

Article

RETRACTED: Next Point of Interest (POI) Recommendation System Driven by User Probabilistic Preferences and Temporal Regularities

Fengyu Liu ¹, Jinhe Chen ² , Jun Yu ³  and Rui Zhong ^{4,*} ¹ College of Information Technology, Shanghai Ocean University, Shanghai 201306, China; 2252119@st.shou.edu.cn² Tianyou College, East China Jiaotong University, Nanchang 330013, China; juanjuan@ecjtu.edu.cn³ Institute of Science and Technology, Niigata University, Niigata 950-2181, Japan; yujun@ie.niigata-u.ac.jp⁴ Information Initiative Center, Hokkaido University, Sapporo 060-0808, Japan

* Correspondence: zhongrui@iic.hokudai.ac.jp

Abstract: The Point of Interest (POI) recommendation system is a critical tool for enhancing user experience by analyzing historical behaviors, social network data, and real-time location information with the increasing demand for personalized and intelligent services. However, existing POI recommendation systems face three major challenges: (1) oversimplification of user preference modeling, limiting adaptability to dynamic user needs, (2) lack of explicit arrival time modeling, leading to reduced accuracy in time-sensitive scenarios, and (3) complexity in trajectory representation and spatiotemporal mining, posing difficulties in handling large-scale geographic data. This paper proposes NextMove, a novel POI recommendation model that integrates four key modules to address these issues. Specifically, the Probabilistic User Preference Generation Module first employs Latent Dirichlet Allocation (LDA) and a user preference network to model user personalized interests dynamically by capturing latent geographical topics. Secondly, the Self-Attention-based Arrival Time Prediction Module utilizes a Multi-Head Attention Mechanism to extract time-varying features, improving the precision of arrival time estimation. Thirdly, the Transformer-based Trajectory Representation Module encodes sequential dependencies in user behavior, effectively capturing contextual relationships and long-range dependencies for accurate future location forecasting. Finally, the Next Location Feature-Aggregation Module integrates the extracted representation features through an FC-based nonlinear fusion mechanism to generate the final POI recommendation. Extensive experiments conducted on real-world datasets demonstrate the superiority of the proposed NextMove over state-of-the-art methods. These results validate the effectiveness of NextMove in modeling dynamic user preferences, enhancing arrival time prediction, and improving POI recommendation accuracy.

Keywords: POI recommendations; Transformer; Self-Attention; generative probabilistic modeling; personalized user preferences; arrival time prediction

MSC: 68T35



Academic Editor: Cheng Tang

Received: 21 March 2025

Revised: 3 April 2025

Accepted: 8 April 2025

Published: 9 April 2025

Retracted: 29 December 2025

Citation: Liu, F.; Chen, J.; Yu, J.; Zhong, R. RETRACTED: Next Point of Interest (POI) Recommendation System Driven by User Probabilistic Preferences and Temporal Regularities. *Mathematics* **2025**, *13*, 1232. <https://doi.org/10.3390/math13081232>

Copyright: © 2025 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

As user expectations for personalized and intelligent services continue to rise, Point of Interest (POI) recommendation systems [1–4] have emerged to provide tailored recommendations by analyzing users' historical behaviors, social network information, and real-time

location data. These systems not only significantly enhance the overall user experience but also enable users to find and select interest points more efficiently and enjoyably, thereby increasing user satisfaction and loyalty to the platform. In addition, with the proliferation of mobile devices and advancements in location data-acquisition technologies [5], the importance of personalized recommendations [6–8] has become increasingly prominent, prompting ongoing research in this area.

Current POI recommendation methods can be categorized into traditional approaches [9–17], Attention-based methods [18–24], and Transformer-based methods [25–28]. Traditional approaches such as Collaborative Filtering [9–11], Matrix Factorization [12,13], K-Nearest Neighbors [14], Content-Based Recommendation and Latent Semantic models [16,17] have proven effective in certain contexts but fall short in capturing dynamic user preferences and complex relationships. Attention-based methods, including Attention Mechanism [18,19], Self-Attention [20,23], Temporal Attention Networks [21,22], and Graph Attention Networks (GAT) [24] for recommendation, have shown improvements in modeling user preferences but still lack explicit modeling of arrival time. Transformer-based methods [25–28] have made breakthroughs in trajectory representation and spatiotemporal mining, yet fail to comprehensively address the challenges posed by dynamic demands and temporal factors.

Despite existing methods demonstrating varying degrees of effectiveness, they face three major challenges: (a) The modeling of user preferences tends to overly simplify the distinction between long-term [29] and short-term [30] preferences, rendering it inflexible to rapidly changing user needs. (b) The lack of explicit modeling of arrival time results in subpar performance regarding timeliness and accuracy, particularly in scenarios requiring swift decision-making [31,32]. (c) The complexity of trajectory representation and spatiotemporal mining remains a pressing issue, especially in handling large-scale, high-dimensional geographic data. Existing studies [33–35] mainly focus on users' movement and behavior patterns, lacking fine-grained representation of semantic features of trajectories. Effectively extracting semantic and spatiotemporal features and integrating user behavior information continues to be a significant challenge.

To effectively address the aforementioned challenges, we propose a comprehensive model named NextMove, structured around four main modules. The Probabilistic User Preference Generation Module utilizes Latent Dirichlet Allocation (LDA) combined with an MLP-based user preference network to model users' geographical preferences, capturing latent topics from historical interaction data and refining these insights to adapt to shifts in user behavior and context. The Self-Attention-based Arrival Time Prediction Module leverages a Multi-Head Attention Mechanism to focus on temporal dynamic features critical for accurately predicting users' arrival times at different POIs, enhancing the accuracy of recommendations, particularly in time-sensitive scenarios. Meanwhile, the Transformer-based Trajectory Representation Module employs Transformer architectures to encode sequential dependencies in user behavior, analyzing the sequence of locations visited to capture contextual relationships and model long-range dependencies, which ultimately informs future location forecasts. Finally, the Next Location Feature-Aggregation Module integrates diverse feature representations generated by the previous modules using fully connected layers and softmax functions, synthesizing inputs into a consolidated output for next location prediction.

The main contributions are summarized as follows:

- The conventional simplifications [22,30,36] of user preference modeling into long-term and short-term categories fails to reflect the complexity and variability of user dynamic needs. To better capture these changes, we adopt the probabilistic generative approach of LDA, allowing the model to flexibly adapt to the evolution of user preferences.

This method considers not only the influence of historical behaviors but also incorporates an understanding of geographical context, rendering recommendations more personalized and aligned with users' current demands.

- The lack of explicit modeling of arrival time in traditional methods [9,16,17,30,35] often overlooks the impact of temporal factors on user decision-making, leading to insufficient accuracy in real-time recommendation scenarios. To address this challenge, we incorporate a Multi-Head Attention Mechanism in the Arrival Time Prediction Module, effectively capturing temporal dynamic features, thereby enhancing the model's predictive capability concerning arrival times. This not only improves the timeliness but also increases the accuracy of user decision-making at specific moments.
- We implemented the Transformer-based Trajectory Representation Module to encode sequential dependencies in user behavior. It analyzes the sequence of locations visited by users, enabling the model to capture contextual relationships and effectively represent long-range dependencies within the trajectory data. Extensive experimental results conducted on two datasets validate the effectiveness of our NextMove model.

The remainder of this paper is organized as follows. In Section 2, we conduct a comprehensive review of the related works regarding POI recommendation. Section 3 focuses on the preliminaries, where we introduce the essential concepts and basic theories. In Section 4, we present our proposed NextMove model in detail, which elaborates on the design principles and architecture of the model. Section 5 is dedicated to the experiments. Here, we describe the setup and experimental results, as well as analyze the performance of the NextMove model. In Section 6, we draw the conclusions of this paper.

2. Related Works

We mainly address the related work from the following three perspectives: traditional approaches [9–17,30,36–38], Attention-based methods [18–24,39], and Transformer-based models [25–28].

2.1. Traditional Approaches

Traditional methods primarily leverage established techniques such as Collaborative Filtering, Matrix Factorization, Markov models, and Recurrent Neural Networks (RNNs) to analyze user mobility patterns. Generally, active learning strategies in Collaborative Filtering-based recommendation systems [9,10] improve recommendation accuracy by selectively acquiring user ratings, with strategies classified according to personalization and hybridization. Matrix Factorization (MF) models [12,13] face some limitations, including assumptions of linear latent structures, reliance on explicit feedback, neglect of contextual information, static user preference modeling, and vulnerability to cold-start problems. These issues underscore the need for more advanced recommendation models capable of capturing complex relationships and adapting to dynamic user preferences. The n-Mobility Markov Chain [15] model effectively predicts an individual's next location based on historical mobility patterns, achieving high accuracy and demonstrating significant applications in geo-privacy and location-based services. Additionally, LSTPM [30] utilizes a context-aware nonlocal network alongside a geo-dilated LSTM to model both long-term user preferences and short-term interests. CSLSL [37] employs a multi-task learning framework that explicitly captures a causal structure representing the "time $\rightarrow$ activity $\rightarrow$ location" decision-making logic. Similarly, Flashback [38] builds upon an RNN framework to introduce a mechanism that revisits past hidden states with significant predictive relevance, optimizing the model's performance on sparse mobility traces. Moreover, the DeepMove [36] model integrates multi-modal embeddings with a Recurrent Neural Network to discern intricate mobility patterns while incorporating a historical Attention

Mechanism to utilize multi-level periodicities in trajectories. However, these traditional approaches often fail to explicitly model the temporal dependencies of arrival time, leading to suboptimal timeliness and accuracy, particularly in scenarios requiring real-time decision-making.

2.2. Attention-Based Methods

In contrast, Attention-based methods have emerged to better capture temporal dependencies and contextual information. For example, the STAN [23] model incorporates a bi-layer Attention Mechanism to understand spatiotemporal correlations in user trajectories while implementing a Personalized Item Frequency (PIF) to account for repetitive behaviors. CTRNext [39] integrates a trajectory semantic similarity module with multi-head Self-Attention to capture collaborative signals from similar users' check-ins, significantly enhancing the recommendation quality. Additionally, the ImNext [24] model addresses irregular user check-in intervals through multi-task learning and employs an Irregular Interval Attention Mechanism (i.e., IrrAttention) to forecast the next POI, including timing and distance intervals. Nevertheless, while Attention Mechanisms improve the extraction of temporal dependencies, they often lack an explicit representation of user arrival dynamics, limiting their ability to handle fine-grained timing predictions essential for real-time recommendations.

2.3. Transformer-Based Models

Transformer-based approaches represent a more recent advancement, showcasing their effectiveness in modeling complex user behavior. The Trans-Aux [26] model utilizes a transformer decoder-based architecture to predict subsequent location visits by integrating historical location sequences, temporal features, and travel modes. In a similar vein, GET-Next [27] develops a global trajectory flow map enhanced by transformer mechanisms to identify common movement patterns between POIs. Furthermore, the EEDN [28] model employs a hybrid hypergraph convolution encoder to optimize user–POI interactions while addressing challenges such as implicit feedback and cold-start scenarios, exhibiting the versatility of neural network architectures. Another approach, SNPM [31], employs a spatial dynamic network graph (SDNG) to capture contextual transitions in user trajectories, revealing latent relationships through RotatE to construct a POI similarity graph. Despite their strong representation power, Transformer-based models often emphasize global contextual relationships while overlooking the precise modeling of temporal constraints in user mobility, which can hinder their effectiveness in capturing real-time POI arrival patterns.

The evolution of POI recommendation systems has seen a significant shift from traditional methodologies grounded in Collaborative Filtering models to advanced neural architectures incorporating Attention Mechanisms and transformers. However, existing approaches still face challenges in explicitly modeling arrival time dependencies, limiting their ability to provide timely and accurate recommendations in real-world applications.

3. Preliminaries

In this section, we present a set of fundamental definitions that form the basis of this study that are crucial for understanding the subsequent sections.

Definition 1 (User Visit Record). *User visit records represent the spatiotemporal activity patterns of a user. Formally, the set of user visit records is defined as $R = \{(u, l, t) \mid u \in U, l \in L, t \in T\}$, where $U = \{u_1, u_2, \dots, u_{|U|}\}$ is the set of users, $L = \{l_1, l_2, \dots, l_{|L|}\}$ is the set of locations or Points of Interest (POIs), and $T = \{t_1, t_2, \dots, t_{|T|}\}$ represents the set of timestamps capturing the temporal dimensions of user interactions.*

Definition 2 (User–Location Frequency Matrix). Each user u_i can be associated with a frequency vector $F_i^u = [f_{i1}, f_{i2}, \dots, f_{i|L|}]$, where f_{ij} denotes the frequency of visits by u_i to location l_j . The user–location frequency matrix $M \in \mathbb{R}^{|U| \times |L|}$ summarizes user–location interactions across all users U and locations L .

Definition 3 (Topic Distribution). Each user u_i is represented by a topic distribution $\theta_i = [\theta_{i1}, \theta_{i2}, \dots, \theta_{iK}]$, where θ_{ik} denotes the probability of u_i engaging with topic k , satisfying $\sum_{k=1}^K \theta_{ik} = 1$. Similarly, each topic k is characterized by a location distribution $\phi_k = [\phi_{k1}, \phi_{k2}, \dots, \phi_{k|L|}]$, where ϕ_{kj} indicates the likelihood of location l_j being associated with topic k .

Definition 4 (Next POI Recommendation). Given a user $u \in U$ and his/her corresponding trajectory sequence $S_u = \{(l_1, t_1), (l_2, t_2), \dots, (l_n, t_n)\}$, the goal of POI recommendation is to predict a next activity location l_{n+1} from the candidate POIs that a particular user is likely to visit in the future.

4. Method

The overall framework of the proposed NextMove model is illustrated in Figure 1.

Our NextMove model consists of four modules: (a) Probabilistic User Preference Generation Module, which models geographical preferences using LDA and an MLP; (b) Self-Attention-based Arrival Time Prediction Module, which captures temporal dynamics with Multi-Head Attention; (c) Transformer-based Trajectory Representation Module, which encodes sequential dependencies via a Transformer; and (d) Next Location Feature Aggregation Module, which integrates these representations through a fully connected layer with softmax for next location prediction.

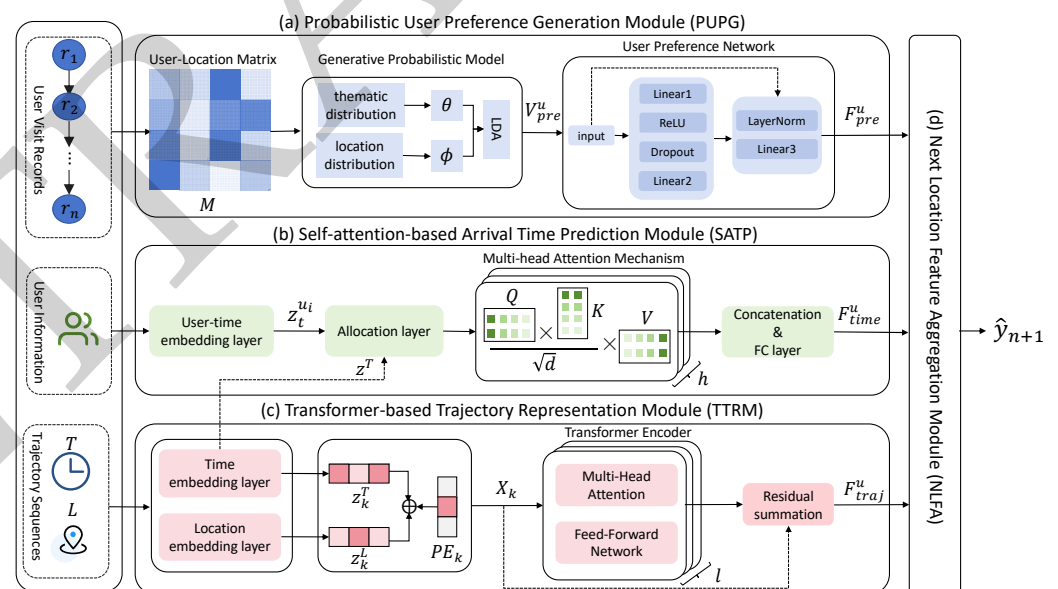


Figure 1. The overall framework of the proposed NextMove model for the next POI recommendation.

4.1. Probabilistic User Preference Generation Module

Our framework systematically encodes geographical preferences through users' historical activity sequences, combining empirical frequency patterns with probabilistic topic modeling [40,41] for explainable recommendations. The mathematical formalization pro-

ceeds through three coherent phases. The behavioral foundation builds from location visit frequencies, represented as:

$$F_i^u = [f_{i1}, f_{i2}, \dots, f_{i|L|}], \quad f_{ij} = \sum_{t \in T} \mathbb{I}((u_i, l_j, t) \in R) \quad (1)$$

where $\mathbb{I}(\cdot)$ indicates the occurrence of visits. Aggregating user vectors constructs the user–location frequency matrix:

$$M = \begin{bmatrix} f_{11} & f_{12} & \cdots & f_{1|L|} \\ f_{21} & f_{22} & \cdots & f_{2|L|} \\ \vdots & \vdots & \ddots & \vdots \\ f_{|U|1} & f_{|U|2} & \cdots & f_{|U||L|} \end{bmatrix} \in \mathbb{N}^{|U| \times |L|}, \quad (2)$$

where M_{ij} explicitly denotes the visit counts between user u_i and location l_j .

Probabilistic preference modeling employs Latent Dirichlet Allocation (LDA) [42,43] with a document–user analogy. In this context, users are treated as documents, and locations are treated as words within those documents. The LDA model generates N latent topics, where each topic represents a thematic pattern connecting activities and locations. For each user u_i , the topic distribution $\theta_i = [\theta_{i1}, \theta_{i2}, \dots, \theta_{iN}]$ (with $\sum_{n=1}^N \theta_{in} = 1$) characterizes the user's preferences across these N topics.

Each topic N is associated with a location distribution $\phi_N = [\phi_{N1}, \phi_{N2}, \dots, \phi_{N|L|}]$ (with $\sum_{j=1}^{|L|} \phi_{Nj} = 1$), where ϕ_{Nj} represents the probability of topic N being linked to location l_j . By integrating θ_i and ϕ_N , the model systematically captures users' activity patterns and spatial preferences, formally expressed as:

$$P(l_j|u_i) = \sum_{n=1}^N P(l_j|z_N)P(z_N|u_i) = \sum_{n=1}^N \phi_{Nj}\theta_{in}. \quad (3)$$

The LDA generative process can be expressed as:

$$\begin{cases} \theta_m \sim \text{Dir}(\alpha), & (\text{topic distribution}) \\ \phi_n \sim \text{Dir}(\beta), & (\text{location distribution}) \\ z_{m,n} \sim \text{Cat}(\theta_m), \\ l_{m,n} \sim \text{Cat}(\phi_{z_{m,n}}), \end{cases} \quad (4)$$

where θ_m represents the distribution of topics for user m , and ϕ_n represents the distribution of locations for topic N . Through this process, the user's activity preference feature vector v_{pre}^u is derived:

$$v_{pre}^u = \text{LDA}(M) \in \mathbb{R}^N, \quad (5)$$

where N denotes the number of predefined topics. This probabilistic framework uncovers interpretable themes in user activity patterns, enabling the characterization of each user's spatiotemporal behavior for recommendation tasks. Parameter estimation via collapsed Gibbs sampling follows this rule:

$$P(z_{m,n} = n | \mathbf{l}, \mathbf{z}_{-}) \propto \frac{(n_{m,n}^- + \alpha_n)}{\sum_{n'} (n_{m,n'}^- + \alpha_{n'})} \cdot \frac{(n_{n,l_{mn}}^- + \beta_{l_{mn}})}{\sum_p (n_{n,p}^- + \beta_p)}, \quad (6)$$

where $n_{m,n}^-$ counts the topic n assignments excluding the current observation. Neural refinement enhances probabilistic priors through a user preference network:

$$\mathbf{F}^u = \text{UserPreNet}(v_{pre}^u), \quad (7)$$

Specifically, $\text{UserPreNet}(\cdot)$ is implemented as follows:

$$\begin{cases} \mathbf{h}_1 = \text{Linear1}(v_{pre}^u), \\ \mathbf{h}_2 = \text{ReLU}(\mathbf{h}_1), \\ \mathbf{h}_3 = \text{Dropout}(\mathbf{h}_2), \\ \mathbf{h}_4 = \text{Linear2}(\mathbf{h}_3), \\ \tilde{\mathbf{h}} = \text{LayerNorm}(\mathbf{h}_4 + v_{pre}^u), \\ \mathbf{F}^u = \text{Linear3}(\tilde{\mathbf{h}}), \end{cases} \quad (8)$$

where the residual connection is applied, and the layer normalization is defined as:

$$\text{LayerNorm}(\mathbf{x}) = \gamma \odot \frac{\mathbf{x} - \mu}{\sigma} + \beta, \quad \mu = \frac{1}{d} \sum x_i, \quad \sigma = \sqrt{\frac{1}{d} \sum (x_i - \mu)^2}. \quad (9)$$

In the layer normalization formulation, $\mathbf{x} \in \mathbb{R}^d$ represents the input activation vector, where d denotes the feature dimension. The statistical parameters μ and σ compute the mean and standard deviation across the feature components of $\mathbf{x}$. The learnable parameters $\gamma \in \mathbb{R}^d$ (scaling vector) and $\beta \in \mathbb{R}^d$ (bias vector) enable affine transformation through element-wise multiplication ($\odot$) and addition, respectively, maintaining the network's expressive power while ensuring stable gradient propagation during training.

The final user representation synthesizes both global and local perspectives:

$$\mathbf{F}_{pre}^u = \text{Agg}(\mathbf{F}^u, e^u) \in \mathbb{R}^d, \quad (10)$$

where $\mathbf{F}^u \in \mathbb{R}^d$ corresponds to the global preference features refined through the MLP processing branch of the user preference network, $e^u \in \mathbb{R}^d$ represents trainable embeddings capturing localized user information, and $\text{Agg}(\cdot)$ denotes vector aggregation along the feature dimension. This composite encoding preserves macroscopic behavioral patterns through $\mathbf{F}^u$ while retaining microscopic interaction characteristics via e^u , enabling comprehensive user modeling across multiple granularities.

4.2. Self-Attention-Based Arrival Time Prediction Module

Many existing methods in time-series analysis and sequential prediction rely heavily on static transition models. These models typically use a predefined, fixed transition matrix $\mathbf{C}$, such that:

$$\mathbf{C} \in \mathbb{R}^{|z| \times |z|}, \quad (11)$$

where $|z|$ denotes the number of states in the transition model. While these methods perform adequately in scenarios with predictable patterns, they fall short in capturing the dynamic nature of real-world data. Static models are inherently unable to adapt to changes in user behavior over time, especially when the preferences of individuals shift due to external factors like seasonality, events, or personal habits. This rigidity leads to a suboptimal representation of temporal dynamics and limits their predictive power in applications such as personalized recommendation systems, mobility prediction, and dynamic scheduling.

The increasing complexity of user interactions and the need for fine-grained temporal understanding call for a more adaptive approach. To address these challenges, we propose a novel framework that integrates personalized user embeddings with hierarchical temporal representations. This framework leverages Multi-Head Attention Mechanisms to dynamically model interactions between temporal and contextual factors.

The structure of the proposed Self-Attention-based Arrival Time Prediction Module is displayed as Figure 2. To accurately model user-specific temporal preferences, we first introduce a user-time embedding $z_t^{u_i}$, which provides a personalized representation of individual user characteristics over time. This embedding captures temporal variability, allowing the model to adapt to shifts in user behavior effectively:

$$z_t^{u_i} = g_u^t(u_i, W_u^t), \quad (12)$$

where u_i represents the i -th user, W_u^t is a learnable parameter matrix, and $g_u^t(\cdot)$ is a non-linear transformation function. This embedding helps in modeling user u_i 's behavior at time t , enabling the model to differentiate between users with similar historical patterns but divergent future preferences.

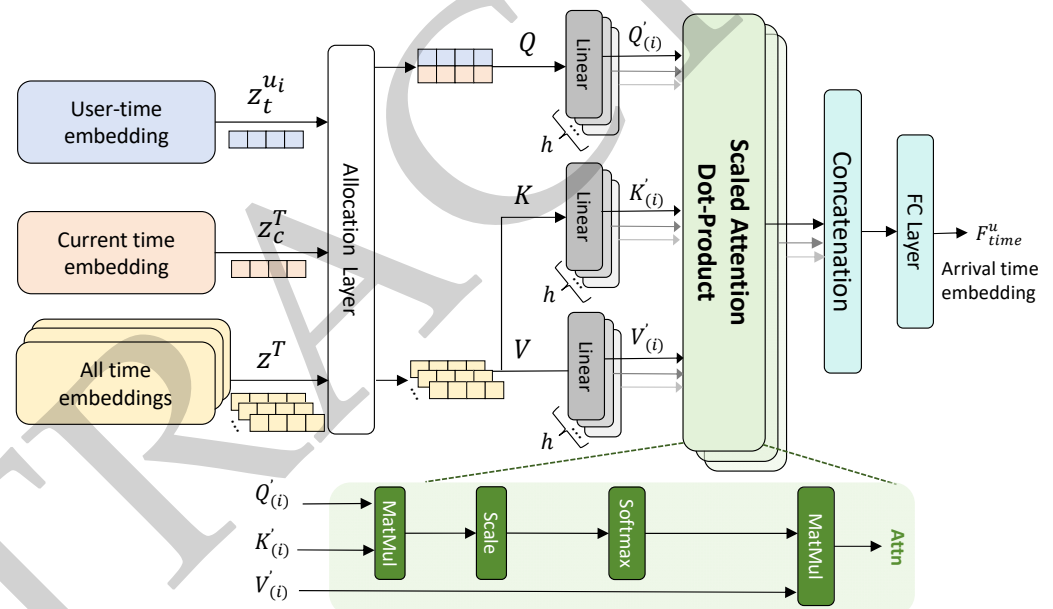


Figure 2. Structure of the proposed Self-Attention-based Arrival Time Prediction Module.

In addition to capturing user-specific behaviors, it is crucial to account for global temporal trends. For this purpose, we define current time embeddings z_c^T and all-time embeddings z^T :

$$z_c^T = f_t(t_c, W_c), \quad z^T = \{z_k^T\}_{k=1}^c, \quad (13)$$

where z_c^T represents the embedding of the current time t_c , and z^T is a collection of embeddings for all possible timestamps. These embeddings allow the model to encode periodic patterns, such as daily or weekly cycles, and adapt to irregular temporal distributions.

Attention Mechanisms have emerged as a powerful tool for modeling dependencies in sequential data. In our framework, we employ a Multi-Head Attention Mechanism to dynamically compute the relationships between personalized user preferences and global temporal contexts. The query (Q), key (K), and value (V) matrices are constructed as:

$$Q = z_t^{u_i} \oplus z_c^T, \quad K = V = z^T, \quad (14)$$

where $\oplus$ indicates the concatenation of user-time embeddings with global temporal embeddings, creating a comprehensive input representation for the Attention Mechanism. Each attention head computes scaled dot-product attention as:

$$\text{Attn}_i = \text{softmax}\left(\frac{Q'_i(K'_i)^\top}{\sqrt{d_h}}\right)V'_i, \quad (15)$$

where d_h is the dimensionality of the key vectors. The final multi-head output F_{time}^u aggregates attention results across all heads:

$$F_{time}^u = \text{FC}(\text{Attn}_1 \oplus \text{Attn}_2 \oplus \dots \oplus \text{Attn}_h), \quad (16)$$

where $\text{FC}(\cdot)$ is a fully connected layer and h is the number of attention heads. This mechanism enables the model to attend to different temporal patterns simultaneously, capturing both short-term fluctuations and long-term trends.

The final output embeddings for arrival time prediction, denoted as F_{time}^u , are generated from the aggregated attention results, allowing for effective modeling of arrival time based on user behavior and temporal context.

4.3. Transformer-Based Trajectory-Representation Module

To effectively model the sequential patterns within users' activity location sequences, we leverage the Transformer architecture, which has demonstrated exceptional capabilities in encoding temporal dependencies in sequential data [25,26]. By applying the Transformer framework, we aim to capture complex spatiotemporal relationships across sequences of location–time pairs $(l_{n-m+1}, t_{n-m+1}), \dots, (l_n, t_n)$, ultimately generating robust sequential embeddings. Each location l_k and timestamp t_k in the sequence is encoded as a vector in a shared d -dimensional latent space. The embeddings are denoted as $z_k^L \in \mathbb{R}^d$ for locations and $z_k^T \in \mathbb{R}^d$ for timestamps. Specifically, the location embedding is computed as:

$$z_k^L = h^L(l_k, W^L), \quad (17)$$

where $h^L(\cdot, \cdot)$ represents an embedding function parameterized by the learnable matrix W^L . The computation of the timestamp embedding z_k^T follows a similar mechanism. To prepare the input for the Transformer, we combine the location and time embeddings with positional encoding. The input representation for each sequence element is formulated as:

$$X_k = z_k^T + z_k^L + \text{PE}_k, \quad k \in \{n-m+1, \dots, n\}, \quad (18)$$

where PE_k denotes the positional encoding that allows the Transformer to incorporate order information. The positional encoding matrix $\mathbf{P} \in \mathbb{R}^{n \times d}$ is defined as follows:

$$\mathbf{P}_{pos,2i} = \sin\left(\frac{pos}{10000^{2i/d}}\right), \quad (19)$$

$$\mathbf{P}_{pos,2i+1} = \cos\left(\frac{pos}{10000^{2i/d}}\right), \quad (20)$$

where pos is the position index, $i \in [0, d/2)$ is the dimension index, and d denotes the embedding dimension. The Transformer model processes the input sequence through stacked layers, each comprising a multi-head Self-Attention (MHSA) mechanism and a feed-forward network (FFN). Residual connections and layer normalization are integrated into both components to improve learning stability and efficiency. The MHSA mechanism

is designed to capture dependencies across all elements of the sequence. For each attention head, the weighted interactions between inputs are computed as:

$$\text{head}_i = \text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d}}\right)V, \quad (21)$$

where Q, K, V are the query, key, and value matrices derived from the input sequence X_k . The outputs of all attention heads are concatenated and transformed:

$$\text{MHSA}(X) = \text{Concat}(\text{head}_1, \dots, \text{head}_h)W_o, \quad (22)$$

where W_o is the output projection matrix. The FFN applies element-wise transformations to enhance non-linear representational capacity:

$$\text{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2, \quad (23)$$

where W_1, W_2 and b_1, b_2 are trainable parameters of the feed-forward layers. Each layer updates its intermediate output using residual connections:

$$h_k = \text{LayerNorm}(X_k + \text{MHSA}(X_k)), \quad (24)$$

$$h_k^{\text{out}} = \text{LayerNorm}(h_k + \text{FFN}(h_k)). \quad (25)$$

To ensure that the model retains rich spatiotemporal information, the final sequential embedding is constructed as the sum of the sequential context and the embeddings of the current location and timestamp:

$$F_{traj}^u = h_n^{\text{out}} + z_n^L + z_n^T. \quad (26)$$

4.4. Next Location Feature-Aggregation Module

After generating three distinct contextual embeddings, the next step is their concatenation, which is then input into a fully connected layer. This layer is essential for computing the probabilities that represent the likelihood of a user transitioning to each potential activity location. This process is mathematically expressed using the softmax function as follows:

$$P(\hat{y}_{n+1}) = \text{softmax}\left(\text{FC}\left(F_{pre}^u \oplus F_{time}^u \oplus F_{traj}^u\right)\right) \quad (27)$$

In this equation, $\text{FC}(\cdot)$ indicates the fully connected layer that integrates residual connections. The architecture is typically composed of two sequential linear transformations, complemented by a ReLU (Rectified Linear Unit) activation function. This design choice enhances the model's ability to process and transform the concatenated embeddings into a form that can accurately predict the transition probabilities for the user.

LDA effectively captures long-term, topic-level user preferences by modeling POIs as distributions over latent topics. This allows the model to extract meaningful semantic structures that are often ignored in traditional Transformer-based approaches. Transformer excels at learning short-term sequential dependencies in user visit behaviors, leveraging Self-Attention to capture temporal dynamics effectively. By combining these two, we enable global preference-aware sequence modeling, where the topic distributions from LDA guide the Self-Attention Mechanism in Transformer. This enhances interpretability and improves the personalization of recommendations. Instead of treating LDA and Transformer outputs as separate features, we introduce a latent space alignment technique, ensuring consistency between topic-based representations and Self-Attention embeddings.

During the training process, the model is framed as a classification task aimed at predicting the next location from a comprehensive set of potential locations, denoted as $\mathcal{P}$. The loss function employed for this prediction task is a multi-class cross-entropy loss, expressed mathematically as:

$$\mathcal{L} = - \sum_{j=1}^{|\mathcal{P}|} P(y_{n+1})_j \log P(\hat{y}_{n+1})_j \quad (28)$$

Here, $P(y_{n+1})_j$ corresponds to the ground truth for the next location, represented in a one-hot encoded format. Specifically, $P(y_{n+1})_j = 1$ if the next activity location aligns with the j -th position in the location set. Conversely, $P(\hat{y}_{n+1})_j$ reflects the model's predicted probability of user activity occurring at location j . By minimizing the cross-entropy loss, it ensures that the NextMove model learns the associations between user behavior and contextual factors, thereby enhancing its predictive accuracy for determining the user's subsequent activity location.

5. Experiments

5.1. Dataset

The proposed framework was evaluated using two real-world datasets, Traffic Camera data (TC) [44] and Mobile Phone data (MP) [45], obtained from a popular location-based social network. Both datasets contain essential check-in records, including user IDs, POI IDs, timestamps, and geographic coordinates. To improve data quality, preprocessing steps were applied to remove users with insufficient interactions and POIs with low visit frequencies. The check-in records were then segmented into trajectory sequences to facilitate the next POI recommendation task. The datasets were partitioned chronologically, with 70% allocated for training, 20% for testing, and 10% for validation to fine-tune model hyperparameters. Figure 3 shows the sequence length distributions of the two datasets. The TC dataset, in Figure 3a, has a long-tailed distribution, with an average sequence length of 110 and most lengths between 80 and 140. The MP dataset, in Figure 3b, has a near-normal distribution, with an average sequence length of 126 and most lengths between 100 and 150. These differences in sequence length distributions highlight distinct behavioral patterns among users in the two datasets. Table 1 summarizes overall dataset statistics, including the number of users, POIs, and total trajectory records.

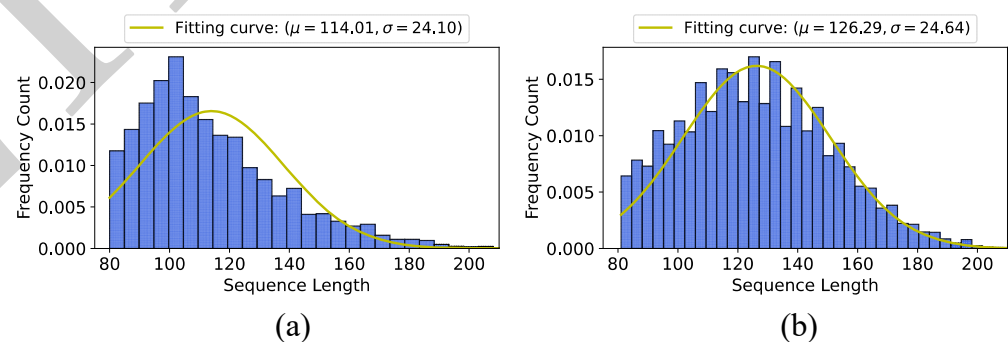


Figure 3. The length distribution of trajectory sequences and fitting curves in two datasets. (a) TC dataset. (b) MP dataset.

Table 1. Statistics of the experimental datasets.

Dataset	# Users	# POIs	# Records	# Days	Min Length	Max Length
TC	7800	2418	1,115,619	61	80	277
MP	10,000	20,607	1,594,551	75	81	223

5.2. Evaluation Metrics

We employ the following metrics to assess the performance of various methods.

1. **Accuracy (Acc):** This metric measures the proportion of correct predictions among the top- K predicted results. We adopt $Acc@K$, where $K \in \{1, 3, 5, 10\}$, and assign a value of 1 if the ground truth appears in the top- K predictions, otherwise 0. This reflects the model's ability to include the correct POI within its K -best recommendations.
2. **Mean Reciprocal Rank (MRR):** MRR evaluates the average reciprocal rank of the ground truth in the prediction list. The calculation is as follows: $MRR = \frac{1}{N} \sum_{i=1}^N \frac{1}{rank_i}$, where N is the number of queries, and $rank_i$ represents the rank position of the ground truth for the i -th query.

5.3. Parameter Settings

We utilize the Adam optimizer to carry out the model optimization process. The optimizer is initialized with a learning rate set to 0.01 and an L2 regularization penalty assigned as 1×10^{-6} . The embedding size $d = (16, 32)$ for (TC, MP) datasets. For the Probabilistic User Preference Generation Module, we set the number of LDA topics N equal to 400 for two datasets. We employ four attention heads $h = 4$ in the Self-Attention-based Arrival Time Prediction Module. We construct the Transformer-based Trajectory Representation Module by stacking two encoder layers for TC and three layers for MP, and each layer is configured with a dropout rate of 0.1 and an activation function of gelu. The NextMove model is then trained for 50 epochs, with a batch size of 256 being used during each training iteration. Our experiments were conducted on a machine with Tesla K80 with 12 GB memory, using PyTorch 2.1 for implementation. To determine the most optimal model parameters, we conduct a grid search. Specifically, we searched over the following ranges: (1) learning rate: $\{0.001, 0.005, 0.01, 0.05\}$; (2) embedding size: $\{8, 16, 32, 64\}$; (3) number of LDA topics: $\{200, 300, 400, 500, 600\}$; (4) number of Transformer layers: $\{1, 2, 3, 4, 5\}$; (5) number of attention heads: $\{1, 2, 4, 8\}$. The selection criteria for optimal hyperparameters were based on the validation performance, using accuracy as the evaluation metric. Section 5.6 provides a detailed demonstration of the impact of different hyperparameter combinations on model performance.

5.4. Baseline Model

We compare the performance of our NextMove model with nine state-of-the-art baselines, which are popular for POI recommendation and location prediction.

- **ARNN [22]** combines sequential regularities with neighboring location transitions to provide personalized next location recommendations, using a knowledge graph to identify similar locations and an Attention Mechanism to integrate these transitions with RNN-processed sequential data.
- **LSTPM [30]** captures users' long-term preferences through a context-aware nonlocal network and a geo-nonlocal structure. Additionally, it employs a geo-dilated LSTM to model users' short-term interests.
- **STAN [23]** incorporates a bi-layer Attention Mechanism to capture spatiotemporal correlations in user trajectories and a Personalized Item Frequency (PIF) to account for repetitive behaviors.
- **GETNext [27]** develops a global trajectory flow map to identify common patterns in user movement between Points of Interest (POIs). This graph-based approach integrates the spatial, categorical, and temporal characteristics of POIs, converting them into latent embeddings using a Graph Convolutional Network (GCN).

- **EEDN** [28] employs a hybrid hypergraph convolution encoder for user–POI interactions and a Matrix Factorization decoder for feature alignment, addressing implicit feedback and cold-start challenges while enhancing recommendation quality.
- **PG2Net** [46] integrates personalized and group preferences using a Bi-LSTM and attention module for individual mobility, and a group preference module for spatiotemporal dynamics. It also uses graph embedding for sequential relationships and an auxiliary loss function to improve accuracy.
- **CTRNext** [39] combines a trajectory semantic similarity module with multihead Self-Attention to capture collaborative signals from similar users' check-ins.
- **STHGCN** [47] uses hypergraphs to capture complex relationships between user check-ins and trajectories. It incorporates spatiotemporal data and aggregates multi-hop trajectory information, enhancing accuracy in sparse or cold-start scenarios.
- **ImNext** [24] tackles irregular user check-in intervals using multi-task learning. It utilizes an Irregular Interval Attention (i.e., IrrAttention) module for preferences, a novel edge-enhanced Graph Attention Network (i.e., EA-GAT) for spatiotemporal influences, and a multi-task framework to predict the next POI, timing, distance intervals, and prior visits.

5.5. Performance Comparasion

Tables 2 and 3 present a comprehensive comparison of the recommendation performance of our NextMove model against various baseline models on the TC and MP datasets. The results lead us to the following conclusions:

- NextMove consistently outperformed baseline models across all evaluation metrics in both datasets. Specifically, in the TC dataset, NextMove achieved an Acc@5 of 65.26, surpassing GETNext by 18.55%. Similarly, in the MP dataset, it reached an Acc@5 of 66.88, outperforming GETNext by 9.29%. This consistent superiority not only highlights NextMove's effectiveness in accurately predicting user preferences but also underscores its role as a leading solution for Point of Interest (POI) recommendations in diverse contexts. The experimental results substantiate the capability of the proposed NextMove model to address users' personalized needs while adeptly modeling dynamic preferences.
- NextMove also exhibited remarkable accuracy across various metrics, achieving a Mean Reciprocal Rank (MRR) of 51.39 on the TC dataset and 51.66 on the MP dataset. These scores signify significant enhancements over competing models, demonstrating NextMove's ability to rank relevant POIs effectively. Additionally, the model's Acc@1 and Acc@3 scores reflect its proficiency in delivering timely and pertinent recommendations, adeptly catering to both immediate and evolving user needs. This capability is particularly crucial in environments where user preferences can change rapidly, necessitating a recommendation system that can adapt in real time.
- The comparative analysis of baseline models reveals distinct strengths and weaknesses that contribute to their overall performance. For instance, ARNN and LSTPM, both relying heavily on RNN architectures, exhibit limitations in their capacity to capture long-term user preferences and temporal dynamics effectively. ARNN achieved an Acc@1 of only 23.60%, while LSTPM recorded a marginally better performance at 25.33%. These models primarily focus on sequential data processing, failing to fully account for the complexities of user arrival dynamics and external contextual factors that are critical for real-time applications. GETNext employs a graph-based approach, achieving an Acc@5 of 55.05%. However, its reliance on static graph structures limits its adaptability to the dynamic nature of user preferences, which can fluctuate based on various situational contexts. Similarly, ImNext enhances performance by incor-

porating multi-modal data, reaching an Acc@5 of 61.33%. Yet, it still lacks explicit temporal modeling, which hampers its ability to provide timely and contextually relevant recommendations.

- While ARNN, LSTPM, GETNext, and ImNext each present valuable approaches to recommendation systems, they collectively fall short in capturing the dynamic and temporal aspects of user behavior effectively. A notable strength of NextMove is its innovative incorporation of temporal dynamics through the Self-Attention-based Arrival Time Prediction Module. NextMove’s architecture directly addresses these limitations by ensuring precise timing predictions that are essential for generating real-time recommendations. This capability not only enhances the relevance of the recommendations but also improves user satisfaction by aligning suggestions with users’ immediate needs and contexts.
- Although some POI recommendation models incorporate distance as a factor, they lack an explicit arrival time modeling mechanism. This omission can lead to suboptimal recommendations, as users may not choose locations that require excessive travel time, even if those locations align with their general preferences. In our model, we address this gap by integrating arrival time prediction directly into the recommendation process. By explicitly modeling temporal constraints, our approach enhances recommendation relevance by ensuring that suggested POIs are not only preferred but also realistically reachable within the user’s schedule. Our experimental results further demonstrate that incorporating arrival time prediction improves the overall recommendation quality, as it better aligns with real-world user behavior.

Table 4 reports the time complexity analysis and memory usage. We can observe that the training time per epoch is 52.64 s on the TC dataset and 105.80 s on the MP dataset, while the inference time per query is 8.34 s and 11.25 s, respectively. The memory usage remains within a reasonable range, with the largest model requiring 2470 MiB. These results suggest that while the model is complex, its computational demands remain manageable. Regarding the Transformer module, its theoretical time complexity is $\mathcal{O}(n^2d)$ for Self-Attention, where n is the sequence length and d is the hidden dimension. However, in our implementation, we mitigate this computational overhead using efficient Attention Mechanisms and optimized matrix operations, making it a trade-off between accuracy and efficiency.

Table 2. Recommendation performance comparison with baselines on the Traffic Camera data (TC).

Models	Acc@1	Acc@3	Acc@5	Acc@10	MRR
ARNN	23.60	36.02	39.98	47.44	31.79
LSTPM	25.33	39.85	42.36	50.72	33.50
STAN	29.77	44.05	51.19	58.55	40.01
GETNext	31.63	46.88	55.05	61.57	42.68
EEDN	32.71	48.97	57.15	63.22	43.87
PG2Net	34.89	51.59	59.02	65.44	45.33
CTRNext	35.69	53.64	60.73	66.05	46.18
STHGCN	36.06	54.52	61.01	68.52	48.11
ImNext	36.28	54.97	61.33	69.71	48.50
NextMove	40.50	58.45	65.26	72.53	51.39

Table 3. Recommendation performance comparison with baselines on the Mobile Phone data (MP).

Models	Acc@1	Acc@3	Acc@5	Acc@10	MRR
ARNN	28.05	44.33	50.05	52.99	36.62
LSTPM	31.57	47.66	55.69	58.06	39.68
STAN	35.00	52.86	59.38	62.91	43.57
GETNNext	36.75	54.00	61.19	65.28	46.99
EEDN	36.98	54.23	61.80	65.93	47.75
PG2Net	37.01	55.80	62.33	67.23	49.05
CTRNext	37.35	55.99	62.85	67.76	49.60
STHGCN	38.00	57.29	64.77	69.18	50.11
ImNext	38.52	58.45	65.01	71.85	50.23
NextMove	40.00	60.46	66.88	73.24	51.66

Table 4. Time complexity analysis and memory usage.

Dataset	Training (s/epoch)	Inference Time (s)	Memory Usage (MiB)	Number of Parameters
TC	52.64	8.34	1844	1,115,730
MP	105.80	11.25	2470	2,659,439

5.6. Hyperparameter Analysis

Figure 4 depicts the impact of LDA topics (N) and embedding size (d) on the next POI recommendation task for the TC and MP datasets. As N increases, the evaluation metric ACC@1 exhibits a substantial upward trend when $N \leq 400$. However, when $N > 400$, the performance gain becomes marginal, suggesting that an excessively large N does not significantly enhance the model's performance. The results reveal that larger embedding sizes ($d = 32$ and $d = 64$) generally outperform smaller ones ($d = 8$ and $d = 16$). Optimal performance is achieved with $d = 16$ and $N = 400$ for the TC dataset, achieving an ACC@1 of 40.50%, and $d = 32$ and $N = 400$ for the MP dataset, achieving an ACC@1 of 40.00%. The superior performance at optimal d values suggests that smaller embeddings may not effectively capture the latent features of the data, leading to information loss, while excessively large embeddings can introduce noise and redundant information, causing overfitting. Based on these observations, setting $N = 400$ and tuning d to align with dataset characteristics is critical for achieving optimal recommendation performance.

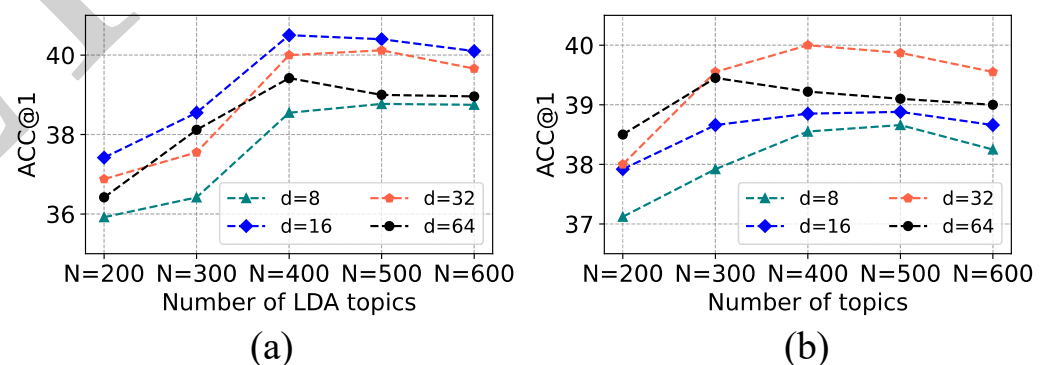
**Figure 4.** The impact of different numbers of LDA topics N and embedding size d in two datasets. (a) TC dataset. (b) MP dataset.

Figure 5 examines the impact of Transformer layer number (l) and attention head number (h) on the next POI recommendation performance for the TC and MP datasets. Consistently, ACC@1 improves as l increases, but the benefit lessens for $l > 4$, suggesting deeper architectures may overfit or fail to generalize with limited dataset sizes. As (h)

increases, performance improves, with $h = 4$ achieving the best results on both datasets. Specifically, for the TC dataset, the optimal configuration is $h = 4$ and $l = 2$, achieving an ACC@1 of 40.50%, while for the MP dataset, the best performance is observed with $h = 4$ and $l = 3$, where ACC@1 reaches 40.00%. The results indicate that inadequate hidden layers (h) restrict the model's ability to model spatiotemporal dependencies, while excessive hidden layers (h) may increase model complexity without significant performance gains. Therefore, balancing the number of hidden layers (h) and layers (l) based on dataset characteristics is critical for optimizing model performance.

The MP dataset has a larger number of unique POIs and more complex temporal dynamics compared to the TC dataset. As a result, the model requires a higher capacity (e.g., more Transformer layers and a larger embedding size) to effectively capture sequential dependencies and probabilistic transitions. The TC dataset, being relatively simple, benefits from a more compact model, reducing the risk of overfitting. The MP dataset exhibits longer-range dependencies and more irregular visit patterns, which require deeper representations to model. Hence, we use a deeper Transformer architecture with more parameters to enhance expressiveness. In contrast, the TC dataset has more regular, short-term dependencies, making a shallower model sufficient for capturing meaningful patterns.

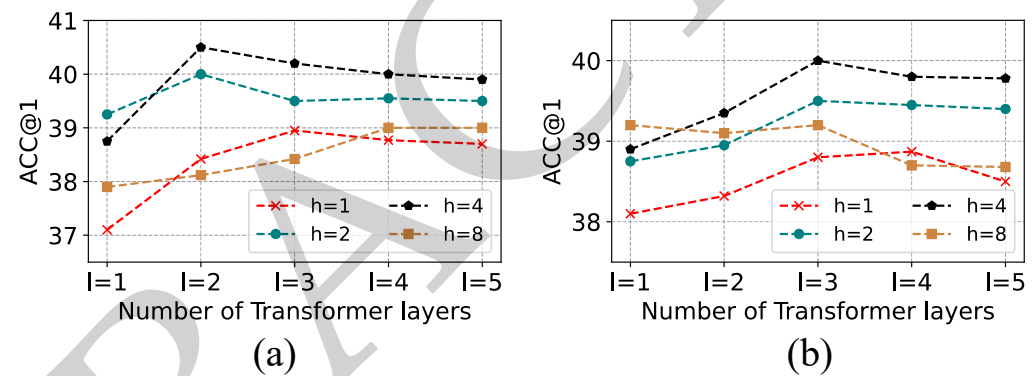


Figure 5. The impact of different numbers of Transformer layers l and attention heads h in two datasets. (a) TC dataset. (b) MP dataset.

5.7. Ablation Study

In order to assess the significance of each core module, we undertook a set of ablation experiments. The proposed NextMove served as the fundamental model, and we generated three distinct variants by eliminating various components within it. The results of these ablation studies are illustrated in Figure 6.

- **w/o PUPG** represents removing the Probabilistic User Preference Generation Module (PUPG) from the basic full NextMove model.
- **w/o SATP** means removing the Self-Attention-based Arrival Time Prediction Module (SATP) from the basic full NextMove model.
- **w/o TTRM** indicates eliminating the Transformer-based Trajectory Representation Module (TTRM) from the proposed full NextMove model.

The ablation study confirms that TTRM is the most critical module, followed by PUPG, while SATP exhibits a more context-dependent effect. The key findings are summarized as follows:

- (1) **The removal of TTRM results in the most significant performance drop**, demonstrating its essential role in the model. TTRM effectively captures sequential dependencies in user trajectories through a Transformer-based architecture, making it indispensable for accurate next location recommendations.

- (2) **Excluding PUPG leads to a considerable decline in accuracy**, highlighting the significance of personalized user preference modeling. The results indicate that incorporating individual behavior patterns enhances the model's ability to capture user interests and improves recommendation performance.
- (3) **The impact of SATP varies across datasets**. In the TC dataset, removing SATP causes a noticeable performance drop when $k = 5$, suggesting that the model heavily relies on precise arrival time predictions to enhance accuracy. In contrast, in the MP dataset, which is based on mobile signal data, the removal of SATP has a relatively small impact, implying that trajectory data plays a more dominant role in this context.
- (4) **PUPG and SATP exhibit varying importance across datasets**. In the TC dataset, PUPG has a more substantial impact when $k = 5$, underscoring its importance in capturing user preferences. Additionally, the varying impact of SATP across datasets suggests that the significance of temporal modeling depends on the characteristics of the data, further emphasizing the complexity of user mobility behavior.

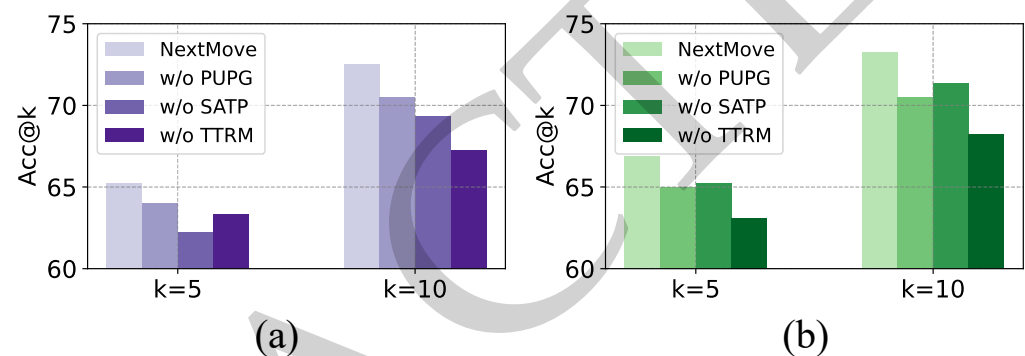


Figure 6. Result of ablation study. (a) TC dataset. (b) MP dataset.

A comparison of different arrival time estimators on two datasets is presented in Tables 5 and 6. The Self-Attention-based Arrival Time Prediction Module (SATP) consistently outperforms both TCN and GRU across all accuracy metrics on both datasets. Arrival time prediction in POI recommendation requires capturing both short-term dependencies (e.g., immediate transitions) and long-term periodic trends (e.g., daily commuting patterns). SATP offers superior modeling fidelity due to the following advantages:

- (1) **Global Context Awareness:** Unlike GRU (which relies on sequential recurrence) and TCN (which has a fixed receptive field), Self-Attention computes dependencies between all timestamps in parallel, effectively modeling both short-term and long-term influences.
- (2) **Handling of Irregular Time Intervals:** POI visits occur at irregular time gaps, making it difficult for RNNs (which assume regular step intervals) and TCNs (which rely on fixed kernel sizes) to generalize well. SATP incorporates continuous-time positional encoding, allowing it to dynamically weigh past events based on actual elapsed time.
- (3) **Efficient Information Propagation:** Unlike GRU, which suffers from vanishing gradients over long sequences, and TCN, which requires deep networks to expand its receptive field, SATP can directly attend to relevant timestamps, leading to more effective long-range dependency modeling.

To rigorously assess whether the observed improvements are statistically significant, we applied a statistical test, i.e., the paired t-test. We tested the average accuracy differences between the proposed NextMove and the three variants (PUPG, SATP, and TTRM) across multiple trials. Each trial corresponds to an independent run with different random seeds, ensuring robustness. Each trial was conducted 10 times on multiple independent test sets, and the average accuracy and MRR of each experiment was recorded to form paired data

(full model vs ablation model). We present the p -values for the comparisons in Table 7. A p -value < 0.05 indicates statistical significance at the 95% confidence level. We can find that all p -values are below 0.05, confirming that the improvements of NextMove over the three variants are statistically significant across all metrics.

Table 5. Comparison of different arrival time estimators on the Traffic Camera data (TC).

Models	Acc@1	Acc@3	Acc@5	Acc@10
SATP	40.50	58.45	65.26	72.53
TCN	38.44	56.08	63.25	70.04
GRU	35.12	53.49	59.78	67.71

Table 6. Comparison of different arrival time estimators on the Mobile Phone data (MP).

Models	Acc@1	Acc@3	Acc@5	Acc@10
SATP	40.00	60.46	66.88	73.24
TCN	37.20	56.15	62.73	68.05
GRU	36.77	54.60	61.85	66.38

To evaluate the contributions of different modules within our NextMove framework, we conducted a multi-module ablation study and provided a visualization to highlight their effects on Mean Reciprocal Rank (MRR), as in Figure 7. The key observations are firstly removing both TTRM and PUPG (w/o TTRM + PUPG) leads to the most significant performance drop on the MP dataset, indicating their crucial role in capturing both trajectory representation and user preference dynamics. Secondly, w/o SATP + PUPG also results in a noticeable decline, emphasizing that modeling timestamp dependencies (SATP) is essential for accurate arrival time prediction and POI recommendation. PUPG plays a significant role in improving recommendation accuracy by dynamically adjusting user preferences. SATP enhances both arrival time prediction and ranking performance, making it a key contributor to NextMove's success.

Table 7. Results of significance tests (p -values) on two datasets.

Dataset	Models	Average Acc	MRR
TC	NextMove vs. PUPG	0.041	0.035
TC	NextMove vs. SATP	0.034	0.012
TC	NextMove vs. TTRM	0.049	0.033
MP	NextMove vs. PUPG	0.025	0.015
MP	NextMove vs. SATP	0.017	0.026
MP	NextMove vs. TTRM	0.044	0.028

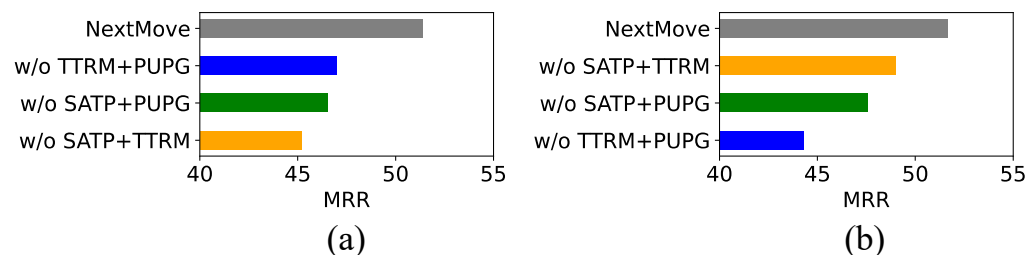


Figure 7. Result of multi-module ablation study. (a) TC dataset. (b) MP dataset.

5.8. Study of Latent Topics in LDA

To further illustrate the stability of latent topics over time, we conducted an empirical analysis using the TC dataset. We selected 10 representative users and visualized their

topic distributions over two different 30-day periods (i.e., the first 30 days vs the last 30 days). Each topic represents a semantic category of POIs (e.g., restaurants, stations, tourist attractions), and each user's topic preference is captured in a 10×400 matrix, where 400 represents the number of topics. Each matrix entry (value in $[0,1]$) denotes the user's preference intensity for a given topic. As shown in Figure 8, the topic distributions remain highly consistent over time, demonstrating that while individual POI preferences shift over time, the underlying latent topics remain stable, capturing consistent behavioral patterns. The low divergence between these distributions confirms that LDA captures robust and transferable behavioral patterns.

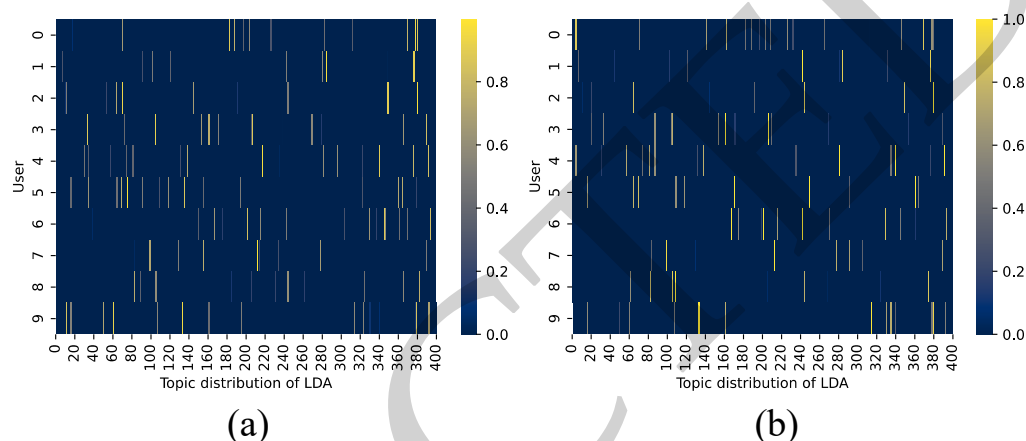


Figure 8. Visualization of topic distributions over time. (a) The first 30 days. (b) The last 30 days.

We conducted three variant experiments, NextMove (LDA), NextMove (DeepWalk), and NextMove (Word2Vec), to validate the effectiveness of LDA for location prediction, as shown in Figure 9. Unlike standard embedding-based approaches (e.g., Word2Vec, graph embeddings-DeepWalk), which rely purely on proximity in latent space, LDA explicitly models distributions over topics. LDA captures high-level behavioral patterns, leading to more transferable representations across users. Our NextMove (LDA) model outperforms purely embedding-based methods in the MRR evaluation metric. Topic-based representations provide a human-interpretable way to understand user preferences. The semantic structure of user preferences remains stable, avoiding the fluctuation issues observed in traditional embeddings. In addition, the topic distributions serve as prior knowledge, guiding the Transformer model in modeling user mobility more effectively.

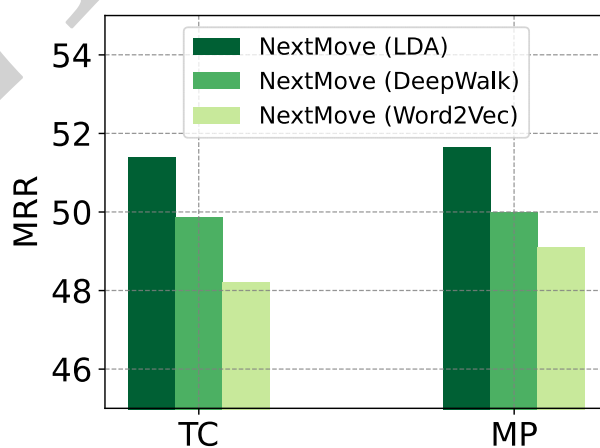


Figure 9. Performance comparison with Word2Vec and DeepWalk.

6. Conclusions

In this paper, we propose NextMove, a novel Point of Interest (POI) recommendation model designed to enhance recommendation accuracy by addressing key challenges in user preference modeling, arrival time prediction, and trajectory representation. NextMove integrates four core modules, utilizing Latent Dirichlet Allocation (LDA) to model dynamic user preferences, leveraging Multi-Head Attention to extract temporal dynamics and improve arrival time accuracy, and using a Transformer architecture to encode sequential dependencies in user behavior for precise location forecasting. Last, the Next Location Feature-Aggregation Module synthesizes extracted features through fully connected layers and softmax functions to generate the final POI recommendation. Extensive experiments on two real-world datasets (TC and MP) demonstrate that NextMove outperforms state-of-the-art models in recommendation accuracy. These findings validate the effectiveness of NextMove in modeling dynamic user preferences, improving arrival time predictions, and enhancing overall POI recommendation accuracy. While NextMove has demonstrated excellent recommendation performance, we acknowledge the importance of considering comprehensively multi-source data. In future work, we will enhance our model by integrating user contextual information, such as weather and social activities, to better capture user preferences and movement patterns.

Author Contributions: F.L.: Conceptualization, Writing—original draft, Writing—review & editing, and Funding acquisition. J.C.: Investigation, Methodology, Supervision, and Writing—review & editing. J.Y.: Investigation, Data curation, and Writing—review & editing. R.Z.: Writing—original draft, Writing—review & editing, Supervision, and Project administration. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The dataset can be downloaded from the link <https://zenodo.org/records/10836269> (accessed on 18 March 2024), and the source code of this research can be downloaded from <https://github.com/Mortal-fy/NextMove> (accessed on 21 March 2025).

Conflicts of Interest: The authors of this publication declare there are no conflicts of interest.

References

1. 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. [\[CrossRef\]](#)
2. Rahmani, H.A.; Deldjoo, Y.; Di Noia, T. The role of context fusion on accuracy, beyond-accuracy, and fairness of point-of-interest recommendation systems. *Expert Syst. Appl.* **2022**, *205*, 117700. [\[CrossRef\]](#)
3. Lim, N.; Hooi, B.; Ng, S.K.; Goh, Y.L.; Weng, R.; Tan, R. Hierarchical multi-task graph recurrent network for next poi recommendation. In Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, New York, NY, USA, 11–15 July 2022.
4. Tsai, C.Y.; Chuang, K.W.; Jen, H.Y.; Huang, H. A Tour Recommendation System Considering Implicit and Dynamic Information. *Appl. Sci.* **2024**, *14*, 9271. [\[CrossRef\]](#)
5. Safavi, S.; Jalali, M.; Houshm, M. Toward point-of-interest recommendation systems: A critical review on deep-learning Approaches. *Electronics* **2022**, *11*, 1998. [\[CrossRef\]](#)
6. Werneck, H.; Silva, N.; Viana, M.; Pereira, A.C.; Mourao, F.; Rocha, L. Points of interest recommendations: Methods, evaluation, and future directions. *Inf. Syst.* **2021**, *101*, 101789. [\[CrossRef\]](#)
7. Gupta, V.; Bedathur, S. Doing more with less: Overcoming data scarcity for poi recommendation via cross-region transfer. *ACM Trans. Intell. Syst. Technol. (TIST)* **2022**, *13*, 1–24. [\[CrossRef\]](#)
8. Perifanis, V.; Drosatos, G.; Stamatiatos, G.; Efraimidis, P.S. FedPOIRec: Privacy-preserving federated poi recommendation with social influence. *Inf. Sci.* **2023**, *623*, 767–790. [\[CrossRef\]](#)
9. Elahi, M.; Ricci, F.; Rubens, N. A survey of active learning in Collaborative Filtering recommender systems. *Comput. Sci. Rev.* **2016**, *20*, 29–50. [\[CrossRef\]](#)

10. Bobadilla, J.; Alonso, S.; Hernando, A. Deep learning architecture for Collaborative Filtering recommender systems. *Appl. Sci.* **2020**, *10*, 2441. [\[CrossRef\]](#)
11. An, J.; Jiang, W.; Li, G. Bidirectional trust-enhanced Collaborative Filtering for point-of-interest recommendation. *Sensors* **2023**, *23*, 4140. [\[CrossRef\]](#)
12. Davtalab, M.; Alesheikh, A.A. A POI recommendation approach integrating social spatio-temporal information into probabilistic Matrix Factorization. *Knowl. Inf. Syst.* **2021**, *63*, 65–85. [\[CrossRef\]](#)
13. Safavi, S.; Jalali, M. RecPOID: POI recommendation with friendship aware and deep CNN. *Future Internet* **2021**, *13*, 79. [\[CrossRef\]](#)
14. Guan, Y.; Lu, R.; Zheng, Y.; Shao, J.; Wei, G. Toward oblivious location-based K-Nearest Neighbor query in smart cities. *IEEE Internet Things J.* **2021**, *8*, 14219–14231. [\[CrossRef\]](#)
15. Gambs, S.; Killijian, M.O.; del Prado Cortez, M.N. Next place prediction using mobility Markov chains. In Proceedings of the First Workshop on Measurement, Privacy, and Mobility, New York, NY, USA, 10 April 2012.
16. Chen, Y.-C.; Thaipisutikul, T.; Shih, T.K. A learning-based POI recommendation with spatiotemporal context awareness. *IEEE Trans. Cybern.* **2020**, *52*, 2453–2466. [\[CrossRef\]](#)
17. Thaipisutikul, T.; Chen, Y.-N. An improved deep sequential model for context-aware POI recommendation. *Multimed. Tools Appl.* **2024**, *83*, 1643–1668. [\[CrossRef\]](#)
18. 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. [\[CrossRef\]](#)
19. Zhao, P.; Luo, A.; Liu, Y.; Xu, J.; Li, Z.; Zhuang, F.; Sheng, V.S.; Zhou, X. Where to go next: A spatio-temporal gated network for next poi recommendation. *IEEE Trans. Knowl. Data Eng.* **2020**, *34*, 2512–2524. [\[CrossRef\]](#)
20. Zuo, J.; Zhang, Y. Diff-DGMN: A Diffusion-Based Dual Graph Multiattention Network for POI Recommendation. *IEEE Internet Things J.* **2024**, *11*, 38393–38409. [\[CrossRef\]](#)
21. 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. [\[CrossRef\]](#)
22. Guo, Q.; Sun, Z.; Zhang, J.; Theng, Y.-L. An attentional Recurrent Neural Network for personalized next location recommendation. *Proc. Aaai Conf. Artif. Intell.* **2020**, *34*, 83–90. [\[CrossRef\]](#)
23. 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.
24. He, X.; He, W.; Liu, Y.; Lu, X.; Xiao, Y.; Liu, Y. ImNext: Irregular interval attention and multi-task learning for next POI recommendation. *Knowl.-Based Syst.* **2024**, *293*, 111674. [\[CrossRef\]](#)
25. Halder, S.; Lim, K.H.; Chan, J.; Zhang, X. Transformer-based multi-task learning for queuing time aware next poi recommendation. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*; Springer International Publishing: Cham, Switzerland, 2021.
26. Hong, Y.; Martin, H.; Raubal, M. How do you go where? improving next location prediction by learning travel mode information using transformers. In Proceedings of the 30th International Conference on Advances in Geographic Information Systems, Seattle, DC, USA, 1–4 November 2022.
27. Song, Y.; Liu, J.; Zhao, K. GETNext: Trajectory flow map enhanced transformer for next POI recommendation. In Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, 11–15 July 2022.
28. Wang, X.; Fukumoto, F.; Cui, J.; Suzuki, Y.; Li, J.; Yu, D. EEDN: Enhanced encoder-decoder network with local and global context learning for POI recommendation. In Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, Taipei, Taiwan, 23–27 July 2023.
29. Massimo, D.; Ricci, F. Combining reinforcement learning and spatial proximity exploration for new user and new POI recommendations. In Proceedings of the 31st ACM Conference on User Modeling, Adaptation and Personalization, New York, NY, USA, 26–29 June 2023.
30. Sun, K.; Qian, T.; Chen, T.; Liang, Y.; Nguyen, Q.V.H.; Yin, H. Where to go next: Modeling long-and short-term user preferences for point-of-interest recommendation. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34.
31. Yin, F.; Liu, Y.; Shen, Z.; Chen, L.; Shang, S.; Han, P. Next POI recommendation with dynamic graph and explicit dependency. *Proc. Aaai Conf. Artif. Intell.* **2023**, *37*, 4827–4834. [\[CrossRef\]](#)
32. Werneck, H.; Santos, R.; Silva, N.; Pereira, A.C.; Mourão, F.; Rocha, L. Effective and diverse POI recommendations through complementary diversification models. *Expert Syst. Appl.* **2021**, *175*, 114775. [\[CrossRef\]](#)
33. Safavi, S.; Jalali, M. DeePOF: A hybrid approach of deep convolutional neural network and friendship to Point-of-Interest (POI) recommendation system in location-based social networks. *Concurr. Comput. Pract. Exp.* **2022**, *34*, E6981. [\[CrossRef\]](#)
34. Lim, J.; Lee, S.; Li, H.; Bok, K.; Yoo, J. POI Recommendation Scheme Based on User Activity Patterns and Category Similarity. *Appl. Sci.* **2024**, *14*, 10997. [\[CrossRef\]](#)

35. Wang, D.; Chen, C.; Di, C.; Shu, M. Exploring behavior patterns for next-poi recommendation via graph self-supervised learning. *Electronics* **2023**, *12*, 1939. [[CrossRef](#)]
36. 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.
37. Huang, Z.; Xu, S.; Wang, M.; Wu, H.; Xu, Y.; Jin, Y. Human mobility prediction with causal and spatial-constrained multi-task network. *EPJ Data Sci.* **2024**, *13*, 22. [[CrossRef](#)]
38. Yang, D.; Fankhauser, B.; Rosso, P.; Cudre-Mauroux, P. Location prediction over sparse user mobility traces using RNNs. In Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, Online, 7–15 January 2021.
39. Zuo, J.; Zhang, Y. Collaborative trajectory representation for enhanced next POI recommendation. *Expert Syst. Appl.* **2024**, *256*, 124884. [[CrossRef](#)]
40. Chauhan, U.; Shah, A. Topic modeling using Latent Dirichlet Allocation: A survey. *Acm Comput. Surv. (CSUR)* **2021**, *54*, 1–35. [[CrossRef](#)]
41. Jelodar, H.; Wang, Y.; Yuan, C.; Feng, X.; Jiang, X.; Li, Y.; Zhao, L. Latent Dirichlet Allocation (LDA) and topic modeling: Models, applications, a survey. *Multimed. Tools Appl.* **2019**, *78*, 15169–15211. [[CrossRef](#)]
42. Perotte, A.; Wood, F.; Elhadad, N.; Bartlett, N. Hierarchically supervised Latent Dirichlet Allocation. *Adv. Neural Inf. Process. Syst.* **2011**, *24*.
43. Zhou, S.; Kan, P.; Huang, Q.; Silbernagel, J. A guided Latent Dirichlet Allocation approach to investigate real-time latent topics of Twitter data during Hurricane Laura. *J. Inf. Sci.* **2023**, *49*, 465–479. [[CrossRef](#)]
44. Yu, F.; Yan, H.; Chen, R.; Zhang, G.; Liu, Y.; Chen, M.; Li, Y. City-scale vehicle trajectory data from traffic camera videos. *Sci. Data* **2023**, *10*, 711. [[CrossRef](#)] [[PubMed](#)]
45. Yabe, T.; Tsubouchi, K.; Shimizu, T.; Sekimoto, Y.; Sezaki, K.; Moro, E.; Pentland, A. YJMob100K: City-scale and longitudinal dataset of anonymized human mobility trajectories. *Sci. Data* **2024**, *11*, 397. [[CrossRef](#)]
46. Wang, B.; Li, H.; Wang, W.; Wang, M.; Jin, Y.; Xu, Y. PG²Net: Personalized and Group Preferences Guided Network for Next Place Prediction. *IEEE Trans. Intell. Transp. Syst.* **2024**, *25*, 8655–8670. [[CrossRef](#)]
47. Yan, X.; Song, T.; Jiao, Y.; He, J.; Wang, J.; Li, R.; Chu, W. Spatio-temporal hypergraph learning for next POI recommendation. In Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, Taipei, Taiwan, 23–27 July 2023.

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.