

Article

Not peer-reviewed version

---

# A Multi-Feature Transition-Aware Framework for Next POI Recommendation

---

Oraya Sooknit , Jakkarin Suksawatchon , [Ureerat Suksawatchon](#) \*

Posted Date: 30 January 2026

doi: 10.20944/preprints202601.2411.v1

Keywords: Transition-Aware; Next POI; Check-in Pattern; Routine Behavior; Non-routine Behavior; Multi-Feature; Feature Effectiveness; Feature Synergy; Sustainable Tourism



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a [Creative Commons CC BY 4.0 license](#), which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

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.

Article

# A Multi-Feature Transition-Aware Framework for Next POI Recommendation

Oraya Sooknit , Jakkarin Suksawatchon  and Ureerat Suksawatchon \* 

Faculty of Informatics, Burapha University, Seansuk, Mueang, Chonburi, 20131, Thailand

\* Correspondence: ureerat@go.buu.ac.th; Tel.: +66-089-1416820

## Abstract

Next Point-of-Interest (POI) recommendation aims to predict a user's next location based on historical check-in data. However, real-world check-in records often contain uncertain check-ins, in which ambiguous spatial, temporal, or behavioral information obscures true mobility patterns and degrades prediction accuracy. To mitigate this issue, this study first learns user preferences from historical trajectories and adjusts transition importance based on temporal and spatial proximity, before modeling transition relationships using three complementary features: category, spatial area, and routine/non-routine behavior patterns. Based on transition probability analysis, feature-level dependencies in user mobility are systematically examined. The results indicate that these transition features contribute unequally to prediction performance, with area-based transitions being the most effective when considered individually. Nevertheless, their integration consistently yields the highest accuracy, highlighting the importance of transition-aware modeling. Experiments on two real-world datasets demonstrate that the proposed framework outperforms state-of-the-art methods in terms of Recall and NDCG, confirming the effectiveness of the approach.

**Keywords:** transition-aware; next POI; check-in pattern; routine behavior; non-routine behavior; multi-feature; feature effectiveness; feature synergy; sustainable tourism

## 1. Introduction

The increasing use of smartphones and social check-in platforms has facilitated the large-scale collection of user trajectory data through location-based social networking (LBSN) platforms, such as Foursquare, Facebook Places, and Google Maps, providing valuable insights into human mobility patterns and enabling personalized location recommendations. Traditional POI recommendation methods primarily focus on static user-location interactions without considering the sequential or temporal nature of human mobility [1,2]. However, with the emergence of large-scale location data, there has been a growing interest in next POI recommendation [3–5] which seeks to predict a user's next destination given their current or recent location. This task poses unique challenges as it must capture not only user preferences but also spatiotemporal transitions and dynamic behavioral patterns. Nevertheless, existing approaches remain insufficient to address the limitations caused by incomplete and sparse check-in data.

Recent studies in Next POI recommendation have explored advanced and adaptive techniques to improve performance and address key challenges in the domain. For example, the HCT model tackles uncertain check-in issues by exploiting category transitions across hierarchical layers [6], while POI2Vec considers user preferences for recent locations together with sequential transitions to enhance prediction accuracy [7]. Despite their effectiveness, most existing methods model user mobility using a single feature dimension, which is insufficient to capture the inherently multidimensional nature of real-world mobility behavior and thus limits accurate next POI prediction. Effective prediction requires the integration of multiple correlated features to capture both the habitual and dynamic aspects of user trajectories and to enhance model adaptability. To overcome the challenges of missing

or sparse information in user trajectories, we therefore incorporate three essential features—category, area, and pattern—that enable the model to learn users’ contextual behavior at a given moment, resulting in a more comprehensive representation of user behavior. The category feature reflects the user’s preference for specific types of POIs, such as restaurants, offices, or entertainment venues, capturing the semantic meaning of their check-ins. The area feature captures spatial correlations among frequently visited locations by identifying geographic clusters that represent meaningful zones such as home, work, or leisure areas. This spatial grouping helps mitigate data sparsity and enhances the understanding of users’ movement continuity within or across nearby regions. Meanwhile, the pattern feature characterizes the temporal and behavioral regularity of user activities, distinguishing between routine and non-routine mobility patterns [8]. By incorporating these three interrelated features, the framework can simultaneously model semantic preferences, spatial continuity, and behavioral dynamics—enabling a more adaptive and accurate prediction of the next POI.

The main contributions of this work are summarized as follows:

1. We design a sequential check-in–driven preference modeling framework that captures user preferences and mobility patterns to support personalized Next POI recommendation.
2. By leveraging three complementary features in combination, our framework effectively mitigates the uncertainty caused by incomplete or sparse check-in data.
3. Experimental results demonstrate that the proposed approach provides more reliable and accurate predictions on real-world datasets compared to state-of-the-art models.

## 2. Related Work

Over the past several years, research on next Point-of-Interest (Next POI) recommendation has gained increasing attention, driven by the growing availability of large-scale data from location-based social networks (LBSNs). Early approaches to POI recommendation primarily focused on modeling static user–location interactions, often employing techniques such as collaborative filtering, matrix factorization, and content-based methods. These models aimed to address challenges including data sparsity, evolving user preferences, and the need for personalized recommendations. However, their limited consideration of sequential dependencies and contextual factors restricted their effectiveness in predicting a user’s next location. Markov chain-based methods estimate transition probabilities between POI, providing a basic form of sequential awareness [9], but often suffer from data sparsity in large POI spaces. Early deep sequential models such as ST-RNN [10] and LSTPM [3] utilize LSTM networks to capture spatio-temporal dependencies in user trajectories, significantly outperforming Markov-based approaches in Next POI prediction

### 2.1. Transition-Aware Next POI Recommendation

In addition to focusing on sequential data learning, recent research has increasingly emphasized modeling the spatio-temporal transitions between locations to improve the accuracy and adaptability of Next POI prediction models. For example, the HCT model leverages category transitions across multiple layers to capture users’ preference transitions at varying levels of granularity [6], while POI2Vec jointly learns embeddings that reflect both recent user preferences and the sequential transition structure [7]. STA model [11], users and POIs are embedded into a shared “translation space”, and each spatiotemporal context is represented as a translation vector, further capture complex transition patterns, but they often rely on a single dominant feature and may be sensitive to incomplete or sparse check-in sequences. KGNext employs transition and interaction graphs to reveal both general mobility patterns and diverse user preferences for POIs and POI categories [12]. We summarize of difference transition-aware Next POI recommendation model as Table 1.

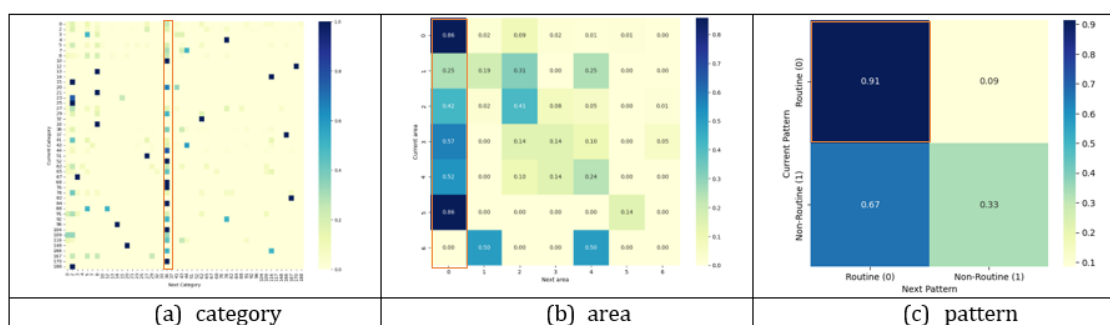
**Table 1.** Summary of difference transition-aware Next POI recommendation models.

Year	Model / Paper	Input Features	Transition Modeling
2017	POI2Vec [7]	POI, Region, Time	Hierarchical category transition
2019	HCT [6]	Category, Location	Hierarchical category transition
2019	RTPM [5]	Time, Category	Time-interval transition
2020	LSTPM [3]	Category, Time, Location	Sequential dependency learning
2024	KGNext [12]	Category, POI	Graph Transition
2025	AFNextPOI [8]	Category, Area, Pattern	Transition-aware correlation weighting
2025	MTF-POI (Proposed)	Category, Area, Pattern	Multi-feature transition matrix

## 2.2. Context-Aware and Multi-Feature Approaches

Recent studies have emphasized the need for incorporating multiple contextual features—such as category, spatial area, and temporal factors—to enhance prediction robustness under real-world conditions. Category transitions have proven effective for modeling semantic relationships between locations, while spatial clustering captures the geographical distribution of user movements. User behavioral patterns further reveal regularities that cannot be captured solely by spatial or categorical attributes. Frameworks such as AFNextPOI [8] integrate pattern-based features with area information, demonstrating improved performance by leveraging complementary contextual signals. However, existing methods often lack a unified mechanism to jointly model these heterogeneous features alongside explicit transition awareness between the current and next POIs. While transition-aware models have significantly advanced the field of next POI recommendation, most existing approaches still focus on a limited set of features, making them vulnerable to the effects of incomplete or sparse check-in data. Prior studies have seldom examined how category, area, and routine/non-routine patterns correlate between consecutive locations. Our analysis shows distinct dependencies, with patterns being the most deterministic, yet such feature-level relationships remain underexplored for improving next location prediction.

Figure 1 illustrates the relationships between check-in points from multiple perspectives. Only specific points, such as category 36, area 0, and the routine pattern, demonstrate a high likelihood of exhibiting relational dependence.

**Figure 1.** Transition probabilities between check-in points.

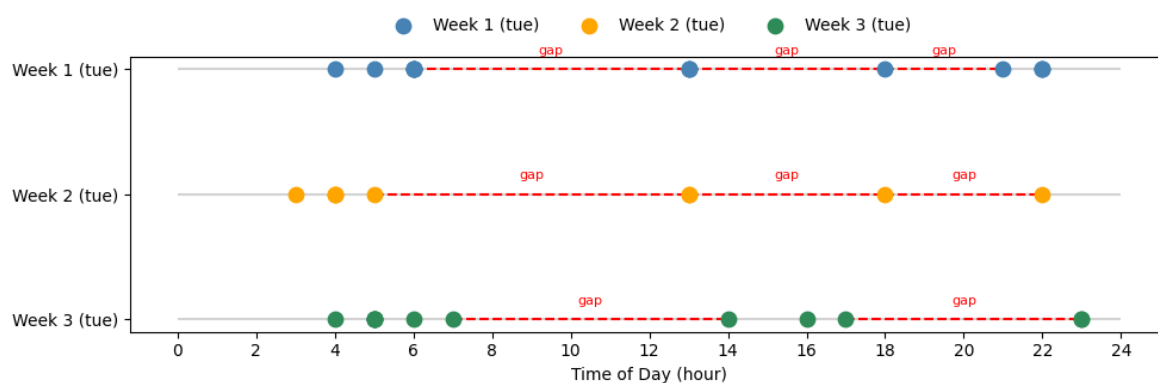
## 3. Problem Definition

### 3.1. Definition of Uncertain Check-In

**Uncertain check-in** refers to a location record whose spatial or temporal attributes are ambiguous, inaccurate, or missing, making it difficult to represent the user's true movement path. In addition, a check-in can also be considered uncertain when its behavioral pattern deviates from the user's normal routine, indicating irregular or unexpected mobility behavior.

In location-based social networks (LBSNs), user trajectories often contain incomplete or inconsistent information due to irregular check-in behavior. Such uncertainty may arise from factors including missing check-ins between consecutive locations, GPS noise, incorrect timestamps, or user behavior that deviates from habitual mobility patterns. These uncertain points introduce noise and discontinuity into the trajectory, leading to degraded model performance in sequential prediction. Therefore,

handling uncertain check-ins effectively is crucial for constructing reliable Next-POI recommendation models, especially when working with real-world datasets characterized by sparsity and irregular sampling. As illustrated in Figure 2, user check-in timeline on the same weekday across different weeks. Each line represents the same user's check-in behavior on Tuesday across three different weeks. The dashed red lines indicate long temporal gaps (> 2 hours) between consecutive check-ins, reflecting missing records and temporal sparsity even within the same weekday pattern.



**Figure 2.** User Check-in Timeline on Tuesday Across Different Weeks.

### 3.2. Problem of Uncertain Check-In

To address the uncertainty in user trajectories, recent studies such as the AHLTP model [13] focuses on predicting hidden or missing locations within a user's trajectory. By leveraging both venue classifications and sequential trajectory patterns, the model infers potentially unrecorded check-ins and predicts future locations, thereby enhancing the accuracy of mobility forecasting in LBSN applications. This approach provides valuable insight into handling trajectory incompleteness, suggesting that incorporating data imputation or hidden-location prediction into future Next POI frameworks could further improve model robustness and adaptability. Unlike models that attempt to infer hidden or missing check-ins, our proposed multi-feature framework learns the contextual correlations underlying user mobility. Inferring hidden locations requires strong assumptions about users' intentions and continuous transitions, which are often unreliable under real-world conditions where data sparsity and noise prevail. In contrast, our multi-feature learning design mitigates the limitations of missing or uncertain check-ins by modeling users' latent mobility preferences rather than explicitly reconstructing the missing points. This representation captures the underlying contextual dependencies among features, enabling the model to produce more robust and accurate predictions under trajectory uncertainty. As a result, it outperforms hidden-location inference methods that are often prone to noise, while maintaining high performance even with incomplete or irregular trajectories.

## 4. Methodology

### 4.1. Data Preprocessing

We assessed the performance of our model on two real-world datasets: Foursquare check-ins in New York (NYC) and Tokyo (TKY), collected between April 12, 2012, and February 16, 2013. Originally, this dataset served to explore the spatial-temporal regularities in user behaviour within LBSNs [14]. The summary of all the important notations in this paper is listed below in Table 2

Table 2. Notation and Descriptions.

Notation	Description
$U \in \{u_1, u_2, u_3, \dots, u_{ U }\}$	Set of users
$L \in \{l_1, l_2, l_3, \dots, l_{ L }\}$	Set of latitude and longitude pairs
$C \in \{c_1, c_2, c_3, \dots, c_{ C }\}$	Set of categories
$hr$	Set of hours $\{0, 1, 2, \dots, 23\}$
$d$	Set of days of week $\{0, 1, 2, \dots, 6\}$
$c(u, i)$	Category of candidate $i$ for user $u$
$a(u, i)$	Area of candidate $i$ for user $u$
$pt(u, i) \in \{R, N\}$	Indicates whether the point is routine (R) or non-routine (N)
$p(u, i)$	Vector $[l, c, h, d]$ of candidate $i$ for user $u$
$P$	Vector $[l, c, h, d]$ of candidate $i$ for all points
$tra(u, i)$	Trajectory of sequence $i$ for user $u$ is represented by category features .
$s(u, i)$	Sequence of candidate $i$ for user $u$
$h(u, i)$	Hidden state of candidate $i$ for user $u$

#### 4.2. MTF-POI Framework

We propose the MTF-POI framework, which is structured into four layers, as illustrated in Figure 3. These four interconnected layers collaboratively transform raw user check-in data into accurate Next Point-of-Interest recommendations. The first layer, the Embedded (User Preference) Layer, focuses on modeling sequential user behavior by transforming historical check-in trajectories into ordered sequences and learning temporal dependencies using an LSTM network. The output hidden state represents the user's latent mobility preference under the current context. The second layer, the Preference Identification and Weighting Layer, captures spatial and behavioral characteristics by clustering POIs into geographic areas using DBSCAN and identifying routine and non-routine movement patterns through Isolation Forest based on contextual features such as area, hour, and day. These components generate adaptive preference weights that reflect the user's spatial familiarity and behavioral regularity. The third layer, the Transition Modeling Layer, explicitly models transitions across multiple dimensions, including category-to-category, area-to-area, and pattern-to-pattern movements. The strength of these transitions is quantified using the Phi-coefficient, enabling the framework to emphasize meaningful movement correlations while suppressing weak or noisy relationships. Finally, the Recommendation Layer integrates user preferences and transition scores to produce ranked POI recommendations by filtering candidate locations, normalizing scores with a softmax function, and sorting the results to return the Top-K recommended POIs. This four-layer architecture enables effective integration of temporal, spatial, categorical, and behavioral information, resulting in robust and personalized Next-POI recommendations.

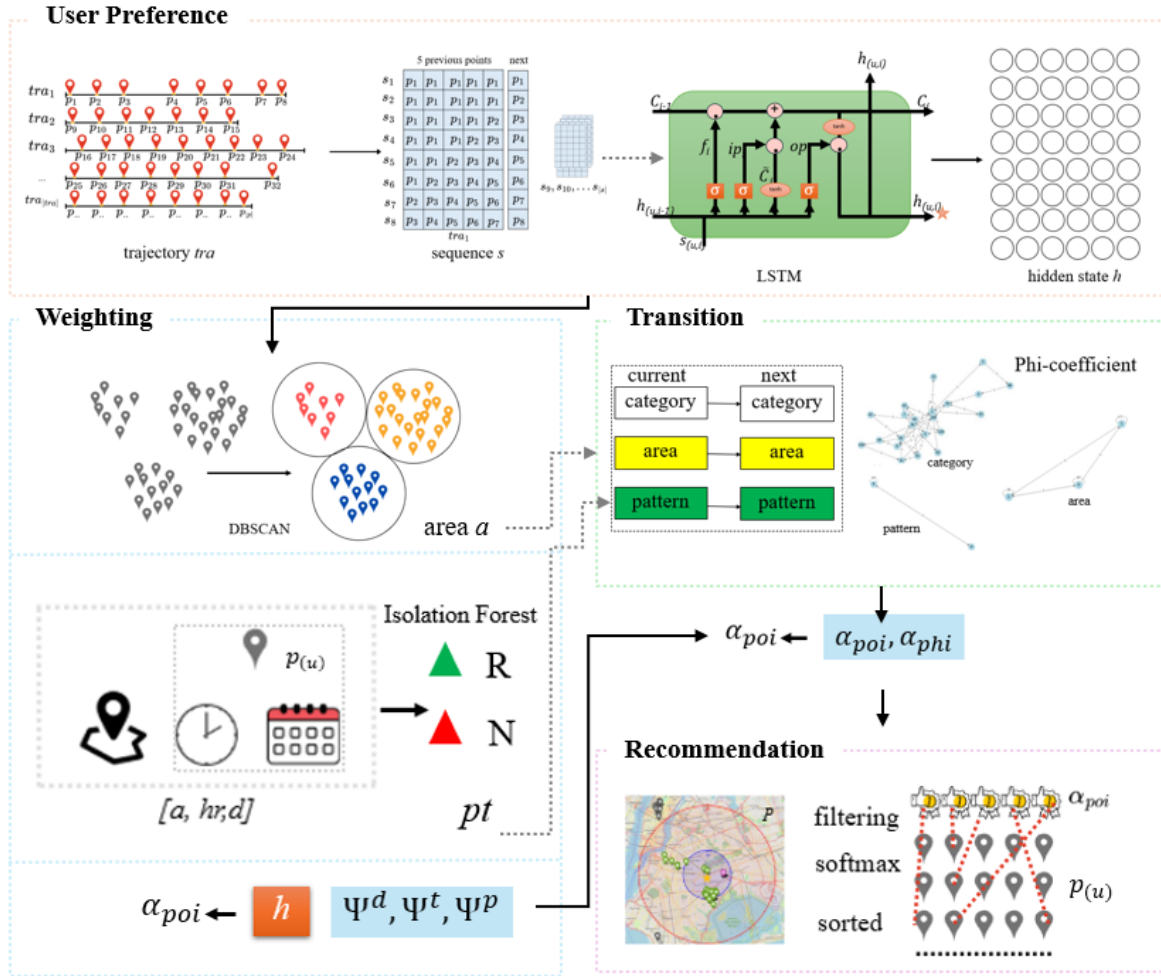


Figure 3. MTF-POI Framework.

#### 4.2.1. User Preference

To construct user trajectories, we segmented check-in data into individual sequences by splitting the trajectory whenever the time gap between consecutive check-ins exceeded 24 hours. For each resulting trajectory, we applied a sliding window of length 5 to generate input sequences for next-location prediction. The sequence was shifted by one position at a time until reaching the final check-in in the trajectory [15]. The entire sequence  $s$  was input into the LSTM model as categorical data to generate the hidden state for the next point presented in 1-3.

$$h_{(u,i)} = op_i \odot \tanh(C_i) \quad (1)$$

$$C_i = f_i \odot C_{(i-1)} + ip_i \odot \tilde{C}_i \quad (2)$$

$$op_i = \sigma(W_o \cdot [h_{(i-1)}, s_i] + b_o) \quad (3)$$

Where:

- $h_{(u,i)}$  denotes the hidden state at time step  $i$  for user  $u$ .
- $o_i$  represents the output gate activation at time step  $i$ .
- $\odot$  indicates element-wise multiplication.
- $\tanh$  denotes the hyperbolic tangent activation function.
- $C_i$  refers to the new cell state, combining the forgotten part of the past state and the new information.

- $f_i$  is the forget gate activation.
- $i_p$  denotes the input gate activation at time step  $i$ .
- $\tilde{C}_i$  represents the candidate values used to update the cell state.
- $\sigma$  denotes the sigmoid activation function.
- $W_o$  refers to the weight matrix for the output gate.
- $b_o$  is the bias term for the output gate.
- $h_{(i-1)}$  is the hidden state from the previous step.
- $s_i$  is the input index of category  $c$  at the current time step  $i$ .

We apply early stopping or train up to a maximum of 200 epochs for each user's model using a Python-based integrated computational framework built on TensorFlow/Keras.

#### 4.2.2. Weighting

The objective of this task is to learn a weight that reflects both the distance  $d$  and the time  $t$  from the current location. We compute the distance  $d$  between the current point and the next candidate point using the Haversine formula [16]. The distance  $d$  is used to weight all candidate points, as shown in Equation 4. The time difference  $t$ , computed between the current time and that of each candidate point, is used to assign weights to all candidates, as shown in the Equation 5. By setting  $\alpha = -1$ , the model assigns higher scores to locations with shorter distances and closer temporal proximity.

$$\Psi_{(u,i)}^d = \frac{\exp(\alpha \cdot d_i)}{\sum_{j=1}^{|h_u|} \exp(\alpha \cdot d_j)} \quad (4)$$

$$\Psi_{(u,i)}^t = \frac{\exp(\alpha \cdot t_i)}{\sum_{j=1}^{|h_u|} \exp(\alpha \cdot t_j)} \quad (5)$$

For candidate check-in points, we apply different weighting based on the pattern.  $\Psi_{(u,i)}^p$  denotes the pattern-aware weight assigned to the  $i$ -th check-in of user  $u$ . When the current check-in is classified as a routine pattern and its area belongs to the user's frequently visited areas ( $pop_{(u)}^a$ ), the weight is maximized to 2. Similarly, if the check-in is non-routine but its category is contained in the user's popular categories ( $pop_{(u)}^c$ ) [8], the weight is also set to 2 because it still reflects the user's stable preferences. For all remaining cases, the weight is set to 1 in order to avoid suppressing potentially valid locations. The weighting criteria are defined in Equation 6.

$$\Psi_{(u,i)}^p = \begin{cases} 2, & \text{if } pt_{(u,i)} = R \text{ and } a_{(u,i)} \in pop_{(u)}^a \\ 2, & \text{if } pt_{(u,i)} = N \text{ and } a_{(u,i)} \in pop_{(u)}^c \\ 1, & \text{otherwise} \end{cases} \quad (6)$$

#### Where:

- $pop_{(u)}^a$  denotes the set of popular areas for user  $u$ .
- $pop_{(u)}^c$  denotes the set of popular categories for user  $u$ .

The exponent term  $\Psi_{(u,i)}^p$  acts as a pattern-aware amplification factor. This mechanism allows the model to highlight POIs adaptively highlight POIs with reliable temporal-spatial-behavioral consistency. as shown in Equation 7.

$$\alpha_{poi} = \left( h_{(u,i)} + \Psi_{(u,i)}^d + \Psi_{(u,i)}^t \right) \Psi_{(u,i)}^p \quad (7)$$

#### 4.2.3. Transition

We cluster user areas using DBSCAN (Density-Based Spatial Clustering of Applications with Noise) [17] from  $p(u)$ . The result is represented as in Figure 4(a). Subsequently, we use  $a$ ,  $hr$ , and  $d$  to identify user patterns using the Isolation Forest algorithm [18] Figure 4(b). In the final step, category

$c$ , area  $a$ , and pattern  $pt$  are utilized to assess their correlation with the next candidate via the Phi coefficient as Figure 4(c). An overview of the process in this step is illustrated in Figure 4.

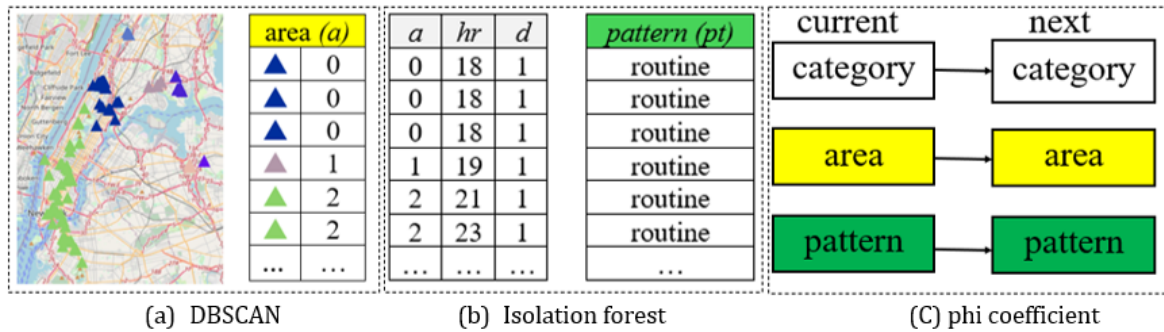


Figure 4. Overview of phi coefficient.

**DBSCAN:** The area  $a$  is divided using the DBSCAN algorithm, where  $N_\epsilon(p)$  denotes the set of neighboring points located within a radius  $\epsilon$  of point  $p$ . The parameter  $\epsilon$  represents the radius distance in DBSCAN, determining whether other points are close enough to be considered neighbors of the point in the dataset  $p_{(u,i)[l]}$ . The function  $\text{dist}(p, q)$  denotes the distance between points  $p$  and  $q$ , as defined in Equation 8. The notation  $|N_\epsilon(p)|$  in Equation 9 indicates the number of neighboring points. The parameter  $MinPts$  (Minimum number of neighbors) refers to the minimum number of neighboring points within the radius  $\epsilon$  required for a point to be regarded as a *core point* (here,  $\epsilon$  is set to 0.5). The density-reachable relationship among core points  $(p_1, p_2, p_3, \dots, p_n)$  is expressed in Equation 10.

$$N_\epsilon(p) = \{q \in p_{(u,i)[l]} \mid \text{dist}(p, q) \leq \epsilon\} \quad (8)$$

$$|N_\epsilon(p)| \geq MinPts \quad (9)$$

$$p_1 = p, \quad p_n = q, \quad p_{i+1} \in N_\epsilon(p_i) \quad (10)$$

**Isolation forest:** The vector  $x = [a, hr, d]$  represents the spatial area, hour, and day features, respectively, which are used as input to the Isolation Forest algorithm for anomaly detection. The resulting anomaly score is either 1 or  $-1$ , indicating whether a check-in point is classified as a *Routine* or *Non-routine* check-in point. A threshold value  $\theta$  is applied to separate the anomaly scores, which is determined based on an expected outlier ratio of 0.1, as shown in Equations 11-13.

$$x = [a, hr, d] \quad (11)$$

$$s(x, n) = 2 \frac{h(x)}{c(n)} \quad (12)$$

$$y(x) = \begin{cases} -1, & \text{if } s(x, n) \geq \theta \quad (\text{Non-routine}) \\ +1, & \text{if } s(x, n) < \theta \quad (\text{Routine}) \end{cases} \quad (13)$$

**Phi Coefficient:** The transition of relation from category, area, and pattern example at Figure 5. Each node represents a location, while the arrows indicate transitions between locations. The number on each arrow denotes the frequency of the movement. Furthermore, phi coefficient shows in Equation 14 by condition of current and next location for all check-in point.

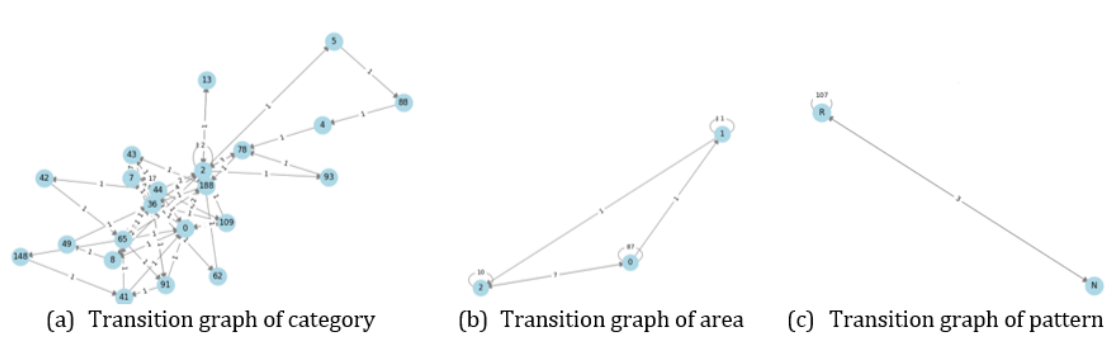


Figure 5. Transition graph of 1 user.

The phi coefficient measures the relationship under three vector conditions. A result of  $\phi = 1$  indicates a perfect positive relationship,  $\phi = -1$  indicates a complete negative relationship, and  $\phi = 0$  indicates no relationship.

$$\phi(\text{current\_loc}, \text{next\_loc}) = \frac{ad - bc}{\sqrt{(a+b)(c+d)(b+d)}} \quad (14)$$

Where:

- $a$  is the number of times where the current location = 1 and next location = 1.
- $b$  is the number of times where the current location = 1 and next location = 0.
- $c$  is the number of times where the current location = 0 and next location = 1.
- $d$  is the number of times where the current location = 0 and next location = 0.

The result is obtained by calculating the weighted average  $w$  of phi coefficients from all check-ins in the training set, as shown in Equation 15. In cases where no prior co-occurrence exists between check-in features, the phi coefficient is assigned a value of 0. The summary of  $\alpha_{poi}$  score is presented in Equation 16.

$$\alpha_{phi} = \frac{\sum_{i=1}^{|\phi|} (\phi_i \cdot w_i)}{\sum_{i=1}^{|\phi|} w_i} \quad (15)$$

Where:

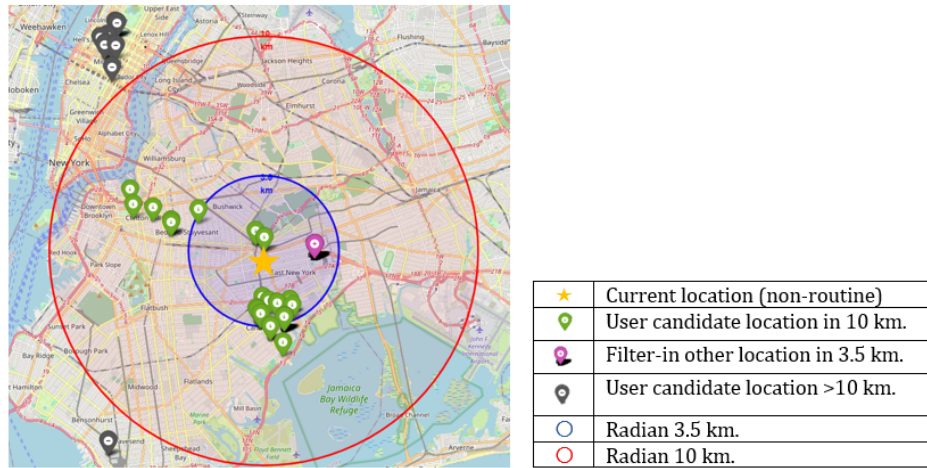
- $\phi = [\phi_c, \phi_a, \phi_{pt}]$  represents the phi coefficients for category, area, and pattern features.
- $w = [w_c, w_a, w_{pt}]$  Normalized Performance Weight applied to each phi component.

Finally, the overall  $\alpha_{poi}$  score is updated as follows:

$$\alpha_{poi(u,i)} = \alpha_{poi(u,i)} + \alpha_{phi(u,i)} \quad (16)$$

#### 4.2.4. Recommendation

We selected alternative locations from  $P$  within a 3.5 km radius of the average next location that matched the user's preferred categories at that time, specifically when the user was in a non-routine pattern. These selected check-in points were assigned the highest score for the user. Otherwise, points were filtered based on the user's average score as Algorithm 1. To eliminate distant locations, any candidate beyond 10 km was given the minimum score as Algorithm 2. Explore the candidate point of user as Figure 6.



**Figure 6.** Filtering points of interest based on a non-routine current location.

We use the SoftMax function to compute output probabilities, where  $e$  exponentiates each element to ensure positivity, and the resulting values sum to 1 as shown in Equation 17. In the final step, we sort the candidate points by score and retrieve the top@ $K$  ranked locations.

$$\text{Softmax}(\alpha_{poi(u,i)}) = \frac{e^{\alpha_{poi(u,i)} - \max(\alpha_{poi(u)})}}{\sum_{i=1}^n e^{\alpha_{poi(u,i)} - \max(\alpha_{poi(u)})}} \quad (17)$$

Where:

- $\sum_{i=1}^n \alpha_{poi(u,i)} = 1$  ensures that all output probabilities sum to one.
- $n$  denotes the total number of candidate  $\alpha_{poi(u)}$  values.

---

#### Algorithm 1 Filter-In Cross-User Candidate POIs Based on User's Popular Categories

---

**Require:** User  $u$ ; current check-in  $(l_u, hr_u)$ ;

- 1: user  $u$ 's historical check-ins  $p_u$ ;
- 2: check-in log of other users  $P$ ;
- 3: radius threshold  $r$  (e.g., 3.5 km);
- 4: time similarity threshold  $\tau_{hr}$ ;
- 5: preference scoring function  $p_u[l_i, hr_i, c_i].score = \alpha_{poi(u,i)}$

**Ensure:** Filtered candidate POI set with scores  $fn_{cat}$

- 6: Initialize  $fn_{cat} \leftarrow \emptyset$
  - 7: Compute user popular categories under current time context:
  - 8:  $pop_u \leftarrow \text{POPULARCATEGORY}(u, hr_u, p_u)$
  - 9: Compute user average and maximum preference scores:
  - 10:  $\bar{s}_u \leftarrow \text{AVERAGESCORE}(u, p_u)$
  - 11:  $s_u^{max} \leftarrow \text{MAXSCORE}(u, p_u)$
  - 12: **for all** check-in  $(i, l_i, hr_i, c_i)$  in  $P$  **do**
  - 13:      $d \leftarrow \text{GEODISTANCE}(l_u, l_i)$
  - 14:      $\Delta t \leftarrow \text{TIMEGAP}(hr_u, hr_i)$
  - 15:     **if**  $d \leq r$  **and**  $\Delta t \leq \tau_{hr}$  **and**  $c_i \in pop_u$  **then**
  - 16:         **if**  $l_i \in p_u$  **then**
  - 17:              $\alpha_{poi(u,i)} \leftarrow s_u^{max}$      ▷ duplicate with user's own POI: assign max score
  - 18:         **else**
  - 19:              $\alpha_{poi(u,i)} \leftarrow \bar{s}_u$      ▷ new cross-user POI: assign average score
  - 20:         **end if**
  - 21:          $fn_{cat} \leftarrow fn_{cat} \cup \{(l_i, s_i)\}$      ▷ filter-in this POI with its score
  - 22:     **end if**
  - 23: **end for**
  - 24:  $fn_{cat} \leftarrow \text{UNIQUEPOISWITHMAXSCORE}(fn_{cat})$
  - 25: **return**  $\alpha_{poi(u,i)}$
-

**Algorithm 2** Filter-Out Low-Preference and Distant POIs from User History**Require:** User  $u$ ; current check-in  $(l_u, hr_u)$ ;

- 1: user  $u$ 's historical check-ins  $p_u$ ;
- 2: maximum distance threshold  $R_{\max}$  (e.g., 10 km);
- 3: minimum score  $s_u^{\min}$

**Ensure:** Filtered personal POI set with scores  $p_u.score = \alpha_{poi(u,i)}$ 

- 4: Compute user average preference score:  $\bar{s}_u \leftarrow \text{AVERAGESCORE}(u, p_u)$
- 5: Initialize  $fn_u \leftarrow \emptyset$
- 6: **for all** check-in  $(j, l_j, hr_j, c_j)$  in  $p_u$  **do**
- 7:    $d_j \leftarrow \text{GEODISTANCE}(l_u, l_j)$
- 8:   **if**  $d_j > R_{\max}$  **then**
- 9:      $\alpha_{poi(u,i)} \leftarrow s_u^{\min}$                    ▷ eliminate distant locations by assigning minimum score
- 10:    **continue**                                   ▷ do not include this POI in  $fn_u$
- 11:   **end if**
- 12: **end for**
- 13: **return**  $\alpha_{poi(u,i)}$

### 4.3. Experiment

We evaluate the performance of all methods using standard metrics, namely Recall@K and NDCG@K, based on the top-K recommended POIs for each user [19], where  $K = \{1, 3, 5\}$ . The definitions for these evaluation metrics are given in Equations 18-20.

$$\text{Recall@K} = \frac{1}{|U|} \sum_{u=1}^{|U|} \frac{|\text{Recall}_u@K(r_u \cap t_u)|}{|t_u|} \quad (18)$$

**Where:**

- $r_u$  is the set of top-K POIs recommended for user  $u$ .
- $t_u$  is the set of ground-truth POIs actually visited by user  $u$  at the next location in the test set.

$$\text{NDCG@K} = \frac{1}{|U|} \sum_{u=1}^{|U|} \frac{\text{DCG}_u@K}{\text{IDCG}_u@K} \quad (19)$$

$$\text{DCG}_u@K = \sum_{i=1}^N \frac{2^{rel_i} - 1}{\log_2(i + 1)} \quad (20)$$

**Where:**

- IDCG is the maximum possible DCG for a given recommendation list.
- DCG (Discounted Cumulative Gain) measures the quality of ranked results.
- $rel_i$  denotes the relevance score of the item at position  $i$ .
- $N$  is the number of correctly recommended POIs.

We compare our model under different feature configurations in the process coefficient. Specifically, four model variants are evaluated: **MTF-c**, which incorporates only category features; **MTF-a**, which integrates area features; **MTF-pt**, which emphasizes user check-in pattern features; and the full **MTF-POI** model, which combines all features (category, area, and pattern) into a unified framework.

The experimental results are presented in Table 3. It can be observed that integrating multiple features yields better overall performance, as the combined model (**MTF-POI**) consistently outperforms the individual feature-based variants across all evaluation metrics.

**Table 3.** Comparison of model performance with coefficients from different feature configurations.

Metric	NYC Dataset				TKY Dataset			
	MTF-c	MTF-a	MTF-pt	MTF-POI	MTF-c	MTF-a	MTF-pt	MTF-POI
<b>Recall@1</b>	0.1715	<u>0.2084</u>	0.1662	<b>0.2480</b>	0.1655	<u>0.2300</u>	0.2021	<b>0.2430</b>
<b>Recall@3</b>	0.2801	<u>0.3799</u>	0.3522	<b>0.3712</b>	0.3116	<u>0.3955</u>	0.3801	<b>0.4244</b>
<b>Recall@5</b>	0.4112	<u>0.4634</u>	0.4409	<b>0.4902</b>	0.4158	<u>0.5144</u>	0.4945	<b>0.5409</b>
<b>NDCG@1</b>	0.1715	<u>0.2084</u>	0.1662	<b>0.2480</b>	0.1655	<u>0.2300</u>	0.2021	<b>0.2430</b>
<b>NDCG@3</b>	0.2203	<u>0.2919</u>	0.2606	<b>0.3007</b>	0.2370	<u>0.3161</u>	0.2958	<b>0.3354</b>
<b>NDCG@5</b>	0.2724	<u>0.3251</u>	0.2959	<b>0.3482</b>	0.2779	<u>0.3634</u>	0.3412	<b>0.3814</b>

### Feature Effectiveness and Synergy in Multi-Feature Transitions

As shown in Table 3, the area-based variant (MTF-a) achieves the best performance among single-feature models, outperforming both the category-based (MTF-c) and pattern-based (MTF-pt) configurations across all evaluation metrics. This finding confirms that spatial correlation is a dominant factor influencing user mobility and that clustering POIs into representative areas enables the model to better capture users' movement tendencies within geographically coherent regions.

More importantly, the combined model (MTF-POI) that integrates category, area, and pattern features yields the highest accuracy overall. Specifically, **Recall@1** increases from **0.2084** (MTF-a) to **0.2480** for NYC and from **0.2300** to **0.2430** for TKY, indicating an additional gain of +3.96 and +1.30 percentage points, respectively. This improvement highlights a *synergistic interaction* among the three features: the *area* feature provides spatial stability, the *category* feature contributes semantic context, and the *pattern* feature refines temporal and behavioral consistency. Together, they enable the model to precisely predict the most probable next POI at the top-1 rank while maintaining steady gains in higher  $K$  metrics (Recall@3, Recall@5, and NDCG).

**Total Performance Score: TPS** as Equation 21 was introduced to provide an integrated evaluation of model performance across multiple metrics and datasets. Specifically, the TPS summary in Table 4 reveals consistent performance trends across feature configurations. Among the evaluated variants, MTF-a achieves the highest TPS on both NYC and TKY datasets, resulting in the largest overall TPS score. This indicates that the attribute-based feature configuration contributes the most comprehensive performance improvements across all ranking levels and metrics. In contrast, MTF-pt consistently outperforms MTF-c, suggesting that personalized temporal patterns capture user mobility behavior more effectively than spatial clustering alone. Notably, the ordering of TPS values (MTF-a > MTF-pt > MTF-c) remains stable across both datasets, demonstrating the robustness of feature importance rankings and reinforcing the generalizability of the proposed feature design. Overall, these results validate the effectiveness of incorporating richer contextual and temporal signals to enhance next POI prediction performance.

$$TPS_f = \sum_{d \in \{NYC, TKY\}} \sum_{k \in \{1, 3, 5\}} \left( \text{Recall}@k_f^{(d)} + \text{NDCG}@k_f^{(d)} \right) \quad (21)$$

**Table 4.** Total Performance Score (TPS) of Feature Variants Across NYC and TKY.

Feature	NYC TPS	TKY TPS	Total TPS
MTF-c	1.5270	1.5733	3.1453
MTF-a	<b>1.8771</b>	<b>2.0494</b>	<b>3.9265</b>
MTF-pt	1.6820	1.9158	3.5978

**Normalized performance Weight: NPW** as Equation 22 quantifies the relative contribution of each feature by integrating its predictive performance across both datasets and all evaluation metrics. First, the aggregate score  $NPW_f$  is computed by averaging 12 values: Recall@1, Recall@3, Recall@5, and NDCG@1, NDCG@3, NDCG@5 from the NYC and TKY datasets using a double summation

over the datasets  $d$  and metrics  $k$ . This results in a unified performance score that reflects the overall effectiveness of the feature  $f \in \{c, a, pt\}$  as Equation 22.

$$NPW_f = \frac{1}{12} \sum_{d \in \{NYC, TKY\}} \sum_{k \in \{1,3,5\}} \left( \text{Recall}@k_f^{(d)} + \text{NDCG}@k_f^{(d)} \right) \quad (22)$$

Based on the Total Performance Score (TPS) aggregated across both datasets, the area-based variant (**MTF-a**) demonstrates the strongest overall effectiveness, achieving a perfect TPS of 3.9265. This result indicates that the **Area Feature** is the most stable and informative signal for characterizing user mobility, as it consistently captures spatial correlations across diverse urban environments. In contrast, the pattern-based model (**MTF-pt**) attains a moderate TPS of approximately 3.5978, suggesting that the **Pattern Feature** contributes meaningful behavioral cues but remains less dominant than spatial information. Meanwhile, the category-based configuration (**MTF-c**) yields the lowest TPS at around 3.1453, implying that the **Category Feature** alone is insufficient for modeling users' movement behavior and lacks the representational strength required for accurate POI transition learning. Although the category feature contributes to the overall performance, its effectiveness is generally weaker and more unstable compared to the area and pattern features. To reflect this observation and reduce the influence of noisy or weak category signals, we introduce a penalty factor  $\lambda_c = 0.3$  to downscale the raw performance score of the category feature before weight normalization. The weight  $w_f$  is then obtained by normalizing  $S_f$  with respect to the total score of all features, ensuring that the weights are comparable and collectively sum to one. Finally, the constraint  $w_c + w_a + w_{pt} = 1$  guarantees that the category, area, and pattern features together account for the full proportion of the performance contribution of the model as Equations 23-24.

$$w_f = \frac{S'_f}{S'_c + S'_a + S'_{pt}}, \quad S'_c = \lambda_c \cdot S_c \quad (23)$$

$$w_c + w_a + w_{pt} = 1 \quad (24)$$

We compared our proposed framework, **MTF-POI**, with several baseline methods commonly used for Next Point-of-Interest (Next-POI) recommendation. To ensure reproducibility and prevent implementation bias in future research, we directly adopted the reported performance results from two benchmark datasets presented in the following studies:

- **LSTM** [20]: A Python-based integrated computational framework built on TensorFlow/Keras, designed for sequential data, similar to our model.
- **GRU** [21]: A Python-based integrated computational framework built on TensorFlow/Keras, designed for sequential data, similar to our model.
- **LSTPM** [3,5]: A model that addresses the limitations of existing POI recommendation methods by incorporating both long-term and short-term user preferences.
- **RTPM** [5]: A model that analyses long-term and short-term user preferences, real-time user behaviors, probability estimation, and recommendation modules.
- **AFNextPOI**: [8] A framework that categorizes user travel patterns into different cases and applies distinct evaluation methods for each, enabling better handling of data uncertainty.
- **MTF-POI (Proposed)**: The proposed framework extends AFNextPOI by integrating multi-feature transition awareness between the current and next locations.

Results from the proposed methodology and baseline comparisons are presented in Table 5. When compared with AFNextPOI, certain metrics such as Recall@3, Recall@5, and NDCG@5 show slightly lower values for MTF-POI. However, MTF-POI achieves higher Recall@1, which is considered the most important evaluation metric for Next-POI recommendation.

Table 5. Result of experiment with baseline Next POI model.

Model	NYC Dataset					TKY Dataset				
	R@1	R@3	R@5	N@3	N@5	R@1	R@3	R@5	N@3	N@5
LSTM	0.1161	0.2621	0.3730	0.1891	0.2337	0.1080	0.2480	0.3514	0.1837	0.2255
GRU	0.0739	0.1974	0.2983	0.1462	0.1814	0.0601	0.1828	0.2891	0.1262	0.1694
LSTPM	0.1836	0.3087	0.3707	0.2559	0.2814	0.2088	0.3492	0.4135	0.2902	0.3168
RTPM	0.1944	0.3182	0.3752	0.2663	0.2898	0.2143	0.3504	0.4151	0.2934	0.3201
AFNextPOI	0.2137	0.3851	0.5011	0.2989	0.3453	0.2352	0.4033	0.4786	0.3228	0.3524
<b>MTF-POI (Proposed)</b>	<b>0.2480</b>	<b>0.3712</b>	<b>0.4901</b>	<b>0.3007</b>	<b>0.3482</b>	<b>0.2430</b>	<b>0.4244</b>	<b>0.5409</b>	<b>0.3354</b>	<b>0.3814</b>
<i>Improvement</i>	+3.43%	-1.39%	-1.10%	+0.18%	+0.29%	+0.78%	+2.11%	+6.23%	+1.26%	+2.90%

Note: R:recall, N:NDCG.

#### 4.3.1. Influence of Multifeature Feature Transition

The consistent superiority of MTF-POI across both datasets demonstrates that multi-feature transition awareness generalizes effectively under different spatial densities and urban structures. Hence, integrating all three correlated features not only enhances overall recall and ranking quality but also strengthens the model's adaptability to both routine and non-routine user behaviors.

The bar chart as Figure 7 presents a comparison of Recall@1 performance across four model version (MTF-c, MTF-a, MTF-pt, and MTF-POI) on two datasets (NYC and TKY). Each dataset has four group of bars, representing the performance of the different version.

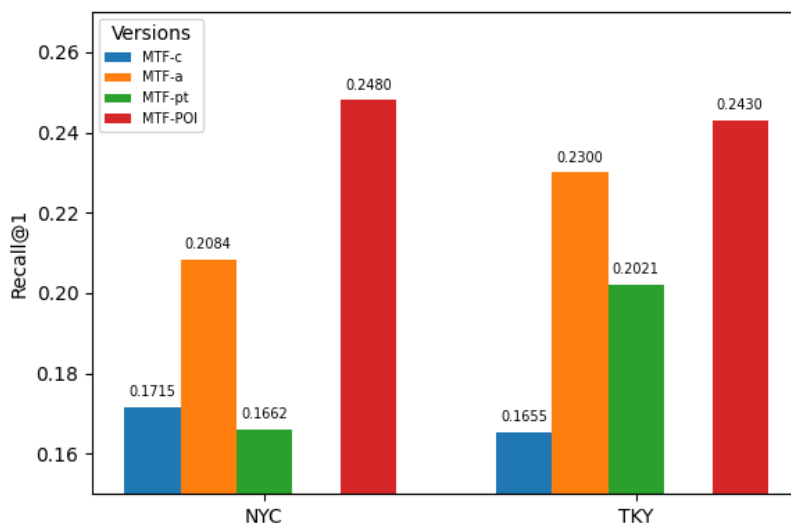


Figure 7. Comparison of Recall@1 Across Versions.

From the results, MTF-POI consistently achieves the highest Recall@1 in both datasets. Among the three versions (MTF-c, MTF-a, and MTF-pt), MTF-a consistently achieves the highest Recall@1 on both the NYC and TKY datasets. This indicates that the feature-enhanced version (MTF-a) provides a stronger performance compared to the category-based (MTF-c) and pattern-based (MTF-pt) versions. By combining the three features and applying a weighted average for scoring, the performance is further improved compared to the MTF-POI version.

The contrasting results of MTF-pt between NYC and TKY reflect differences in mobility regularity and city structure. In the NYC dataset, MTF-pt performs slightly worse than MTF-c because the routine/non-routine pattern signal is not strongly correlated with the actual next-location transitions. NYC exhibits high mobility diversity, where users frequently visit new places, change routes dynamically, and have irregular schedules. As a result, binary pattern-type labels (routine vs. non-routine) provide little predictive power and may even introduce noise—causing MTF-pt to underperform compared to the simpler category-based feature.

In contrast, TKY displays much more structured and repeatable mobility behavior, driven by predictable commuting flows, concentrated activity zones, and higher regularity in daily routines. In such an environment, pattern-type information becomes highly valuable: whether a movement

is routine or non-routine strongly influences the likelihood of the next POI. This makes MTF-pt substantially outperform MTF-c in the TKY dataset.

#### 4.3.2. Influence of Hidden Location Attraction

The MTF-c-AHLTP follows the same modeling process as MTF-c, where next-POI prediction is based on category transitions. The key difference is that MTF-c-AHLTP augments user trajectories with hidden locations inferred from category associations using the AHLTP [13] approach prior to model training.

The comparison between MTF-c and MTF-c-AHLTP in Figure 8 shows that incorporating hidden locations does not improve top-rank accuracy or ranking quality. While MTF-c-AHLTP slightly increases candidate coverage at higher K values, it introduces noise into category transitions, leading to lower Recall@1 and NDCG@K. These results suggest that, for large-scale next-POI prediction, using only real check-in transitions is more effective than augmenting trajectories with inferred locations.

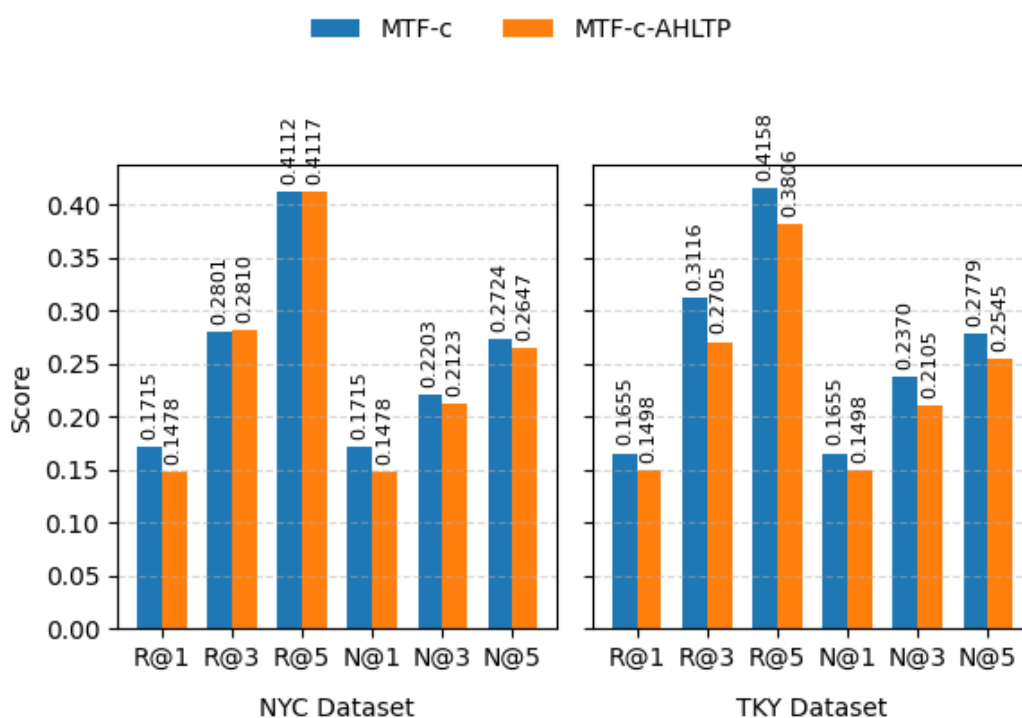


Figure 8. Comparison of hidden-location attraction between MTF-c and MTF-c-AHLTP.

More importantly, this result indicates that addressing uncertain check-ins through feature transfer is more effective than explicitly inserting inferred locations into user trajectories. Rather than modifying the original trajectory structure, feature transfer enables the model to adaptively reweight or transform existing features based on contextual and behavioral information, thereby mitigating uncertainty without amplifying error propagation. In contrast, hidden location insertion alters transition sequences by introducing inferred check-ins that may not reflect users' true short-term intent, leading to degraded ranking performance.

These findings suggest that, for large-scale next-POI prediction, modeling uncertainty at the feature level provides a more robust solution than augmenting trajectories with inferred locations.

## 5. Discussion

The performance differences observed in Table 5 can be explained by how each component of the proposed framework interacts with user mobility characteristics. In particular, improvements in top-rank accuracy are closely related to the way short-term intent and contextual uncertainty are handled across layers, rather than to the reconstruction of missing trajectory points.

At the sequence level, the preference layer based on LSTM captures temporal dependencies among consecutive check-ins, which provides a compact representation of users' recent mobility context. This representation reduces sensitivity to sporadic or irregular check-ins and allows the model to focus on short-term decision patterns. In contrast, models that directly operate on augmented trajectories may propagate uncertainty introduced at earlier steps, especially when inferred locations are not aligned with users' immediate intent.

The weighting layer further influences performance by differentiating candidate POIs according to spatial and behavioral contexts. Area-based clustering enables the model to emphasize movements within frequently visited regions, while the distinction between routine and non-routine patterns adjusts the influence of regular and exploratory behaviors. This mechanism helps explain why the proposed approach remains stable under heterogeneous user behaviors, whereas methods that rely on fixed trajectory structures are more affected by noisy or ambiguous check-ins.

Differences in ranking quality are mainly attributable to the transition layer. By integrating category, area, and pattern transitions and refining their contributions using the Phi coefficient, the model selectively amplifies meaningful correlations while suppressing weak or spurious ones. This explains the observed gains in NDCG@K, as relevant POIs are more likely to be ranked higher when multiple contextual signals consistently support the transition.

A notable contrast can be observed when comparing MTF-c with MTF-c-AHLTP. Although augmenting trajectories with hidden locations slightly increases candidate coverage at higher K values, it also alters category transition distributions by introducing inferred check-ins. These inferred transitions may not correspond to users' actual short-term choices, leading to degraded top-rank accuracy and ranking quality. This comparison suggests that uncertain check-ins are better handled through feature-level adaptation across layers than through explicit trajectory augmentation.

## 6. Conclusions

This study presented a multi-feature framework for next point-of-interest recommendation that addresses the challenges of sparse, heterogeneous, and uncertain check-in data. By leveraging temporal user preferences and multiple contextual signals, the proposed approach achieves consistent improvements over state-of-the-art baselines on two large-scale real-world datasets.

The experimental results demonstrate that integrating complementary mobility perspectives leads to more accurate and reliable next-POI predictions than relying on a single modeling dimension or trajectory reconstruction strategies. In particular, the findings highlight that uncertainty in user mobility can be effectively handled without explicitly inserting inferred or hidden locations into trajectories. Instead, preserving observed check-ins while incorporating contextual information yields better top-rank accuracy and ranking quality.

Beyond performance improvements, this work provides insights into human mobility behavior by showing that routine and non-routine movements can be jointly modeled within a unified recommendation framework. These insights support the practical applicability of the proposed method in real-world location-based services, where user behavior is dynamic and often irregular.

Future work will explore adaptive feature weighting strategies and cross-domain transfer learning to further enhance generalization across different cities and user populations. Additionally, incorporating external contextual signals, such as social or environmental factors, may further improve robustness under highly uncertain mobility scenarios.

**Author Contributions:** Conceptualization, O.S. and J.S.; methodology, O.S.; software, O.S.; validation, O.S., J.S., and U.S.; formal analysis, O.S.; investigation, O.S., and U.S.; resources, O.S., J.S. and U.S.; data curation, O.S., J.S. and U.S.; writing-original draft, O.S., writing-review & editing, O.S., J.S., and U.S.; visualization, O.S., supervision, J.S. and U.S.; project administration, J.S. and U.S., funding acquisition, O.S., J.S. and U.S.; Corresponding author: U.S. (email: ureerat@go.buu.ac.th). All authors have read and agreed to the published version of the manuscript.

**Acknowledgments:** The authors would like to express their gratitude to the Thailand scholarship through Rajamangala University of Technology Srivijaya for providing the Ph.D. Scholarship to O.S. Additionally, this study was supported by the Graduate School, Burapha University.

**Data Availability Statement:** The Foursquare dataset was collected from Foursquare. It can be accessed via: <https://sites.google.com/site/yangdingqi/home/foursquare-dataset>.

## References

1. Zhao, P.; Zhu, H.; Liu, Y.; Li, Z.; Xu, J.; Sheng, V.S. Where to go next: A spatio-temporal lstm model for next poi recommendation. *arXiv preprint arXiv:1806.06671* **2018**.
2. 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 Transactions on Knowledge and Data Engineering* **2020**, *34*, 2512–2524.
3. 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 Proceedings of the AAAI conference on artificial intelligence, 2020, Vol. 34, pp. 214–221.
4. Wu, Y.; Li, K.; Zhao, G.; Qian, X. Personalized long-and short-term preference learning for next POI recommendation. *IEEE Transactions on Knowledge and Data Engineering* **2020**, *34*, 1944–1957.
5. Liu, X.; Yang, Y.; Xu, Y.; Yang, F.; Huang, Q.; Wang, H. Real-time POI recommendation via modeling long-and short-term user preferences. *Neurocomputing* **2022**, *467*, 454–464.
6. Zhang, L.; Sun, Z.; Zhang, J.; Kloeden, H.; Klanner, F. Modeling hierarchical category transition for next POI recommendation with uncertain check-ins. *Information Sciences* **2020**, *515*, 169–190.
7. Feng, S.; Cong, G.; An, B.; Chee, Y.M. Poi2vec: Geographical latent representation for predicting future visitors. In Proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence, 2017, Vol. 31.
8. Sooknit, O.; Suksawatchon, J.; Suksawatchon, U. Awareness of Check-In Patterns for an Adaptive Framework in Next POI Recommendation. *ECTI Transactions on Computer and Information Technology (ECTI-CIT)* **2025**, *19*, 645–655.
9. Gambis, S.; Killijian, M.O.; del Prado Cortez, M.N. Next place prediction using mobility markov chains. In Proceedings of the Proceedings of the first workshop on measurement, privacy, and mobility, 2012, pp. 1–6.
10. Liu, Q.; Wu, S.; Wang, L.; Tan, T. Predicting the next location: A recurrent model with spatial and temporal contexts. In Proceedings of the Proceedings of the AAAI conference on artificial intelligence, 2016, Vol. 30.
11. Qian, T.; Liu, B.; Nguyen, Q.V.H.; Yin, H. Spatiotemporal representation learning for translation-based POI recommendation. *ACM Transactions on Information Systems (TOIS)* **2019**, *37*, 1–24.
12. Kong, X.; Chen, Z.; Li, J.; Bi, J.; Shen, G. Kgnext: Knowledge-graph-enhanced transformer for next poi recommendation with uncertain check-ins. *IEEE Transactions on Computational Social Systems* **2024**, *11*, 6637–6648.
13. Bahgat, E.M.; Abo-alian, A.; Rady, S.; Gharib, T.F. Efficient Trajectory Prediction Using Check-In Patterns in Location-Based Social Network. *Big Data and Cognitive Computing* **2025**, *9*, 102.
14. Yang, D.; Zhang, D.; Zheng, V.W.; Yu, Z. Modeling user activity preference by leveraging user spatial temporal characteristics in LBSNs. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* **2014**, *45*, 129–142.
15. Sooknit, O.; Suksawatchon, J.; Suksawatchon, U. Next Point of Interest Recommendation Using Adaptive Weights for Specific Behavioral Patterns. In Proceedings of the 2024 28th International Computer Science and Engineering Conference (ICSEC). IEEE, 2024, pp. 1–6.
16. Panigrahi, N. *Computing in geographic information systems*; CRC Press, 2014.
17. Ram, A.; Jalal, S.; Jalal, A.S.; Kumar, M. A density based algorithm for discovering density varied clusters in large spatial databases. *International Journal of Computer Applications* **2010**, *3*, 1–4.
18. Al Farizi, W.S.; Hidayah, I.; Rizal, M.N. Isolation forest based anomaly detection: A systematic literature review. In Proceedings of the 2021 8th International Conference on Information Technology, Computer and Electrical Engineering (ICITACEE). IEEE, 2021, pp. 118–122.
19. Schütze, H.; Manning, C.D.; Raghavan, P. *Introduction to information retrieval*; Vol. 39, Cambridge University Press Cambridge, 2008.

20. Bhandari, H.N.; Rimal, B.; Pokhrel, N.R.; Rimal, R.; Dahal, K.R. LSTM-SDM: An integrated framework of LSTM implementation for sequential data modeling. *Software Impacts* **2022**, *14*, 100396.
21. Dey, R.; Salem, F.M. Gate-variants of gated recurrent unit (GRU) neural networks. In Proceedings of the 2017 IEEE 60th international midwest symposium on circuits and systems (MWSCAS). IEEE, 2017, pp. 1597–1600.

**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.