POIs and realized dynamic embedding of nodes (representing users and POIs). Additionally, a self-attention aggregator was designed to capture the sequence information and update the dynamic embedding of nodes from a spatiotemporal perspective. Wu et al. [12] proposed a hierarchical attention-based graph convolutional network (HAGCN) for next POI recommendation. The HAGCN model adopted a dual attention layer to adaptively learn the representations of POIs. Additionally, a dynamic preference estimation strategy was designed to extract the user preferences from the learned POIs’ presentations and previous check-in activities.

2.2. Differentially Private Recommendation

Dwork [13] put forward differential privacy theory, which realizes privacy preservation through data perturbation, that is, injecting noise into the original data. Nowadays, differential privacy has been successfully adopted for privacy-preserving recommendation systems. Wei et al. [14] presented a privacy-preserving scheme for trajectory community recommendation. The given scheme first generated noisy locations according to the locations of the real trajectory and then constructed the noisy trajectory, which was highly similar to the real one. The similarity of trajectories was measured by geographical distance. In response to the problems of recommendation accuracy and location privacy, Zhang et al. [15] proposed a location recommendation framework named PLORE, which characterized users’ sequence pattern via the additive Markov chain and computed probability results for better POI recommendation. To preserve the location privacy of users, they also implemented
a probabilistic differential privacy mechanism. For next POI recommendation in edge computing, Kuang et al. [16] first adopted a hidden Markov model (HMM) to represent the user’s sequence pattern. Then, they proposed a weighted noise-injection algorithm based on check-in center distance to preserve the user’s real location. Finally, a forward algorithm was designed to compute the probability of the user’s next movement.

Riboni et al. [17] provided POI recommendation for users with the same profiles according to their locations, interests, and preferences and applied differential privacy for counting users’ POI preferences and ensuring that the probability distribution of statistical data will not change significantly. For POI recommendation, Yin et al. [18] designed a location-sensitivity-based recommendation method. The given method first utilized trajectory and check-in data to classify location sensitivity levels and then allocated a different privacy budget according to the corresponding level and determined the amount of Laplace noise for location perturbation. Adopting the concept of population density, Huo et al. [19] converted the user’s location into a virtual circle with a dynamic radius and proposed a geographic location privacy-preservation algorithm (GLP) to realize $<r, h>$-privacy. By adding Laplace noise into the trust value between users, a friend relationship privacy-preservation algorithm (FRP) was presented. Finally, the GLP algorithm and FRP algorithm were integrated into a general recommendation framework to achieve a privacy-preserving POI recommendation.

3. Preliminaries

3.1. Differential Privacy

Differential privacy is a privacy protection technology proposed by Dwork. Briefly, differential privacy perturbs the original dataset by adding controllable noise, so that the noisy dataset can keep the original statistical attributes.

**Definition 1** ($\varepsilon$-Differential Privacy [13]). Let $M$ denote a randomized algorithm, and $S$ represents an arbitrary set of possible outputs of $M$. For any two neighboring datasets $D_1$ and $D_2$ (which differ in only one record), the algorithm $M$ is said to be $\varepsilon$-differentially private if:

$$\Pr[M(D_1) \in S] \leq e^\varepsilon \times \Pr[M(D_2) \in S]$$

where $\Pr[\cdot]$ denotes the probability of an event.

**Definition 2** (Sensitivity). For a query function $f : D \rightarrow \mathbb{R}^d$, the global sensitivity of $f$ is defined as:

$$\Delta f = \max_{D_1, D_2} \|f(D_1) - f(D_2)\|_1$$

where $D_1$ and $D_2$ are any two neighboring datasets, and $\|f(D_1) - f(D_2)\|_1$ denotes the Manhattan distance between $f(D_1)$ and $f(D_2)$.

**Definition 3** (Laplace Mechanism). For a query function $f : D \rightarrow \mathbb{R}^d$ with sensitivity $\Delta f$, if the output result satisfies Formula (3), algorithm $M$ is said to provide $\varepsilon$-differential privacy:

$$M = f(D) + \text{Laplace}(\Delta f / \varepsilon)$$

where the noise level is proportional to the global sensitivity $\Delta f$ and inversely proportional to the privacy parameter $\varepsilon$.

**Theorem 1** (Sequential Combination). Supposing there is a set of random algorithms $\{M_1, M_2, \ldots, M_n\}$, each $M_i (1 \leq i \leq n)$ satisfies $\varepsilon_i$-differential privacy on the dataset $D$. Then, the set of $M_i$ sequence privacy mechanisms provides $(\sum_{i=1}^{n} \varepsilon_i)$-differential privacy.

**Theorem 2** (Parallel Combination). Suppose the dataset $D$ can be divided into a series of independent and non-overlapped subsets $\{D_1, D_2, \ldots, D_n\}$ and there is a set of random algorithms
\{M_1, M_2, \ldots, M_n\}. If each \(M_i(1 \leq i \leq n)\) satisfies \(\varepsilon_i\)-differential privacy on \(D_i(1 \leq i \leq n)\), the set of randomized algorithms can provide \(\max\{\varepsilon_i\}\)-differential privacy on the dataset \(D\).

3.2. Matrix Factorization

Hypothetically, there are \(n\) users and \(m\) items; \(r_{ij}\) denotes the \(i\)-th user’s rating for the \(j\)-th item, and \(R = [r_{ij}]_{n \times m}\) denotes the user-item rating matrix. Matrix factorization [20] is adopted to predict the unobserved rating values in \(R\). To serve this purpose, \(R\) is decomposed into user profile matrix \(U \in \mathbb{R}^{n \times d}\) and item profile matrix \(V \in \mathbb{R}^{m \times d}\), where \(d\) is the number of latent features and \(d \ll \min\{n, m\}\). The corresponding optimization function is given as follows:

\[
\arg\min_{r_{ij} \neq 0} \sum_{r_{ij} \neq 0} (r_{ij} - u_i v_j^\top)^2 + \lambda_u \sum_{i=1}^n \|u_i\|^2 + \lambda_v \sum_{j=1}^m \|v_j\|^2
\]

(4)

where \(u_i\) is the \(i\)-th row vector of \(U\), \(v_j\) is the \(j\)-th row vector of item profile matrix \(V\), \(v_j^\top\) is the transposed vector of \(v_j\), and \(\lambda_u\) and \(\lambda_v\) are hyper parameters with positive value. The first term \(\sum_{r_{ij} \neq 0} (r_{ij} - u_i v_j^\top)^2\) is a loss function that measures the calculating deviation, and the last two terms \(\lambda_u \sum_{i=1}^n \|u_i\|^2\) and \(\lambda_v \sum_{j=1}^m \|v_j\|^2\) are calculating variance and determine whether or not the model is overfitting. The missing/unobserved rating value can be computed by \(u_i v_j^\top\). Stochastic gradient descent (SGD) or alternating least squares (ALS) are two common strategies for minimizing the above optimization function.

3.3. HITS

Kleinberg proposed the HITS algorithm for network search problems [21], aiming at extracting information from link structure. In the algorithm, each web page has two attributes: hub value and authority value, which are recursively defined. Therefore, hub value and authority value show a mutually reinforcing relationship, as shown in Figure 1.

![Figure 1. Hubs and authorities in HITS algorithm.](image)

If \(A\) is the adjacency matrix of the above graph, \(a\) is the component vector of authority value, and \(h\) is the component vector of hub value, the updated operation of \(a\) and \(h\) is given as follows [21]:

\[
\begin{cases}
  a = A^\top \cdot h \\
  h = A \cdot a
\end{cases}
\]

(5)

4. The PPTA-RM Method

4.1. Overview

In the PPTA-RM method, the threat model is briefly described as follows: there are mobile users, the data aggregator, and the recommendation system. Mobile users are responsible for providing check-in data and will not collude with each other. The data
aggregator is responsible for collecting original data from mobile users and is assumed to be credible. The recommendation system is honest but curious, that is, it can honestly perform its duties and will not collude with any other malicious entity. After data collecting by the data aggregator, the differentially private POI recommendation is achieved in the recommendation system to avoid the privacy leakage of users. However, an adversary or the recommendation system may try to perform an inference attack to analyze the sensitive information of users exposed by the recommendation service for its benefit.

The PPTA-RM method adopts the strategy of combining coarse-grained recommendation with fine-grained recommendation. The components of the PPTA-RM method are shown in Figure 2.

![Figure 2. The framework of the PPTA-RM method.](image)

For coarse-grained recommendation, the privacy-preserving category preference prediction algorithm (PrivCP) based on matrix factorization is designed to recommend the POI categories that users prefer by replacing the traditional user-POI check-in matrix with the user-category check-in matrix. In addition, gradient perturbation is carried out during the matrix factorization process to protect the POI category preference of users. For fine-grained recommendation, the privacy-preserving weighted HITS-based recommendation algorithm (PrivWHBR) is presented, which exploits users’ check-in data and considers the weighted combination of users’ social attributes and POI geographic distance attribute to achieve POI recommendation. A privacy budget allocation strategy based on the visit count of POIs is designed and Laplace noise is injected into the check-in data of users. Finally, by combining the prediction results of the above two algorithms, accurate next POI recommendation can be achieved. In the rest of this section, Section 4.2 describes the PrivCP algorithm, Section 4.3 presents the PrivWHBR algorithm, and Section 4.4 demonstrates a privacy analysis of the PPTA-RM method.
4.2. PrivCP Algorithm

The PrivCP algorithm aims to determine the Top $- N_1$ POI categories that users may be interested in in the next time slot and simultaneously protect users’ preference data for the POI categories. The PrivCP algorithm mainly consists of three steps: (1) matrix construction; (2) privacy-preserving POI category preference modeling; and (3) preference category prediction.

Taking time factors into consideration, the check-in data are firstly divided into $T$ time slots. For the check-in data in each time slot, Step 1 constructs a user–category matrix and a category–category matrix and normalizes the values of these two matrices, respectively. In this step, the sparsity of check-in data is alleviated by introducing a location category attribute. Step 2 performs privacy-preserving category preference modeling based on matrix factorization, obtains the user feature matrix and category feature matrix in each time slot, and adds Laplace noise during the training process to protect the user’s category preference. Step 3 calculates the prediction matrices of the user feature matrix and category feature matrix in the $T + 1$ time slot based on SSA, completes the prediction matrices, and finally determines $N_1$ POI categories that users may be interested in in the $T + 1$ time slot.

4.2.1. Matrix Construction

In real life, there are numerous POIs, but a user can only visit a small number of locations and cannot rate every item. The sparsity of the rating matrix may lead to overfitting in the recommendation algorithm and reduce the recommendation quality. Hence, adopting the concept of transfer learning, the PrivCP algorithm extracts the location category data from the users’ check-in history as auxiliary information, which alleviates the influence of the sparsity of the rating matrix to a certain extent.

Specifically, the PrivCP algorithm needs to extract two matrices from users’ check-in history: the user–category matrix and the category–category matrix. The PrivCP algorithm extracts the visit count of a user to each category and constructs the user–category matrix, which reflects the preference relationship between user and categories; by using a first-order Markov chain, the PrivCP algorithm extracts the number of transitions between two categories and captures the transition pattern between categories. The PrivCP algorithm can learn the relationship between users and categories through the user–category matrix and category–category matrix in the process of category preference modeling.

User–category matrix: suppose that the users’ check-in matrix for categories can be depicted by a matrix $P \in \mathbb{R}^{n \times m}$, where $n$ is the number of users, $m$ is the number of categories of POIs, and $p_{ij}$ is the check-in frequency of user $i$ for POI category $j$.

Category–category matrix: supposing that the appearance of the next POI category depends on the category of POI that the user currently visited, the transition relationship between two categories of POIs can be modeled through the category–category matrix. This matrix can be depicted by a matrix $Q \in \mathbb{R}^{m \times m}$, where $m$ is the number of POI categories, and $q_{kj}$ is the number of transitions from category $k$ to category $j$, representing the correlation between two categories.

4.2.2. Privacy-Preserving POI Category Preference Modeling

With the help of the user–category matrix and category–category matrix, the PrivCP algorithm can learn two latent matrices—the user latent feature matrix and category latent feature matrix—and adds Laplace noise to the user latent feature matrix during the training process, so as to protect the POI category preference data of users.

The PrivCP algorithm adopts a matrix factorization strategy based on category sharing, as shown in Figure 3. $P \in \mathbb{R}^{n \times m}$ is the user–category matrix, and $Q \in \mathbb{R}^{m \times m}$ is the category–category matrix, where $n$ is the number of users, and $m$ is the number of categories of POIs. $U \in \mathbb{R}^{n \times d}$ is the user latent feature matrix, $U = \{u_i \in \mathbb{R}^d\}_{1 \leq i \leq n}$, in which $u_i$ is the latent feature vector of user $i$; $C \in \mathbb{R}^{m \times d}$ is the category latent feature matrix, $C = \{c_j \in \mathbb{R}^d\}_{1 \leq j \leq n}$.
in which \( c_j \) is the latent feature vector of category \( j \); \( C^\top \) is the transpose matrix of \( C \); and \( d \) is the number of latent features.

![Diagram](image)

**Figure 3.** Matrix factorization strategy based on category sharing.

As can be seen from Figure 3, the category latent feature matrix \( C \) is shared by the user-category matrix \( P \) and category–category matrix \( Q \) and plays a connecting role. Therefore, the number of user visits to categories and the transition patterns between categories will be integrated into the learning process.

In this way, the loss function of privacy-preserving category preference modeling is given as follows:

\[
L = \left\| P - UC^\top \right\|^2 + \left\| Q - CC^\top \right\|^2 + \lambda (\|U\|^2 + \|C\|^2) \tag{6}
\]

where the first two terms of Formula (6) measure the calculating deviation, and \( \lambda \) is the regularization term to prevent the model from overfitting.

The PrivCP algorithm exploits the SGD strategy to minimize Formula (6). However, directly using visiting counts and transition patterns to optimize the above loss function may lead to the violation of the user’s category-preference privacy. Thus, the PrivCP algorithm adopts gradient perturbation to protect the user’s category preference from being leaked. As shown in Formulas (7) and (8), the partial derivatives of parameter \( u_i \) and \( c_j \) are firstly obtained, respectively, as:

\[
\frac{\partial L}{\partial u_i} = -2 \sum_{j=1}^{m} (p_{ij} - u_i c_j^\top) c_j + 2\lambda u_i \tag{7}
\]

\[
\frac{\partial L}{\partial c_j} = -2 \sum_{i=1}^{n} (p_{ij} - c_j u_i^\top) u_i - 2 \sum_{k=1}^{m} (q_{jk} - c_j c_k^\top) c_k + 2\lambda c_j \tag{8}
\]

In order to protect the category preferences of users, Laplace noise is added to the error \( \tilde{e}_{ij} = p_{ij} - u_i c_j^\top \) in Formula (7) when learning the user latent feature vector \( u_i \), as shown in Formula (9):

\[
\tilde{e}_{ij} = p_{ij} - u_i c_j^\top + \text{Lap}(\text{iter} \cdot \Delta p / \varepsilon_1) \tag{9}
\]

where \( \text{Lap}(\cdot) \) represents the added noise satisfying Laplace distribution, \( \text{iter} \) is the number of iterations, \( \Delta p = |p_{\text{max}} - p_{\text{min}}| \) is the sensitivity of the algorithm, \( p_{\text{max}} \) is the maximum value of elements, \( p_{\text{min}} \) is the minimum value of elements in the user–category matrix \( P \), respectively, and \( \varepsilon_1 \) is privacy budget. Therefore, Formula (8) can be modified as follows:

\[
\frac{\partial L}{\partial u_i} = -2 \sum_{j=1}^{m} \tilde{e}_{ij} c_j + 2\lambda u_i \tag{10}
\]
Then, the parameters are learned and updated according to Formulas (8) and (10):

\[
u_i \leftarrow u_i + \alpha \frac{\partial L}{\partial u_i}
\]

(11)

\[
c_j \leftarrow c_j + \alpha \frac{\partial L}{\partial c_j}
\]

(12)

where \(\alpha\) is learning rate. Formulas (11) and (12) are repeatedly executed, until the error of the loss function is less than the threshold value or the maximum number of iterations is reached. Then, the learning process is stopped, and the user latent feature matrix \(U\) and category latent feature matrix \(C\) for each time slot are consequently obtained.

4.2.3. Preference Category Prediction

After the first two steps above, the user latent feature matrix \(U_t(1 \leq t \leq T)\) and category latent feature matrix \(C_t(1 \leq t \leq T)\) can be obtained. The purpose of Step 3 is to predict the \(U_{T+1}\) and \(C_{T+1}\) through the previous \(T\) latent feature matrix sequences based on SSA and complete the matrices, finally achieving \(Top - N_1\) category prediction.

SSA mainly consists of two stages: model fitting and model prediction. In this section, the user latent feature matrix sequence \(X = U_t(1 \leq t \leq T)\) is used to illustrate the two stages.

In the model fitting stage, a proper window length is firstly selected, and then the original time series \(X\) is transformed into the trajectory matrix \(Y\), as shown in Formula (13), where \(L(2 \leq L \leq T/2)\) represents the window length, \(K(K = T - L + 1)\) is the lag parameter, and \(T\) is the length of \(X\).

\[
Y = \begin{bmatrix}
U_1 & U_2 & \cdots & U_L \\
U_2 & U_3 & \cdots & U_{L+1} \\
\vdots & \vdots & \ddots & \vdots \\
U_K & U_{K+1} & \cdots & U_T
\end{bmatrix}
\]

(13)

Then, singular value decomposition is performed on the trajectory matrix \(Y\), as shown in Formula (14):

\[
Y = \sum_{i=1}^{d} Y_i = \sum_{i=1}^{d} E_i \sqrt{\tau_i} F_i
\]

(14)

where \(d = \text{rank}(Y)\) is the rank of the trajectory matrix \(Y\), \(\tau_1 \cdots \tau_d\) represents the singular value of matrix \(Y\), and \(E_1 \cdots E_d\) and \(F_1 \cdots F_d\) are the left singular vector and right singular vector of matrix \(Y\), respectively. The tuple \((\tau_i, E_i, F_i)\) is called the \(i\)-th feature triad of singular value decomposition.

Finally, the signals represented by each eigenvalue are analyzed and combined, and a new time series is reconstructed by the time empirical orthogonal function and time principal component, which is given in Formula (15):

\[
y_i = \left\{ \begin{array}{ll}
\frac{1}{T} \sum_{j=1}^{L} a_{i-j} E_j & 1 \leq i \leq L - 1 \\
\frac{1}{T} \sum_{j=1}^{L} a_{i-j} E_j & L \leq i \leq T - L + 1 \\
\frac{1}{T} \sum_{j=i-T+L}^{L} a_{i-j} E_j & T - L + 2 \leq i \leq T
\end{array} \right.
\]

(15)

where \(y_i\) is the \(i\)-th column of matrix \(Y\), and \(E_j\) is the \(j\)-th left singular vector, \(a_i = \sqrt{\tau_i} F_i\). In this way, the sum of all the reconstructed series is equal to the original series.

In the model prediction stage, linear recursion is adopted to calculate \(U_{T+1}\) with the linear combination of the previous data in the window, as shown in the following formula:

\[
U_{T+1} = \sum_{k=T-L+1}^{L} a_k U_k
\]

(16)
Similarly, category feature matrix $C_{T+1}$ can be obtained as shown in the following formula:

$$C_{T+1} = \sum_{k=T-L+1}^{L} a_k C_k$$  \hspace{1cm} (17)

Finally, according to the user latent feature matrix $U_{T+1}$ and category latent feature matrix $C_{T+1}$, and considering the current location of the user, the category candidate set that the user may be interested in in the $T+1$ time slot is predicted. Specifically, the prediction value of user $i$ for category $k$ is calculated as follows:

$$\text{pref}^k_i = u_i c_k^\top + c_j c_k^\top$$  \hspace{1cm} (18)

where $u_i$ is the $i$-th row vector of matrix $U_{T+1}$, and $c_j$ and $c_k$ are the $j$-th row and $k$-th row vectors of matrix $C_{T+1}$, respectively. The first term $u_i c_k^\top$ denotes the long-term preference of user $i$ for category $k$, and the second term $c_j c_k^\top$ represents the transition preference from category $j$ to category $k$. By calculating the sum of the two preferences and then arranging them in descending order, $\text{Top} - N_1$ POI categories that the user may be interested in in the $T+1$ time slot can be predicted.

4.3. PriWHBR Algorithm

The purpose of the PriWHBR algorithm is to recommend POIs with higher authority value (determined by the user’s hub value) to the user. In social networks, the users’ social relationship and the geographical location of POIs will also affect the POI that users may visit. However, the original HITS model is used to solve the problem of network search without considering factors such as social relationship and geographical location, and it cannot be used directly.

To solve the above problem, the PriWHBR algorithm improves the original HITS algorithm by using the social relationship between users and the geographical relationship between POIs as the weight values of corresponding connections in the social network.

The network model used in the PriWHBR algorithm is shown in Figure 4. This model is divided into two layers. The upper undirected graph is a social relationship graph, vertices represent users, and edges symbolize social relationships among users. If the set of edges is $E^f$ and there exists a social relationship between user $i$ and user $k$, edge $e_{ik} \in E^f$ exists. The lower undirected graph is a geographical distance graph, with vertices representing POIs and edges representing the geographical distance relation between POIs. If the set of edges is $E^g$, the cosine similarity $\text{sim}_{xy}$ based on geographical coordinates of POI $l_x$ and $l_y$ is calculated, and if $\text{sim}_{xy} > 0$, edge $e_{xy} \in E^g$. The check-in data of a user are regarded as the directed connections between the user and POIs (dotted line with arrow in the figure), connecting the upper and lower layers. If $E^c$ represents the set of directed check-in edges and user $i$ has visited POI $l_x$, edge $e_{ix} \in E^c$.

The PriWHBR algorithm assumes that each user owns a hub value, and the larger the value is, the stronger the visit capability of the user, namely, the larger the count of POIs that the user has visited. Each POI has an authority value, and the higher the value is, the better the quality of the POI, that is, the greater the probability that it will be visited by users. In other words, a person who has checked in to numerous POIs in a certain region can have a thorough view of that region, while POIs visited by a knowledgeable person may be superior POIs. Hub values of users and authority values of POIs influence each other in recursion.
Comprehensively considering the social attributes of users, check-in data, and geographical distance of POIs, the recommendation model of the PrivWHBR algorithm can be expressed as:

\[
\begin{align*}
    a_{pOIs} &= \beta W_u^T h_{users} + (1 - \beta) W_{poi}a_{pOIs} \\
    h_{users} &= \gamma W_u h_{users} + (1 - \gamma) W_{u-poi} a_{pOIs}
\end{align*}
\]

(19)

The authority value \( a_{pOIs} \) of a POI is calculated by the weighted sum of two parts: the first part describes the hub values of users who have checked in to this POI, and the second part depicts the authority values of the POIs near to it. The hub value \( h_{users} \) of a user is also computed by weighted sum of two parts: the first part is hub values of their friends, and the second part denotes the authority values of the POIs that they have checked in to, where \( 0 < \beta < 1 \) and \( 0 < \gamma < 1 \) are weight factors, and \( W_u, W_{poi}, \) and \( W_{u-poi} \) are the user–user adjacency matrix, POI–POI adjacency matrix, and user–POI adjacency matrix, respectively, which are given in Formula (20) to Formula (22):

\[
W_u(i, j) = \begin{cases} 
    \delta s_{ij} / \sum_{k=1}^{n} s_{ik}, & e_{ij} \in E^f \text{ and } n_i^f > 0 \\
    0, & \text{otherwise}
\end{cases}
\]

(20)

where \( 0 < \delta < 1 \) is the weight factor; if user \( i \) has a social relationship with user \( j, s_{ij} = 1, \) otherwise \( s_{ij} = 0; \) \( n_i^f \) is the number of POIs that user \( i \) has checked in to.

\[
W_{poi}(x, y) = \frac{\text{sim}_{xy} \log \text{sim}_{xy}}{(\sum_{y=1}^{m} \text{sim}_{xy} \log \text{sim}_{xy}),} \quad e_{xy} \in E^g
\]

(21)

Here, \( m \) is the number of POIs, and \( \text{sim}_{xy} \) is the cosine similarity between POI \( l_x \) and \( l_y \) based on geographical coordinates.

\[
W_{u-poi}(i, x) = \begin{cases} 
    (1 - \delta) u_{ipix} \log u_{pix} / (\sum_{y=1}^{n} u_{p_y} \log u_{p_y}), & e_{ix} \in E^c \text{ and } n_i^e > 0 \\
    u_{ipix} \log u_{pix} / (\sum_{y=1}^{n} u_{p_y} \log u_{p_y}), & e_{ix} \in E^e \text{ and } n_i^f = 0
\end{cases}
\]

(22)

Here, \( u_{ipix} \) represents the preference of user \( i \) for POI \( l_x, \) which is expressed by visit count; \( n_i^f \) indicates the amount of friends that user \( i \) owns.

In addition, for a certain POI, larger visit counts of users often represent greater preference of users. Attackers can learn users’ preferences by analyzing the visit count of
specific POIs. In other words, matrix $W_{u-poI}$ regarding a user’s check-in data may reveal their preferences. Thus, the PrivWHBR algorithm designs a privacy budget allocation strategy according to the visit count of POIs to effectively protect matrix $W_{u-poI}$. Specifically, if a POI has larger visit counts, the allocated privacy budget is smaller and the added Laplace noise is larger, which means stronger privacy preservation.

Therefore, the perturbed user–POI adjacency matrix $W_{u-poI}$ is given as follows:

$$
\tilde{W}_{u-poI}(i, x) = W_{u-poI}(i, x) + Lap(\Delta W / \epsilon_x)
$$

where $Lap(\cdot)$ is the Laplace noise; $\Delta W = |W_{u-poI}^{max} - W_{u-poI}^{min}|$ is the sensitivity; $W_{u-poI}^{max}$ and $W_{u-poI}^{min}$ are the maximum value and the minimum value in $W_{u-poI}$, respectively; $count(l_x)$ is the visit count of POI $l_x$; and $\epsilon_x$ is the privacy budget and related to the sensitivity of POI $l_x$. $\epsilon_x$ can be calculated as follows:

$$
\epsilon_x = (SL(l_x) / \sum_{y=1}^{m} SL(l_y)) \epsilon_2
$$

where $\epsilon_2$ is the total privacy budget of the PrivWHBR algorithm, $SL(l_x)$ is the attenuation function, and $\sigma$ is the attenuation constant.

In general, the PrivWHBR algorithm can recommend POIs with high authority values to users.

Finally, by combining the result of the PrivCP algorithm, the $Top - N_2$ POIs with the highest scores in $Top - N_1$ categories will be returned to the user as a recommendation list for the $T + 1$ time slot.

4.4. Privacy Analysis

In this section, the proving process of privacy analysis is divided into three steps: in the first two steps, we prove the PrivCP algorithm and the PrivWHBR algorithm satisfy differential privacy requirements, respectively; in the last step, we prove the PPTA-RM method satisfies differential privacy requirements.

(1) The PrivCP algorithm satisfies $\epsilon_1$-differential privacy

Assuming that $D$ and $D'$ are adjacent datasets, $\tau$ is the output of the algorithm, $\eta_{xy}$ is the added Laplace noise, namely $\eta_{xy} \sim Lap(\iter \cdot \Delta p / \epsilon_1)$, and $\iter$ is the number of iterations, the final user feature matrix $U$ generated by the algorithm satisfies $\epsilon_1$-differential privacy.

Proof. In an iteration,

$$
\Pr(U = \tau | D) = \frac{\Pr(U = \tau | D) \cdot \Pr(U = \tau | D)}{\Pr(U = \tau | D)}
$$

where $\Pr(U = \tau | D)$ is the visit count of POI $l_x$; and $\epsilon_x$ is the privacy budget and related to the sensitivity of POI $l_x$. $\epsilon_x$ can be calculated as follows:

$$
\epsilon_x = (SL(l_x) / \sum_{y=1}^{m} SL(l_y)) \epsilon_2
$$

where $\epsilon_2$ is the total privacy budget of the PrivWHBR algorithm, $SL(l_x)$ is the attenuation function, and $\sigma$ is the attenuation constant.

In general, the PrivWHBR algorithm can recommend POIs with high authority values to users.

Finally, by combining the result of the PrivCP algorithm, the $Top - N_2$ POIs with the highest scores in $Top - N_1$ categories will be returned to the user as a recommendation list for the $T + 1$ time slot.

4.4. Privacy Analysis

In this section, the proving process of privacy analysis is divided into three steps: in the first two steps, we prove the PrivCP algorithm and the PrivWHBR algorithm satisfy differential privacy requirements, respectively; in the last step, we prove the PPTA-RM method satisfies differential privacy requirements.

(1) The PrivCP algorithm satisfies $\epsilon_1$-differential privacy

Assuming that $D$ and $D'$ are adjacent datasets, $\tau$ is the output of the algorithm, $\eta_{xy}$ is the added Laplace noise, namely $\eta_{xy} \sim Lap(\iter \cdot \Delta p / \epsilon_1)$, and $\iter$ is the number of iterations, the final user feature matrix $U$ generated by the algorithm satisfies $\epsilon_1$-differential privacy.

Proof. In an iteration,
(2) The PrivWHBR algorithm meets max\{\epsilon_x\}-differential privacy

For any two adjacent datasets \(D\) and \(D'\), \(\omega\) is the output of the algorithm and \(\eta_{ix}\) is the added Laplace noise, namely \(\eta_{ix} \sim \text{Lap}(\Delta W/\epsilon_x)\); therefore, the user–POI adjacency matrix \(\tilde{W}_{u-poi}\) generated by the algorithm satisfies max\{\epsilon_x\}-differential privacy.

Proof.

For any element \(\tilde{W}_{u-poi}(i, x)\) of the user–POI adjacency matrix \(\tilde{W}_{u-poi}\), since the elements in matrix \(\tilde{W}_{u-poi}\) are disjointed, the PrivWHBR algorithm satisfies max\{\epsilon_x\}-differential privacy on the basis of parallel combination theorem. \(\square\)

(3) The PPTA-RM method satisfies \(\epsilon\)-differential privacy

Because the PrivCP algorithm and the PrivWHBR algorithm can be executed in parallel, and the data processed by the two algorithms belong to disjointed datasets, the PPTA-RM method satisfies max\{\epsilon_1, \max \{\epsilon_x\}\}-differential privacy on the basis of parallel combination theorem.

\[
\epsilon_1 + \sum_{i=1}^{m} \epsilon_x = \sum_{i=1}^{m} S_l(x) \leq \epsilon_2
\]

If \(\epsilon_1 = \epsilon_2 = \epsilon\), the PPTA-RM method satisfies \(\epsilon\)-differential privacy.

5. Experiments

5.1. Experimental Settings

We implemented our method in Python programming language (version 3.6.9) and conducted experiments on a machine with Intel (R) Core i5-6500/3.2 GHz/64GB hardware configuration and a 64-bit Windows 10 operating system.

In the experiment, we performed an experimental study by employing two real datasets: Gowalla [22] and Foursquare [19]. Since these two datasets are sparse, for better experimental results, we selected the check-in records of active users within a certain period. Taking the Foursquare dataset as an example, the data we selected were sourced from between March 2010 and December 2011 in different urban areas. Following the common practice in the literature for POI recommendations, we removed duplicate check-in records and users with less than five check-in records. Table 1 describes the statistical data of the two datasets.

Table 1. Statistics of datasets.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Users</th>
<th>POIs</th>
<th>Check-in Records</th>
<th>Social Relations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gowalla</td>
<td>18,737</td>
<td>32,510</td>
<td>1,982,035</td>
<td>46,001</td>
</tr>
<tr>
<td>Foursquare</td>
<td>24,941</td>
<td>43,593</td>
<td>1,278,274</td>
<td>120,883</td>
</tr>
</tbody>
</table>

5.2. Experimental Results

To evaluate the performance of the PPTA-RM method, we applied three widely used performance metrics in recommendation approaches: Precision, Recall, and F1-Score.

Precision: the ratio of the number of actually visited POIs in the recommended POI is set to the number of recommended POIs, which is given as follows:

\[
\text{precision}@N = \frac{\sum_{i \in U} |\text{TopN}_i \cap \text{L}_i|}{\sum_{i \in U} |\text{TopN}_i|}
\]  

Recall: the ratio of the number of recommended POIs in the actually visited POI is set to the number of actually visited POIs, which is given as follows:

\[
\text{recall}@N = \frac{\sum_{i \in U} |\text{TopN}_i \cap \text{L}_i|}{\sum_{i \in U} |\text{L}_i|}
\]
where \( U \) depicts the user set, \( TopN_i \) is the POI recommendation set for user \( i \) in the test set, and \( L_i \) is the POI recommendation set for user \( i \) in the training set.

F1-Score indicates the comprehensive recommendation quality based on precision and recall. If the value of F1-Score is larger, it means that the recommended quality is higher.

\[
F1 - Score = 2 \times \text{Precision} \times \text{Recall} / (\text{Precision} + \text{Recall})
\]  

(28)

To analyze the performance of the PPTA-RM method, we compared our proposal with the following alternative approaches:

- **TRS**: the non-privacy-preserving version of the PPTA-RM method.
- **DP-SVD**: a classic recommendation method based on matrix factorization, which implements gradient perturbation during the process of SGD.
- **PMLS**: Yin et al. [18] proposed a differentially private recommendation method which adopts the concept of location sensitivity. The PMLS method classifies sensitivity according to the hierarchy of locations in the prefix tree and allocates different privacy budgets for corresponding sensitivity levels.
- **GLP_FRP**: Huo et al. [19] put forward the geographic location privacy-preserving algorithm (GLP) and friend relationship privacy-preserving algorithm (FRP) and integrated both the GLP algorithm and FRP algorithm into a recommendation system.

The PPTA-RM method has some predefined parameters, which are described and set out in Table 2. The number of time slots is \( T = 42 \) because each day is divided into six time segments (i.e., 0:00–7:00, 7:00–9:00, 9:00–12:00, 12:00–14:00, 14:00–18:00, and 18:00–0:00), which meets the work and life patterns of most people. Therefore, each week is divided into 42 time slots identified by slot ID.

**Table 2.** Predefined parameters of the PPTA-RA method.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Meaning</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( d )</td>
<td>The number of latent features</td>
<td>20</td>
</tr>
<tr>
<td>( \alpha )</td>
<td>Learning rate</td>
<td>0.01</td>
</tr>
<tr>
<td>( iter_{max} )</td>
<td>The maximum number of iterations</td>
<td>30</td>
</tr>
<tr>
<td>( N_1 )</td>
<td>The number of recommended categories</td>
<td>15</td>
</tr>
<tr>
<td>( T )</td>
<td>The number of time slots</td>
<td>42</td>
</tr>
<tr>
<td>( \beta, \gamma )</td>
<td>Weight factor</td>
<td>0.375</td>
</tr>
<tr>
<td>( \delta )</td>
<td>Weight factor</td>
<td>0.5</td>
</tr>
</tbody>
</table>

We conducted comparative experiments from two aspects. The first experiment investigated the impact of varying privacy budgets on the performance of the above recommendation methods. The second experiment validated the impact of varying \( N_2 \) (the number of POIs recommended to users) on the performance of above methods. We adopted a five-fold cross-validation strategy to carry out the experiments and used the mean values of precision and recall as the final output.

### 5.2.1. Impact of Varying Privacy Budget

The privacy budget \( \varepsilon \) determines the degree of privacy protection. In this experiment, we investigated the performance of the above methods when varying the privacy budget. We set \( N_2 = 10 \) and varied the privacy budget from 0.1 to 1 with the step of 0.1.

Figure 5 shows the trends of the precision of the above methods for the Gowalla and Foursquare datasets, and Figure 6 shows the trend of recall of the above methods for these two datasets. Except for TRS, both the precision and recall of all the other four methods
showed a steady upward trend as the privacy budget increased. This trend is predictable, since added noise decreases as the privacy budget increases.

![Figure 5](image1.png)  
**Figure 5.** Comparison of precision under different privacy budgets. (a) Gowalla dataset; (b) Foursquare dataset.

![Figure 6](image2.png)  
**Figure 6.** Comparison of recall under different privacy budgets. (a) Gowalla dataset; (b) Foursquare dataset.

As shown in Figure 5, the precision value of the PMLS method is the smallest, while the precision values of the other three methods are close. The precision value of the PPTA-RM method is much higher than that of the DP-SVD method and GLP_FRP method when the privacy budget is small. In general, the proposed method achieves better recommendation accuracy for these two datasets.

As can be seen from Figure 6, except for the DP-SVD method, the other three privacy-preserving methods maintain similar recall values. From the view of recall, when \( \epsilon \geq 0.9 \), the GLP_FRP method is slightly better than the PPTA-RM method for the Gowalla dataset. When \( \epsilon \geq 0.7 \), the GLP_FRP method is slightly better than the PPTA-RM method for the Foursquare dataset, but when \( \epsilon \leq 0.4 \), the PPTA-RM method has a higher recall value, which also indicates that the PPTA-RM method can maintain high recommendation quality when the privacy budget is small.

The F1-Score comparison of the above methods is shown in Figure 7. Since the user’s check-in data are sparse, this has a certain impact on F1-Score. The F1-Score of each privacy-preserving method grows as the privacy budget increases. With the exception of the F1-Score of the GLP_FRP method being very slightly higher than that of the PPTA-RM
method when the privacy budget is 1, the F1-Score of the PPTA-RM method is better than the other three privacy-preserving methods as a whole. Compared with the DP-SVD, PMLS, and GLP_FRP methods, the PPTA-RM method improved the F1-Score@10 on the Gowalla dataset by 10.4%–15.6%, 5.8%–10.9%, and 0.4%–5.6%, respectively, and improved the F1-Score@10 on the Foursquare dataset by 8.3%–21.8%, 2.8%–12.5%, and 0.8%–11.2%, respectively.

![Figure 7. Comparison of F1-Score under different privacy budgets. (a) Gowalla dataset; (b) Foursquare dataset.](image)

### 5.2.2. Impact of Varying $N_2$

In the setting of this experiment, the privacy budgets of all the privacy-preserving methods were fixed to 0.5. For the Gowalla and Foursquare datasets, Figures 8 and 9 demonstrate the precision and recall of the above five methods with different values of $N_2 (N_2 = \{5, 10, 20, 50\})$, respectively. As shown in these two figures, the precision value of each method shows a decreasing tendency as the value of $N_2$ increases, while the recall value of each method presents the opposite tendency.

![Figure 8. Comparison of precision under different $N_2$ values. (a) Gowalla dataset; (b) Foursquare dataset.](image)
As shown in Figure 8, the precision values of both the PPTA-RM and DP-SVD methods at \( N_2 = 20 \) and \( N_2 = 50 \) are significantly higher than that of the PMLS and GLP_FRP methods for these two datasets, implying that these two methods can maintain better recommendation accuracy when the number of POIs recommended is large. A possible reason may be that matrix-factorization-based methods are better at dealing with sparse check-in data.

As shown in Figure 9, the recall values of the PPTA-RM method and the other three privacy-preserving methods are at a similar level, and the recall value of the PPTA-RM method is slightly lower than that of the GLP_FRP method when \( N_2 = 50 \) for both the Gowalla and Foursquare datasets. One possible reason is that the PPTA-RM method adopts two-stage prediction, which first predicts the categories that users may be interested in and then makes accurate POI recommendations in these categories. When numerous POIs exist, the number of categories limits the number of POIs that can be recommended. Relatively speaking, the PPTA-RM method is more suitable for scenarios with small values of \( N_2 \), which is also consistent with real-life situations since more POI recommendation lists are more prone to information redundancy. On the whole, the proposed method shows better recommendation quality than the other three privacy-preserving methods.

6. Conclusions

Trying to solve the problem of next POI recommendation under the condition of privacy preservation, a time-aware recommendation method named PPTA-RM is presented in this paper.

The advantages of the PPTA-RM method can be summarized as follows: the PPTA-RM method takes both users’ preferences and users’ time requirements into consideration. Two algorithms named PrivCP and PrivWHBR were designed for coarse-grained-level POI category recommendation and fine-grained-level POI recommendation. The first models the time-evolving POI category preference of users based on matrix factorization and SSA, and Laplace noise is added to the gradients during the iterative process of matrix factorization to protect the privacy of the user’s category preference. The second combines the user’s social attributes and the geographical distance of POIs to recommend POIs. Different noise is added to check-in data according to the different visit counts of POIs.

Some drawbacks also exist in the proposed method: The PrivCP algorithm adopts the average allocation strategy for gradient perturbation; if too many iterations exist, the amount of added noise is large, and the learning effect will be greatly reduced. One of our future research directions is to investigate the self-adaptive allocation strategy of the privacy budget. Additionally, the recommendation performance of the proposed method will sharply decrease when facing a massive number of POIs; thus, distributed training

![Figure 9. Comparison of recall under different N2 values. (a) Gowalla dataset; (b) Foursquare dataset.](image-url)
scenarios with local differential privacy could be considered to perform privacy-preserving next POI recommendations and reduce the computational complexity of the method.

**Author Contributions:** Conceptualization, S.L.; methodology, J.F.; software, C.P.; validation, C.P. and Y.G.; formal analysis, J.F.; investigation, C.P.; resources, C.P.; data curation, Y.G.; writing—original draft preparation, J.F.; writing—review and editing, S.L.; visualization, Y.G.; supervision, S.L.; project administration, J.F. and S.L.; funding acquisition, S.L. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by “Key Research and Development Program of Shaanxi Province”, grant number 2021GY-090.

**Data Availability Statement:** The data presented in this study are available in references [19,22].

**Conflicts of Interest:** The authors declare that they have no conflict of interest to report regarding the present study.

**References**

12. Wu, J.; Jiang, S.; Shi, L. A Next POI Recommendation Based on Graph Convolutional Network by Adaptive Time Patterns. *Electronics* 2023, 12, 1241. [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.