CLIFT: Cross-City Mobility-Derived Lifestyle Pattern Transfer for Improved Multi-City Next Location Prediction

HARU TERASHIMA, Graduate School of Engineering, Nagoya University, Nagoya, Japan
NAOKI TAMURA, Graduate School of Engineering, Nagoya University, Nagoya, Japan
KAZUYUKI SHOJI, Graduate School of Engineering, Nagoya University, Nagoya, Japan
TAHERA HOSSAIN, Graduate School of Engineering, Nagoya University, Nagoya, Japan
SHIN KATAYAMA, Graduate School of Engineering, Nagoya University, Nagoya, Japan
KENTA URANO, Graduate School of Engineering, Nagoya University, Nagoya, Japan
TAKURO YONEZAWA, Graduate School of Engineering, Nagoya University, Nagoya, Japan
NOBUO KAWAGUCHI, Institutes of Innovation for Future Society, Nagoya University, Nagoya, Japan

We propose **CLIFT** (**C**ross-City **Lif**estyle Pattern **T**ransfer for Human Mobility Prediction), a novel framework that enhances human mobility prediction by integrating general lifestyle patterns shared across cities with city-specific mobility patterns. Accurate human mobility prediction in urban environments is critical for transportation planning, marketing strategies, and disaster response. However, most existing deep learning approaches use only single-city data and exhibit significant performance degradation in small cities with limited training data. These limitations motivate methods that jointly leverage cross-city behavioral patterns and city-specific mobility characteristics. CLIFT addresses this challenge through dual complementary encoders: one captures general lifestyle patterns shared across cities, and the other captures city-specific mobility patterns; their outputs are integrated with a Transformer-based mobility predictor (LP-BERT). This architecture enables the model to jointly capture cross-city transferable behavioral patterns and city-specific mobility characteristics. We evaluated the effectiveness of CLIFT through experiments on the multi-city human mobility dataset LYMob-4Cities, comparing its performance with both single-city and multi-city deep learning-based methods. On average, CLIFT improved GEOBLEU and Top-1 accuracy by 11.1% and 10.6% over the single-city baseline, and by 5.0% and 7.9% over the multi-city baseline, respectively. Furthermore, CLIFT outperformed the top-ranked teams in the international competition, *Human Mobility Prediction Challenge 2024*, demonstrating superior predictive performance under the same dataset and task setting.

This work was partially supported by JST CREST (JPMJCR22M4), JST RISTEX (JPMJRS23K), NEDO SIP3 (JPJ012495), JSPS KAKENHI (22H03580), and JSPS KAKENHI (22K18422).
Authors' Contact Information: Haru Terashima (corresponding author), Graduate School of Engineering, Nagoya University, Nagoya, Aichi Prefecture, Japan; e-mail: haru@ucl.nuee.nagoya-u.ac.jp; Naoki Tamura, Graduate School of Engineering, Nagoya University, Nagoya, Aichi Prefecture, Japan; e-mail: tam@ucl.nuee.nagoya-u.ac.jp; Kazuyuki Shoji, Graduate School of Engineering, Nagoya University, Nagoya, Aichi Prefecture, Japan; e-mail: shoji@ucl.nuee.nagoya-u.ac.jp; Tahera Hossain, Graduate School of Engineering, Nagoya University, Nagoya, Aichi Prefecture, Japan; e-mail: tahera@ucl.nuee.nagoya-u.ac.jp; Shin Katayama, Graduate School of Engineering, Nagoya University, Nagoya, Aichi Prefecture, Japan; e-mail: shinsan@ucl.nuee.nagoya-u.ac.jp; Kenta Urano, Graduate School of Engineering, Nagoya University, Nagoya, Aichi Prefecture, Japan; e-mail: vrano@ucl.nuee.nagoya-u.ac.jp; Takuro Yonezawa, Graduate School of Engineering, Nagoya University, Nagoya, Aichi Prefecture, Japan; e-mail: takuro@nagoya-u.jp; Nobuo Kawaguchi, Institutes of Innovation for Future Society, Nagoya University, Nagoya, Aichi Prefecture, Japan; e-mail: kawaguti@nagoya-u.jp.

ACM 2374-0353/2026/05-ART10
https://doi.org/10.1145/3811037

CCS Concepts: • **Information systems → Spatial-temporal systems**; • **Human-centered computing → Ubiquitous and mobile computing**; • **Computing methodologies → Machine learning;**

Additional Key Words and Phrases: Human mobility, next location prediction, machine learning, transformer

ACM Reference Format:
Haru Terashima, Naoki Tamura, Kazuyuki Shoji, Tahera Hossain, Shin Katayama, Kenta Urano, Takuro Yonezawa, and Nobuo Kawaguchi. 2026. CLIFT: Cross-City Mobility-Derived Lifestyle Pattern Transfer for Improved Multi-City Next Location Prediction. *ACM Trans. Spatial Algorithms Syst.* 12, 2, Article 10 (May 2026), 24 pages. https://doi.org/10.1145/3811037

1 Introduction

The proliferation of GPS-enabled mobile devices has facilitated the large-scale collection of location data in urban environments. Human mobility histories derived from such location data provide valuable insights into behavioral patterns and have been widely utilized to address urban challenges [1–4]. In particular, large-scale and long-term mobility prediction has emerged as a crucial task across numerous applications, including the efficient allocation of shared mobility services, congestion mitigation during large-scale events, and the design of marketing strategies.

Recent deep learning approaches using RNNs [5], LSTMs [6], and Transformers [7] have significantly improved mobility prediction accuracy by capturing complex spatiotemporal dependencies in user behavior. However, these models require large, diverse datasets for training—typically thousands of users with months of historical data—making their performance highly dependent on city scale and user population. In small and medium-sized cities, limited user populations yield sparse mobility data, preventing models from learning stable spatiotemporal patterns. More fundamentally, training solely on single-city data restricts the diversity of behavioral patterns that a model can learn. These challenges motivate methods that transfer mobility-related knowledge from data-rich cities to data-scarce ones.

To address these challenges, we propose **CLIFT** (**C**ross-City **Lif**estyle Pattern **T**ransfer for Human Mobility Prediction), a novel framework that enhances human mobility prediction by integrating general lifestyle patterns shared across cities with city-specific mobility patterns. While mobility patterns vary across cities due to different urban structures and geographies, lifestyle patterns often exhibit partial consistency across cities. As typical examples of such lifestyle patterns, daily routines such as commuting between residential and office areas, shopping in commercial districts, and engaging in leisure activities during evenings or weekends are commonly shared, making them transferable across cities. CLIFT exploits this observation through a dual-encoder architecture that separately captures general lifestyle patterns (as transitions between urban function types of locations) and city-specific mobility patterns (as transitions between geographic coordinates of locations). In this formulation, lifestyle patterns are characterized by when people visit locations in each urban function category (e.g., offices, commercial districts, or dining areas) over the course of a day or week, and such temporal visitation patterns tend to be partially shared across cities among similar population groups. To explicitly represent such lifestyle patterns, it is necessary to abstract individual locations (defined as equal-sized grid cells) into functional categories that reflect their urban roles. Therefore, we estimated the urban function of each location using two approaches: an Area2Vec-based [8] approach, which classifies locations based on visit patterns, and a POI-based approach that uses point-of-interest (POI) data. Both approaches provide consistent functional representations across cities, enabling CLIFT to model transferable lifestyle patterns. Figure 1 illustrates the overall architecture of CLIFT. CLIFT consists of two encoders: the *general lifestyle pattern encoder*, which learns lifestyle features shared across cities, and the *city-specific mobility pattern*

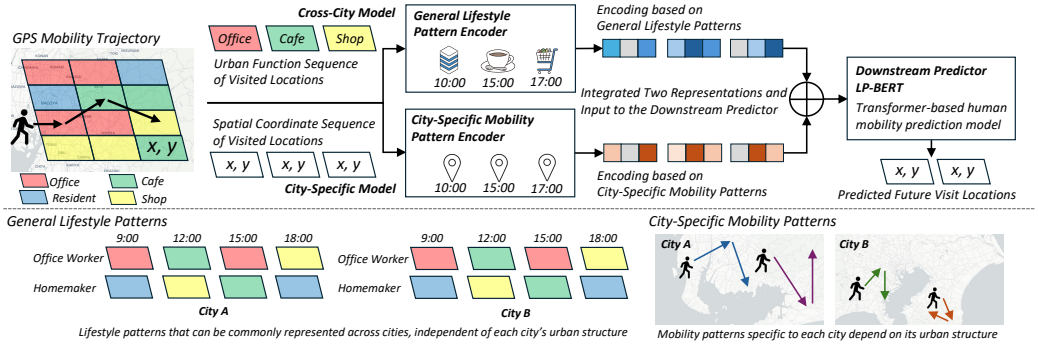

Fig. 1. The overall workflow of the proposed method, which combines cross-city behavioral features and city-specific mobility features through two complementary encoders for human mobility prediction.

encoder, which learns local mobility patterns—together with the downstream human mobility predictor, ***Location Prediction BERT (LP-BERT)*** [9]. The two encoders are pre-trained independently on multi-city and city-specific data, respectively, and their outputs are combined element-wise and fed into the downstream predictor, LP-BERT, a Transformer-based [10] human mobility prediction model. By integrating both encoders with LP-BERT, CLIFT simultaneously captures global lifestyle patterns and local mobility patterns, achieving higher prediction accuracy than single-city approaches.

We conducted experiments using an open human mobility dataset "LYMob-4Cities," which contains mobility data from multiple cities in Japan [11]. The dataset records 75 days of user mobility histories in each city at 30-minute intervals. Each movement in the mobility history is represented as a transition between locations defined by 500-meter square grid cells. We targeted four cities with different numbers of users. For each city, 20% of users were designated as prediction targets, and the task was to predict all visited locations at 30-minute intervals over the final 15 days of their mobility histories. In the experiments, we compared CLIFT with existing deep learning-based human mobility prediction methods designed for single-city and multi-city prediction. On average, CLIFT improved GEOBLEU by 0.032 points (11.1%) and Top-1 accuracy by 2.91 percentage points (10.6%) compared to the single-city baseline, and by 0.015 points (5.0%) and 2.21 percentage points (7.9%) compared to the multi-city baseline. Furthermore, CLIFT outperformed the top-ranked teams in the ***Human Mobility Prediction Challenge 2024 (HuMob Challenge 2024)*** [12], which was held at ACM SIGSPATIAL 2024. HuMob Challenge is one of the largest international competitions in urban human mobility prediction, and CLIFT achieved superior predictive performance on the same dataset and task setting.

The contributions of this study are summarized as follows:

— We introduce general lifestyle patterns as cross-city features for human mobility prediction, complementing city-specific mobility patterns in small-scale and medium-scale cities with limited mobility data.
— We propose **CLIFT**, a novel framework that integrates two complementary encoders—one capturing transferable general lifestyle patterns and the other capturing city-specific mobility patterns—with the downstream predictor LP-BERT for human mobility prediction.
— Through experiments on the open multi-city human mobility dataset "LYMob-4Cities", we demonstrate that CLIFT outperforms existing deep learning-based prediction models, improving GEOBLEU by 11.1% and Top-1 accuracy by 10.6% over the single-city baseline, and by 5.0% and 7.9% over the multi-city baseline.

2 Related Work

This section reviews related studies on human mobility in urban environments. We first discuss research on trajectory-based analysis and modeling. We then review prior work on human mobility prediction.

2.1 Trajectory-Based Human Mobility Studies

With the proliferation of mobile devices and location-based services, research on human mobility in urban environments has rapidly expanded [13–16]. Human mobility trajectories provide valuable insights into mobility patterns, lifestyle behaviors, and urban dynamics [17–19]. Recent studies utilizing human mobility data have focused on a wide range of tasks. These include trajectory completion and reconstruction [20, 21], which recover missing or sparsely sampled trajectories, and representation learning methods such as ZE-Mob and Area2Vec [8, 22], which learn low-dimensional location embeddings that capture semantic and spatial relationships. Other studies focus on user attribute estimation [23], trajectory similarity analysis [24], and trajectory-based location modeling [25–27]. Among these, human mobility prediction has become one of the most active research topics.

For trajectory-based studies, mobility data are generally categorized into (1) POI-based check-in data and (2) GPS-based mobility trajectories. POI-based datasets, such as Gowalla and Foursquare, collect voluntary check-ins at specific POIs [5, 25, 28], while GPS-based datasets consist of continuously sampled trajectories recorded by mobile devices [7, 21, 23]. Although POI-based datasets are easily obtained from location-based social networks, they primarily capture visits to commercial or leisure-oriented POIs, while largely missing residential and other non-commercial movements. As a result, these datasets provide only a partial view of urban mobility and offer limited support for modeling comprehensive lifestyle patterns. In contrast, GPS-based mobility traces capture user movements across the entire urban area, offering high spatial coverage and detailed representations of daily mobility behavior. These differences suggest that datasets with continuous and city-wide geographic coverage are better suited to fine-grained mobility analysis. Given this motivation, we adopted the open GPS-based dataset "LYMob-4Cities" [11], which provides city-wide mobility coverage while anonymizing data. The dataset has also been utilized in an international human mobility prediction challenge, providing a reliable benchmark for evaluating model performance.

2.2 Human Mobility Prediction

Human mobility prediction has evolved from early individual-based approaches to recent deep learning models. Early work treated mobility patterns as largely individual-specific, predicting future movements solely from a user's own historical trajectories [29, 30]. However, such individual-centric approaches are highly sensitive to data sparsity and often fail when only a limited mobility history is available. To overcome these limitations, deep learning-based models such as RNNs and LSTMs have been widely applied to human mobility prediction [5, 6]. These recurrent architectures learn from multiple users' mobility histories and can effectively capture short-range and mid-range temporal dependencies, leading to improved robustness against individual data sparsity. Beyond these baseline models, numerous RNN-based and LSTM-based variants have been proposed for human mobility prediction, including methods that emphasize long-term and short-term periodicity, incorporate group-level learning, and are tailored to sparse trajectories [31–35].

More recently, Transformer-based architectures [10] have advanced human mobility prediction by effectively modeling long-range spatial-temporal dependencies and capturing periodic mobility patterns through multi-head attention. Among attention-based approaches, the STAN

Table 1. Terminology and Notation

Terminology	*Notation*	*Description*
Mobility History	$M_u = [m_1, \dots, m_T]$	Sequence of movements where each $m_i = (l_i, f_i, d_i, t_i, w_i, \Delta t_i)$.
Location	$L_u = [l_1, \dots, l_T]$	Grid cell dividing the city into equal units ($500m \times 500m$).
Urban Function	$F_u = [f_1, \dots, f_T]$	Estimated urban function associated with each location.
Date	$D_u = [d_1, \dots, d_T]$	Sequential day index starting from day 0.
Time	$T_u = [t_1, \dots, t_T]$	Discrete time slots dividing a day into equal intervals (30 minutes).
Weekday	$W_u = [w_1, \dots, w_T]$	Categorical variable indicating one of the seven days of the week.
Timediff	$\Delta T_u = [\Delta t_1, \dots, \Delta t_T]$	Time interval between consecutive movements.

model [28] predicts future movements by leveraging sequence-level attention over historical locations. Building on this direction, a Transformer decoder-based architecture [7] achieves higher accuracy by capturing more complex temporal dependencies, outperforming earlier recurrent and single-attention models. Our previous work, LP-BERT [9], showed that masked training with a bidirectional Transformer encoder effectively captures fine-grained spatiotemporal patterns in mobility trajectories. However, most existing studies train prediction models using data from only the target city, limiting their applicability in cities with limited mobility records. Although several approaches leverage mobility patterns shared across multiple cities [36, 37], they often lack mechanisms to capture city-specific characteristics, as they do not perform any explicit adaptation or tuning tailored to each city, leading to reduced performance in diverse urban environments. These limitations highlight the need for methods that jointly model both transferable cross-city behaviors and city-specific mobility patterns.

3 Preliminaries

This section introduces the terminology and notation adopted in this study, followed by the problem formulation for human mobility prediction. We then describe the method for estimating the urban function of each location, which serves as the basis for representing general lifestyle patterns shared across cities.

3.1 Mobility Data Representation

The terminology and notation used in this study are summarized in Table 1. Let M_u denote the mobility history of user u, represented as an ordered sequence of visited locations together with their urban functions and temporal attributes:

$$M_u = [(l_1, f_1, d_1, t_1, w_1, \Delta t_1), (l_2, f_2, d_2, t_2, w_2, \Delta t_2), \dots, (l_T, f_T, d_T, t_T, w_T, \Delta t_T)], \tag{1}$$

where l_i, f_i, d_i, t_i, w_i, and Δt_i denote the location, its corresponding urban function, date, time, weekday, and time difference of the ith visit, respectively. We partition each city into equally sized grid cells ($500m \times 500m$), where each cell represents a distinct spatial *location* (l). Throughout this paper, location refers to grid cells rather than individual POIs. Urban functions capture the semantic characteristics of locations (e.g., residential, commercial, and office) and are estimated using Area2Vec or POI-based methods described in Section 3.3. In addition to these spatial and functional attributes, we incorporate multiple temporal features—*date* (d), *time* (t), *weekday* (w), and *timediff* (Δt)—for each movement step. Here, *timediff* represents the time interval between consecutive movements, computed from the absolute timestamp obtained by combining the *date* and *time*, so that it reflects the true elapsed time between visits ($\Delta t_1 = 0$, $\Delta t_i = (d_i \cdot 48 + t_i) - (d_{i-1} \cdot 48 + t_{i-1})$ $(i > 1)$). These temporal attributes enable the model to capture both short-term and long-term periodicities in user mobility behaviors.

3.2 Problem Formulation

Given a user's mobility history M_u defined in Equation (1), where each element represents the visited location, its urban function, and temporal context, the goal is to predict the sequence of future visit locations as follows:

$$\hat{L}_u^{T+1:T+K} = [\hat{l}_{T+1}, \hat{l}_{T+2}, \ldots, \hat{l}_{T+K}], \tag{2}$$

where $\hat{L}_u^{T+1:T+K}$ represents the predicted location sequence for the future interval from $T+1$ to $T+K$. In this study, K corresponds to up to 15 days with a 30-minute resolution, i.e., at most 720 timesteps, although in practice only the timesteps at which movements are actually observed in the dataset are included in the prediction window. Note that temporal attributes for the prediction window—*date* (d), *time* (t), *weekday* (w), and *timediff* (Δt) for each step from $T+1$ to $T+K$—are assumed to be known and can be used as inputs during prediction. This formulation captures three types of dependencies: spatial transitions between locations, functional transitions across urban function types, and temporal dynamics including daily and weekly cycles, providing a unified basis for modeling human mobility patterns.

3.3 Methods for Estimating Urban Functions of Locations

To represent general lifestyle patterns shared across cities, CLIFT requires each location to be associated with an urban function (e.g., residential, office, and commercial). Because the dataset used in this study anonymizes city names, it is difficult to use external geographic resources such as official land-use maps. Therefore, we estimate the urban function of each location using only the mobility trajectories and POI information included in the dataset. Various methods have been proposed for modeling locations using mobility data [8, 22, 26, 27, 38]. For cross-city modeling, it is essential to employ approaches that yield consistent location representations across different cities. To achieve this, we adopt two approaches for estimating the urban function of each location: the *Area2Vec*-based approach [8], which models locations based on user visit patterns extracted from mobility trajectories, and the *POI*-based approach, which represents each location using the POI distribution provided in the dataset.

Area2Vec-based approach. Area2Vec estimates the functional characteristics of locations by capturing users' visit behaviors. Locations frequently visited during weekday daytime tend to be characterized as office areas, whereas those with higher nighttime visits are often associated with residential districts. In this approach, each location is first encoded into a vector representation based on behavioral features such as visit-time statistics and weekday visit distributions (e.g., hourly visit frequencies for each day of the week). The resulting embeddings are then clustered into a predefined number of groups using the k-means++ algorithm [39], with each cluster treated as a distinct urban function. A key advantage of this method is that it derives functional representations solely from mobility trajectories, without requiring any external geographic information.

POI-based approach. In the POI-based approach, the urban function of each location is inferred from the POI category distribution associated with that location. For each location, we use the POI category count vector provided in the dataset, normalize it, apply PCA for dimensionality reduction, and then cluster the resulting representations into a predefined number of groups using the k-means++ algorithm. Each cluster represents an urban function derived from the POI distribution associated with each location.

In summary, locations are modeled using both the Area2Vec-based and POI-based approaches, and the resulting clusters are treated as the urban functions of the corresponding locations.

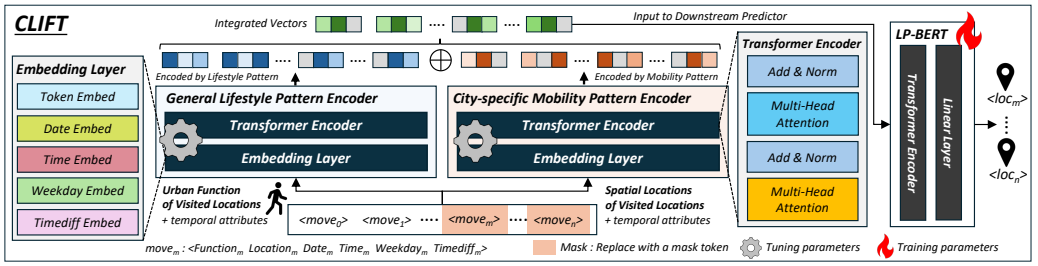

Fig. 2. Overview of the proposed **CLIFT** framework. CLIFT consists of two Transformer-based encoders: (1) the *general lifestyle pattern encoder* that learns cross-city transferable lifestyle patterns based on the urban functions of visited locations, and (2) the *city-specific mobility pattern encoder* that learns mobility patterns unique to each city based on visited locations. The outputs of these encoders are fused and passed to the downstream predictor, *LP-BERT*, which performs multi-step mobility prediction.

4 Methodology

In this section, we describe the overall architecture of the **CLIFT** framework, which consists of three main components: (1) the general lifestyle pattern encoder that learns transferable behavioral patterns across cities, (2) the city-specific mobility pattern encoder that captures local spatial characteristics such as urban structures and movement tendencies, and (3) LP-BERT, a downstream predictor that fuses both representations and predicts future human mobility. Both encoders are pre-trained independently on their respective training data and subsequently fine-tuned during LP-BERT training. Figure 2 illustrates the overall architecture and process flow of the CLIFT framework.

4.1 General Lifestyle Pattern Encoder

The general lifestyle pattern encoder is designed to learn transferable lifestyle patterns across cities based on *urban functions* of visited locations. These patterns, such as commuting to office areas, shopping in commercial districts, and leisure activities during evenings, exhibit consistent temporal and functional characteristics across different urban contexts. As shown in Figure 3, the encoder consists of an *Embedding layer* and multiple *Transformer Encoder layers*. We adopt the Transformer architecture because its self-attention mechanism is well suited to capturing complex, long-range temporal dependencies in time-series human mobility data. For each user u, the input to the general lifestyle encoder is the entire mobility history, represented as:

$$M_u^{life} = [(f_i, d_i, t_i, w_i, \Delta t_i) \mid i = 1, \ldots, T], \tag{3}$$

where f_i denotes the *urban function* assigned to the ith visited location, and d_i, t_i, w_i, and Δt_i denote the date, time, weekday, and time difference between consecutive movements (timediff), respectively. Here, T denotes the number of movements observed in the user's mobility history. During training, the urban functions for a subset of movements in the sequence are randomly masked, and the model learns to predict the masked values from the surrounding context, following the masked modeling strategy popularized by BERT [40]. Since the encoder is designed to learn general contextual representations, the masked positions are sampled independently rather than as contiguous spans. Each input token $(f_i, d_i, t_i, w_i, \Delta t_i)$ is mapped to dense embeddings and summed to form a unified representation:

$$\mathbf{m}_i^{\text{life}} = \mathbf{e}_{f_i} + \mathbf{e}_{d_i} + \mathbf{e}_{t_i} + \mathbf{e}_{w_i} + \mathbf{e}_{\Delta t_i}, \tag{4}$$

where $\mathbf{e}_{f_i}$, $\mathbf{e}_{d_i}$, $\mathbf{e}_{t_i}$, $\mathbf{e}_{w_i}$, $\mathbf{e}_{\Delta t_i}$ are learned embeddings for urban function, date, time, weekday, and timediff, respectively. The resulting sequence $\mathbf{M}^{\text{life}} = [\mathbf{m}_1^{\text{life}}, \ldots, \mathbf{m}_T^{\text{life}}]$ is processed by a stack of L

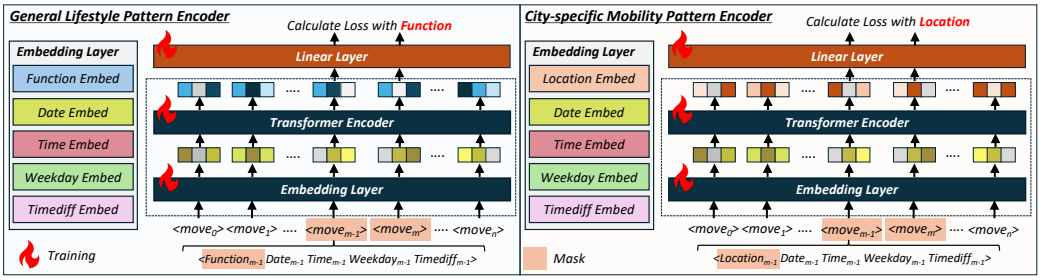

Fig. 3. Detailed architecture of the dual encoders used in CLIFT. Each encoder comprises an *Embedding layer* and multiple *Transformer Encoder layers*, with a linear layer module applied exclusively during encoder pre-training.

Transformer encoder layers to obtain contextualized representations:

$$\mathbf{H}^{\text{life}} = \text{TransformerEnc}(\mathbf{M}^{\text{life}}) = [\mathbf{h}_1^{\text{life}}, \mathbf{h}_2^{\text{life}}, \dots, \mathbf{h}_T^{\text{life}}]. \tag{5}$$

Each output vector $\mathbf{h}_i^{\text{life}}$ captures cross-city temporal and functional dependencies across the entire sequence, representing the contextualized lifestyle pattern embedding for the ith timestep. A linear output layer is attached only during the pre-training phase of the encoder to predict the masked urban functions. Let $\mathcal{M}$ denote the set of masked positions. The model is optimized using the cross-entropy loss computed only over these positions:

$$\mathcal{L}_{\text{life}} = -\sum_{i \in \mathcal{M}} \log P(f_i \mid \mathbf{H}^{\text{life}}), \tag{6}$$

where $P(f_i \mid \mathbf{H}^{\text{life}})$ is obtained by applying a softmax layer over all possible urban function types. The general lifestyle pattern encoder is pre-trained on multi-city data to learn generalizable lifestyle patterns.

4.2 City-specific Mobility Pattern Encoder

The city-specific mobility pattern encoder captures mobility patterns unique to each city, reflecting its local urban structure and movement characteristics based on the visited *locations*. Similar to the general lifestyle pattern encoder described in Section 4.1, it consists of an *Embedding layer* and multiple *Transformer Encoder layers*. Unlike the general lifestyle pattern encoder trained on multi-city mobility data, this encoder is trained separately for each city to capture local mobility characteristics. For each user u, the input to the city-specific mobility pattern encoder is the entire mobility history, represented as:

$$M_u^{mob} = [(l_i, d_i, t_i, w_i, \Delta t_i) \mid i = 1, \dots, T], \tag{7}$$

where l_i represents the visited location, while d_i, t_i, w_i, and Δt_i denote the associated temporal attributes. Following the same training strategy as the general lifestyle pattern encoder, the locations for a subset of movements in the sequence are randomly masked, and the model learns to predict the masked values from the surrounding context. Each input token $(l_i, d_i, t_i, w_i, \Delta t_i)$ is embedded as:

$$\mathbf{m}_i^{\text{mob}} = \mathbf{e}_{l_i} + \mathbf{e}_{d_i} + \mathbf{e}_{t_i} + \mathbf{e}_{w_i} + \mathbf{e}_{\Delta t_i}, \tag{8}$$

where $\mathbf{e}_{l_i}$, $\mathbf{e}_{d_i}$, $\mathbf{e}_{t_i}$, $\mathbf{e}_{w_i}$, $\mathbf{e}_{\Delta t_i}$ are learned embeddings for location, date, time, weekday, and timediff, respectively. The resulting sequence $\mathbf{M}^{\text{mob}} = [\mathbf{m}_1^{\text{mob}}, \dots, \mathbf{m}_T^{\text{mob}}]$ is then processed by the Transformer encoder layers to obtain contextualized mobility representations:

$$\mathbf{H}^{\text{mob}} = \text{TransformerEnc}(\mathbf{M}^{\text{mob}}) = [\mathbf{h}_1^{\text{mob}}, \mathbf{h}_2^{\text{mob}}, \dots, \mathbf{h}_T^{\text{mob}}]. \tag{9}$$

Each output vector $\mathbf{h}_i^{\text{mob}}$ captures city-specific spatial and temporal dependencies, serving as the contextualized representation of the ith movement. A linear output layer is attached only during the pre-training phase of the encoder to predict the masked locations. As in the general lifestyle pattern encoder, let $\mathcal{M}$ denote the set of masked positions. The model is optimized using the cross-entropy loss computed only over these positions:

$$\mathcal{L}_{\text{mob}} = -\sum_{i \in \mathcal{M}} \log P(l_i \mid \mathbf{H}^{\text{mob}}), \tag{10}$$

where $P(l_i \mid \mathbf{H}^{\text{mob}})$ is computed via a softmax over the set of locations in the target city. The city-specific mobility pattern encoder is pre-trained on single-city data to learn local mobility patterns.

4.3 Downstream Prediction with LP-BERT

LP-BERT [9] serves as the downstream predictor in CLIFT. Originally proposed as a single-city next-location prediction model, LP-BERT consists of an *Embedding layer*, multiple *Transformer Encoder layers*, and a *Linear output layer*, with all components trained separately for each city. As illustrated in Figure 2, we extend LP-BERT to the multi-city setting by replacing its *Embedding layer* with the two pre-trained encoders—the *general lifestyle pattern encoder* and the *city-specific mobility pattern encoder*—whose fused outputs are fed into the shared Transformer encoder layers for downstream prediction.

Given a user's mobility history $M_u^{CLIFT} = [(l_1, f_1, d_1, t_1, w_1, \Delta t_1), \ldots, (l_T, f_T, d_T, t_T, w_T, \Delta t_T)]$, the two encoders process temporally aligned sequences derived from M_u^{CLIFT}, receiving different input attributes. The *general lifestyle pattern encoder* takes urban functions $(f_1, \ldots, f_T)$ with temporal features $(d_i, t_i, w_i, \Delta t_i)$, whereas the *city-specific mobility pattern encoder* takes locations $(l_1, \ldots, l_T)$ with the same temporal features. Each encoder outputs a temporally aligned representation sequence:

$$\mathbf{H}^{\text{life}} = \{\mathbf{h}_1^{\text{life}}, \ldots, \mathbf{h}_T^{\text{life}}\}, \quad \mathbf{H}^{\text{mob}} = \{\mathbf{h}_1^{\text{mob}}, \ldots, \mathbf{h}_T^{\text{mob}}\}, \tag{11}$$

where $\mathbf{H}^{\text{life}}$ and $\mathbf{H}^{\text{mob}}$ denote the encoded representation sequences produced by the general lifestyle pattern encoder and the city-specific mobility pattern encoder, respectively. These two encoded sequences are fused in an element-wise manner, where the corresponding representations at each timestep are added together to form the integrated representation sequence:

$$\mathbf{Z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_T\}, \quad \text{where } \mathbf{z}_i = \mathbf{h}_i^{\text{life}} + \mathbf{h}_i^{\text{mob}}, \tag{12}$$

where $\mathbf{Z}$ represents the fused sequence obtained by summing the outputs of the two encoders. The fused sequence $\mathbf{Z}$ is processed by the Transformer Encoder layers of LP-BERT to capture contextual dependencies across all timesteps:

$$\mathbf{H}^{\text{CLIFT}} = \text{TransformerEnc}(\mathbf{Z}) = [\mathbf{h}_1^{\text{CLIFT}}, \mathbf{h}_2^{\text{CLIFT}}, \ldots, \mathbf{h}_T^{\text{CLIFT}}]. \tag{13}$$

Each output vector $\mathbf{h}_i^{\text{CLIFT}}$ encodes contextual information from both cross-city and city-specific patterns. During training, we randomly select consecutive movements in the sequence and mask both their locations and urban functions in the inputs; LP-BERT then learns to predict the masked locations from the surrounding context in $\mathbf{H}^{\text{CLIFT}}$. Unlike the original BERT model in natural language processing, which masks random tokens, LP-BERT masks consecutive movements to capture the temporal continuity of human mobility better. The model is optimized using the cross-entropy loss computed only over the masked positions ($\mathcal{M}$):

$$\mathcal{L}_{CLIFT} = -\sum_{i \in \mathcal{M}} \log P(l_i \mid \mathbf{H}^{\text{CLIFT}}), \tag{14}$$

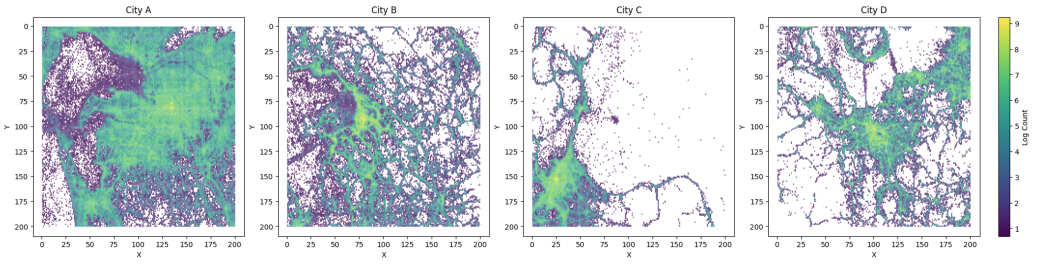

Fig. 4. Spatial distribution of data volume per location in the dataset.

where $P(l_i \mid \mathbf{H}^{\mathrm{CLIFT}})$ is computed via a softmax over the set of locations in the target city. Although both locations and urban functions are masked as inputs, the loss is computed only on the masked locations. Gradients are propagated through the entire architecture, and the parameters of LP-BERT and two pre-trained encoders are jointly optimized for each target city. For each target city, LP-BERT is trained on the mobility histories of users in that city. This enables CLIFT to preserve cross-city generalization from pre-training while adapting to city-specific characteristics.

During inference, the model receives a user's complete mobility history up to time T and the temporal attributes of future steps $(d_{T+1:T+K}, t_{T+1:T+K}, w_{T+1:T+K}, \Delta t_{T+1:T+K})$. For the prediction window, the future locations and their corresponding urban functions are masked and fed as inputs to both encoders. We replace them with the same dedicated mask tokens as during training, and the model then predicts all unknown locations simultaneously in a single forward pass from the resulting contextual representations:

$$\hat{l}_i = \arg\max P(l_i \mid \mathbf{H}^{\mathrm{CLIFT}}), \quad i \in [T+1, T+K], \tag{15}$$

where $\hat{l}_i$ denotes the location that attains the maximum predicted probability of being visited at timestep i.

5 Experiments

This section presents the experimental design. We first describe the dataset and task settings used in the experiments, followed by the procedures for estimating the urban function of locations and training the models. Finally, we conduct comparative experiments against baseline methods, perform ablation studies to analyze the contribution of each component in CLIFT, and investigate fusion methods for the two encoder outputs.

5.1 Dataset and Task Settings

We conducted experiments on LYMob-4Cities, an open multi-city human mobility dataset constructed as a four-city subset of the large-scale GPS-based YJMob100K dataset [11]. This dataset has been adopted as the official benchmark in the HuMob Challenge 2024. It contains user mobility histories from four Japanese cities with varying scales. Table 2 summarizes the statistics of the dataset used in the experiments, and Figure 4 visualizes the spatial distribution of data volume across locations. Each mobility history spans 75 consecutive days and is recorded at 30-minute intervals, with each record representing a movement between 500m × 500m grid cells. Time intervals without observed movements are treated as missing data, and the dataset consists only of actually observed movements. The names of the cities and the data collection period are not disclosed in order to protect user privacy. We designated 20% of users per city as prediction targets and an additional 10% for validation. The task is to predict the visited locations of each target user at 30-minute

Table 2. Dataset Summary

	City A	*City B*	*City C*	*City D*
Number of Users	100,000	20,000	15,000	3,000
Number of Target Users	20,000	4,000	3,000	600
Number of Records	111,535,175	20,253,615	14,425,227	4,352,478
Number of Locations	40,000	40,000	40,000	40,000

intervals during the 15 days following day 60 of their mobility histories, where only timesteps with actually observed movements are included. For training all models (the two encoders and LP-BERT), we used the mobility histories of non-target users across all 75 days, along with the first 60 days of the target users' histories. Since the dataset provides only location, date, and time for each movement, we derived additional temporal features: weekday (inferred from daily data volume patterns) and time difference between consecutive movements (timediff). Concretely, we identified a clear 7-day periodicity in the daily movement volume, treated days with markedly fewer movements as weekends, and aligned the local minima of this 7-day pattern with these days to reconstruct weekday labels. Data volume analysis is provided in Appendix A.

To evaluate prediction accuracy, we employed two metrics: GEOBLEU and Top-k accuracy.

— **GEOBLEU [41]**
GEOBLEU is an evaluation metric for mobility prediction inspired by BLEU [42] in natural language processing. It measures the local sequence-level accuracy of mobility trajectories based on N-grams. Higher values indicate better performance: a score of 1 represents a perfect match. In this study, we set N to the default value of 3.

— **Top-k Accuracy**
Top-k accuracy measures the probability that the ground-truth location is among the k predicted locations. We report results for $k = 1$ and $k = 5$.

5.2 Estimating the Urban Function of Each Location

We estimated the urban function of each location using two approaches: the Area2Vec-based approach [8] and the POI-based approach, as described in Section 3.3. The number of clusters was determined through preliminary experiments comparing multiple settings (e.g., 16, 36, and 64 clusters) presented in Appendix B; we adopted 36 clusters. Figure 5 visualizes the resulting functional clusters obtained from both approaches.

Area2Vec-based approach. We trained Area2Vec using mobility histories from all four cities (excluding the data from the prediction target period days 61–75 for target users) to learn latent representations of locations based on visit patterns over days of the week and times of day. The model was trained for 100 epochs to obtain stable embeddings, learning for each location an 8-dimensional latent representation from its visit-frequency patterns over 7 weekdays and 24 hourly time slots. The dimensionality was set to 8, following the original Area2Vec study, as this was found to be sufficient to represent the visit patterns. After training, we jointly clustered the location vectors from all four cities into 36 groups using the k-means++ algorithm, yielding a shared set of urban function categories across cities. Locations with insufficient visit data (10 or fewer visits) were assigned to an additional cluster, resulting in 37 clusters in total.

POI-based approach. For the POI-based estimation, we used the POI data provided in the same dataset [11]. Each location was represented as an 85-dimensional count vector corresponding to POI categories (e.g., restaurants, cafes, and retail stores). The vectors were L1-normalized and reduced to eight dimensions using PCA, with the dimensionality chosen to match that of the Area2Vec

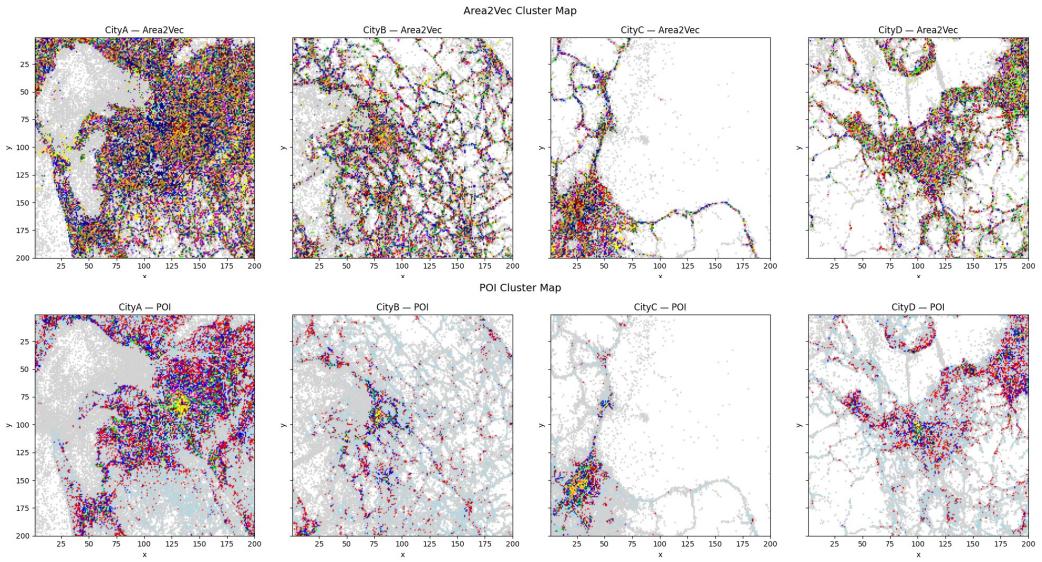

Fig. 5. Comparison of estimated urban functions of locations between the Area2Vec-based and POI-based approaches. For each city, locations are classified into 37 clusters, each represented by a distinct color (the grey cluster represents locations where mobility data or POI data are sparse). Each cluster is treated as the urban function assigned to the corresponding locations.

embeddings. We then jointly clustered the locations from all four cities into 36 groups using the k-means++ algorithm. Locations without any POIs were assigned to an additional cluster, also yielding 37 clusters.

5.3 Training CLIFT (Pre-trained Encoders and Downstream Predictor)

This subsection details the training procedures for the two pre-trained encoders and the downstream predictor, LP-BERT. The hyperparameters used in all training processes are summarized in Table 3. All models were implemented in PyTorch and experiments were conducted on the NVIDIA H100 GPU, with 96 GB of memory.

5.3.1 Pre-training the Encoders. The general lifestyle pattern encoder was trained as a shared model on mobility data from all four cities, whereas the city-specific mobility pattern encoder was trained independently for each city on its own mobility data. For both encoders, we used the mobility histories of non-target users across all 75 days and the first 60 days of the target users' histories. During pre-training of these encoders, 20% of the movements in each mini-batch were randomly selected and masked, and the masked positions were re-sampled at every training iteration. This follows the masked modeling strategy, where the model learns to reconstruct masked tokens from their surrounding context. The encoders were optimized using the masked prediction losses described in Sections 4.1 and 4.2, and each encoder was trained for 300 epochs using the AdamW optimizer (learning rate: 1e-4).

5.3.2 Training LP-BERT. LP-BERT takes the outputs of the two pre-trained encoders as input and is trained on users' mobility histories in each city. As in pre-training the encoders, we used the mobility histories of non-target users across all 75 days and the first 60 days of the target users'

Table 3. Hyperparameters of CLIFT

	Layers	*Attention Heads*	*Learning Rate*	*Batch Size*	*Embed Size*
General Lifestyle Pattern Encoder	4	8	1e-4	64	128
City-Specific Mobility Pattern Encoder (City A, B, C)	4	8	1e-4	16	128
City-Specific Mobility Pattern Encoder (City D)	3	8	1e-4	8	128
Downstream LP-BERT (City A, B, C)	2	8	5e-5	16	128
Downstream LP-BERT (City D)	2	8	5e-5	8	128

histories for training. For each user, a continuous 15-day window—set to match the prediction period—was masked as the prediction targets, with the starting position randomly re-sampled at every training iteration. For non-target users, this 15-day window could span any part of the 75-day history, whereas for target users it was sampled within the first 60 days. All model parameters, including those of both encoders and LP-BERT, were jointly optimized for each target city, as described in Section 4.3. LP-BERT was trained for 100 epochs using the AdamW optimizer (learning rate: 5e-5).

5.4 Comparison with both Single-City and Multi-City Baseline Models

We compared CLIFT against six baselines spanning recurrent models (LSTM, ST-LSTM, and STGN) and Transformer architectures (Transformer, LP-BERT, Cross-city BERT). For the single-city baselines (LSTM, ST-LSTM, STGN, Transformer, and LP-BERT), we trained an independent model for each city using only the mobility data from that city, while Cross-city BERT was trained once as a multi-city model on the combined data from all four cities, consistent with its original formulation. The baselines were also provided with the temporal features for both the historical sequence and the prediction window. All models were implemented in PyTorch, following their original descriptions, with hyperparameters tuned for the best Top-1 accuracy (details in Appendix C).

— **LSTM [43]**
Model that processes visited locations sequentially with LSTM cells and predicts the next location at each step.
— **ST-LSTM [6]**
LSTM-based model that integrates spatial and temporal intervals into the gating mechanism to capture the effects of travel distance and time gaps.
— **STGN [44]**
LSTM-based model that integrates time and distance information to model dependencies between consecutive movements within a trajectory.
— **Transformer [7]**
Transformer Decoder-based model that leverages attention mechanisms to learn long-range spatial and temporal dependencies in mobility trajectories.
— **LP-BERT [9]**
Transformer Encoder-based model that learns spatiotemporal mobility patterns through masked learning inspired by the BERT architecture.
— **Cross-city BERT [36]**
Transformer Encoder-based model that jointly learns mobility patterns across multiple cities to improve adaptation in cities with limited data.

Table 4 summarizes the prediction results of CLIFT and the baseline models. Since CLIFT (Area2Vec) and CLIFT (POI) exhibit comparable performance, we report the absolute improvements using CLIFT (Area2Vec) as a representative configuration. Based on the average performance across the four cities, CLIFT (Area2Vec) improves GEOBLEU from 0.2905 to 0.3229 and Top-1

Table 4. Prediction Results for Each Model Across Cities [GEOBLEU (GEO)↑, Top-1↑, and Top-5↑]

Model	*City A (100,000)*			*City B (20,000)*			*City C (15,000)*			*City D (3,000)*		
	GEO	Top-1	Top-5	GEO	Top-1	Top-5	GEO	Top-1	Top-5	GEO	Top-1	Top-5
LSTM [43]	0.1825	18.26	36.79	0.1632	15.73	33.39	0.1684	16.58	32.44	0.0777	6.18	15.77
ST-LSTM [6]	0.2007	20.04	39.14	0.1757	16.88	35.04	0.1771	17.14	33.75	0.0803	6.65	14.94
STGN [44]	0.2029	20.31	40.41	0.1905	18.95	37.92	0.1945	19.55	37.29	0.1777	15.37	31.74
Transformer [7]	0.2561	22.94	47.57	0.2358	20.80	43.49	0.2264	20.81	41.65	0.2462	18.90	42.58
LP-BERT [9]	0.3264	29.96	56.47	0.2919	27.64	54.06	0.2894	27.84	52.27	0.2543	24.00	50.04
Cross-city BERT [36]	0.3232	29.55	56.12	0.3090	28.59	55.67	0.3056	28.87	54.06	0.2925	25.20	53.29
CLIFT (Area2Vec)	**0.3460**	**32.06**	**58.53**	**0.3281**	**30.99**	**57.49**	**0.3228**	**31.05**	**55.62**	**0.2946**	**26.96**	**53.55**
CLIFT (POI)	**0.3459**	**32.07**	**58.49**	**0.3281**	**30.80**	**57.41**	**0.3223**	**30.97**	**55.65**	**0.2943**	**27.15**	**53.61**

The bold entries indicate the results of the proposed method (CLIFT).

accuracy from 27.36% to 30.27%, corresponding to absolute gains of 0.032 and 2.91 percentage points, respectively, over the single-city baseline (LP-BERT). Compared to the multi-city baseline (Cross-city BERT), CLIFT (Area2Vec) improves GEOBLEU from 0.3076 to 0.3229 and Top-1 accuracy from 28.05% to 30.27%, corresponding to absolute gains of 0.015 and 2.21 percentage points, respectively. These results confirm the effectiveness of CLIFT in multi-city human mobility prediction.

The comparison between LP-BERT and Cross-city BERT highlights the effect of city scale and data availability. In smaller cities with fewer users (Cities B, C, and D), Cross-city BERT achieves higher accuracy by transferring patterns from data-rich cities to these smaller cities. In the largest City A, LP-BERT performs better, indicating that when sufficient data are available, single-city models can fully learn local mobility patterns, whereas multi-city models may suffer from the additional noise introduced by other cities. CLIFT addresses this trade-off by jointly leveraging cross-city and city-specific patterns, enabling the model to consider them simultaneously. However, for City D—the smallest city with only 3,000 users—the improvement over the multi-city baseline is more limited than in other cities. This may be attributed to the extremely limited number of users, which prevents the city-specific mobility pattern encoder from sufficiently learning local mobility patterns, while also making it harder for the general lifestyle pattern encoder to adapt to the city's characteristics.

We also examined the differences between the two urban function estimation methods. As shown in Table 4, the difference in prediction accuracy between the Area2Vec-based and POI-based approaches is minimal. Figure 5 further shows that the Area2Vec-based approach provides broader spatial coverage, whereas the POI-based approach leaves some regions unclassified (grey indicates locations with sparse mobility or POI data). The POI-based method performs well where POI data are abundant—typically in city centers—but degrades in peripheral areas with limited POI data. In contrast, the Area2Vec-based method is more robust in POI-sparse regions or when spatial cells are small, although it may become noisy when cell sizes are too large. Overall, these results suggest that the choice of estimation method should be guided by the availability and characteristics of both POI and mobility data. Given these findings and the negligible difference in accuracy, we adopt the Area2Vec-based estimation in subsequent experiments. We then conduct an ablation study to assess the contribution of each component in CLIFT.

5.5 Ablation Study

To evaluate the contribution of each component in CLIFT, we conducted an ablation study using model variants with specific modules removed. Throughout this subsection, we used the Area2Vec-based urban function representation. We compared the following three configurations:

— **LP-BERT w/ Urban Function of Locations (UF)**
Baseline model that directly embeds the urban function of locations and feeds them into LP-BERT without using both of the proposed pre-trained encoders. Concretely, the urban

Table 5. Prediction Results for Ablation Models Across Cities [GEOBLEU (GEO)↑, Top-1↑, and Top-5↑]

Model	*City A (100,000)*			*City B (20,000)*			*City C (15,000)*			*City D (3,000)*		
	GEO	Top-1	Top-5	GEO	Top-1	Top-5	GEO	Top-1	Top-5	GEO	Top-1	Top-5
LP-BERT w/ UF	0.3254	29.93	56.42	0.2938	27.73	54.32	0.2894	27.90	52.50	0.2633	24.41	50.98
CLIFT w/o LE	0.3423	31.66	58.13	0.3223	30.23	56.83	0.3163	30.31	55.02	0.2816	25.87	52.29
CLIFT w/o ME	0.3411	31.63	57.79	0.3218	30.52	56.70	0.3187	30.78	55.10	0.2880	26.72	52.84
CLIFT (Area2Vec)	**0.3460**	**32.06**	**58.53**	**0.3281**	**30.99**	**57.49**	**0.3228**	**31.05**	**55.62**	**0.2946**	**26.96**	**53.55**

The bold entries indicate the results of the proposed method (CLIFT).

function embedding is added element-wise to the original LP-BERT input embedding before being passed to the Transformer of LP-BERT.

— **CLIFT w/o General Lifestyle Pattern Encoder (LE)**
Variant that uses only the city-specific mobility pattern encoder, while directly embedding the urban function of locations as additional inputs to LP-BERT. Concretely, the urban function embedding is added element-wise to the output of the city-specific mobility pattern encoder before being passed to LP-BERT.

— **CLIFT w/o City-specific Mobility Pattern Encoder (ME)**
Variant that uses only the general lifestyle pattern encoder, while directly embedding locations as additional inputs to LP-BERT. Concretely, the location embedding is added element-wise to the output of the general lifestyle pattern encoder before being passed to LP-BERT.

The results are summarized in Table 5. CLIFT (complete model) achieved the highest overall accuracy among all variants, demonstrating the effectiveness of using two complementary encoders to model both general lifestyle patterns and city-specific mobility patterns. Simply adding urban function embeddings to the original LP-BERT inputs (LP-BERT w/ UF) yielded the lowest accuracy, indicating that providing urban function information as an additional input is insufficient for capturing transferable lifestyle patterns. Removing the general lifestyle pattern encoder (CLIFT w/o LE) prevented the model from capturing cross-city commonalities, leading to a noticeable decrease in accuracy, particularly for data-scarce cities (Cities B–D). Similarly, removing the city-specific mobility pattern encoder (CLIFT w/o ME) limited the model's ability to learn local mobility characteristics, which also resulted in reduced performance. Overall, the ablation study shows that CLIFT effectively integrates transferable cross-city knowledge with city-specific mobility characteristics, validating the design choice of using two complementary encoders for robust mobility prediction.

5.6 Fusion Methods for the Two Encoder Outputs

In CLIFT, the outputs of the two encoders—the general lifestyle pattern encoder and the city-specific mobility pattern encoder—are added as in Equation (12) and fed into the downstream predictor LP-BERT. However, other fusion strategies for combining the encoder outputs are also conceivable. We therefore compare this addition-based fusion with a concatenation-based fusion:

$$\mathbf{Z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_T\}, \quad \mathbf{z}_i = \mathrm{concat}\big(\mathbf{h}_i^{\mathrm{life}}, \mathbf{h}_i^{\mathrm{mob}}\big), \tag{16}$$

where $\mathrm{concat}(\cdot,\cdot)$ denotes vector concatenation. As summarized in Table 6, under the addition-based fusion strategy the dimensionality of the vector sequence fed into LP-BERT is identical to the output dimensionality of each encoder, whereas under the concatenation-based strategy the dimensionality of the representations received by LP-BERT is doubled. In all other respects, the architecture and training hyperparameters of LP-BERT are kept identical across the two fusion strategies. The experimental results are summarized in Table 7. As shown in Table 7, concatenation-based fusion achieves slightly higher prediction accuracy than addition-based fusion. However, concatenation doubles the dimensionality of the representations fed into LP-BERT, which may

Table 6. Dimensionality of Representations in Each Component of CLIFT

Component	*Input and Output Dimension*
General Lifestyle Pattern Encoder	128
City-Specific Mobility Pattern Encoder	128
Downstream LP-BERT Addition	128
Downstream LP-BERT Concatenation	256

Table 7. Prediction Results for Different Fusion Strategies Across Cities [GEOBLEU (GEO)↑, Top-1↑, and Top-5↑]

Model	*City A (100,000)*			*City B (20,000)*			*City C (15,000)*			*City D (3,000)*		
	GEO	Top-1	Top-5	GEO	Top-1	Top-5	GEO	Top-1	Top-5	GEO	Top-1	Top-5
CLIFT Addition	0.3460	32.06	58.53	0.3281	30.99	57.49	0.3228	31.05	55.62	0.2946	26.96	53.55
CLIFT Concatenation	0.3460	32.17	58.85	0.3290	31.06	57.99	0.3248	31.20	56.32	0.3033	27.55	54.97

increase representational capacity but also incurs higher computational cost. In CLIFT, LP-BERT can effectively learn from the combined representations, and addition-based fusion therefore maintains strong performance while keeping the model more computationally efficient.

6 Discussions

In this section, we further discuss the performance and characteristics of CLIFT. We first compare its prediction performance under the same experimental setting and dataset as the international competition HuMob Challenge 2024. We then examine its prediction performance under different prediction conditions, and finally analyze how city characteristics influence the learned lifestyle pattern representations.

6.1 Comparison with the HuMob Challenge 2024 Results

The dataset used in this study, "LYMob-4Cities", is the same dataset employed in the international competition HuMob Challenge 2024. Since the scores of the top-performing teams in the HuMob Challenge 2024[1] are publicly available, we can directly compare our prediction performance with theirs under the same task setting. To this end, we conducted an experiment comparing CLIFT with the top-ranked teams from the challenge. This experiment used the full version of the LYMob-4Cities dataset (City A: 100,000 users; City B: 25,000 users; City C: 20,000 users; City D: 6,000 users). Following the official task definition, the goal was to predict the final 15 days of 30-minute-interval movements after day 60 for 3,000 users from each of City B, City C, and City D. Evaluation was conducted using GEOBLEU and DTW (Dynamic Time Warping) [45], following the official evaluation scripts released by the organizers; higher GEOBLEU and lower DTW indicate better performance. The final score was computed as the average performance across users in the three target cities. For comparison, we adopted the top-3 teams in GEOBLEU [36, 46, 47] and the top-3 teams in DTW [47–49] as baselines, which include Transformer-based models and LLM-based methods. For training CLIFT, we adopted the same configuration as in the HuMob Challenge 2024, using all 75 days of mobility histories from non-target users in Cities A–D together with the first 60 days of the target users' trajectories. All training hyperparameters were the same as those used in Section 5.3. The prediction results are summarized in Table 8. CLIFT outperformed the top-ranked approaches in the HuMob Challenge 2024 in both GEOBLEU and DTW, achieving superior performance over a wide range of methods under the same dataset and task setting.

[1] https://wp.nyu.edu/humobchallenge2024/final-results/

Table 8. Prediction Results for Comparison With the HuMob Challenge 2024 [GEOBLEU (GEO)↑, DTW↓]

Model	*Average (City B, City C, City D)*	
	GEO↑	DTW↓
1st prize GEOBLEU [46]	0.319	28.21
2nd prize GEOBLEU [47]	0.309	27.96
3rd prize GEOBLEU [36]	0.305	30.45
1st prize DTW [48]	0.290	27.15
2nd prize DTW [49]	0.226	27.70
3rd prize DTW [47]	0.309	27.96
CLIFT (Area2Vec) Addition	**0.320**	**25.81**
CLIFT (Area2Vec) Concatenation	**0.322**	**25.72**

The bold entries indicate the results of the proposed method (CLIFT).

6.2 Evaluation of Prediction Performance under Varying Conditions

In this section, we evaluate the prediction performance of CLIFT under different prediction conditions. The experiments in Section 5 used a fixed setting: a 60-day input history and a 15-day prediction window. Here, we extend this evaluation to two alternative prediction windows (7 and 10 days) and to incomplete input settings in which 25% or 50% of the input mobility history is randomly removed. As baselines, we use LP-BERT and Cross-city BERT, which showed strong prediction performance in Section 5. No additional training is conducted in this section; instead, we perform prediction only under each condition using the models trained in Section 5.

Figure 6 shows the relationship between prediction window length and prediction accuracy, and Figure 7 shows the relationship between the input trajectory ratio and prediction accuracy. As shown in Figure 6, prediction accuracy tends to increase as the prediction window becomes shorter. This result can be attributed to the fact that longer prediction windows involve greater uncertainty in future mobility sequences, making it more difficult to predict future behavior from the mobility history observed up to the present. In contrast, in City D, which has the smallest number of users, the difference in prediction accuracy across prediction windows was limited. This suggests that, because of the small amount of training data in City D, the models may not have captured mobility patterns in sufficient detail, making differences in prediction difficulty across window lengths less likely to be reflected in the prediction accuracy. Figure 7 also shows that prediction accuracy tends to improve as the observed ratio of the input trajectories increases. This result indicates that richer input trajectories enable the models to capture users' mobility context more accurately, leading to better prediction performance. Although similar trends are observed for the baseline models, CLIFT maintains higher prediction performance than the other methods under altered inference conditions.

6.3 Regional Characteristics of Lifestyle Patterns

In the experiments described in Section 5, the general lifestyle pattern encoder—one of the core components of CLIFT—was trained on mobility data combined from all four cities. However, certain aspects of lifestyle patterns can differ across cities due to regional differences. To examine how the amount and diversity of training data affect its representational capacity, we conducted additional experiments in which only the training data for the general lifestyle pattern encoder were varied. In these experiments, we reused the same set of Area2Vec-based urban function clusters introduced in Section 5 and kept them fixed across all settings. We then considered the following conditions:

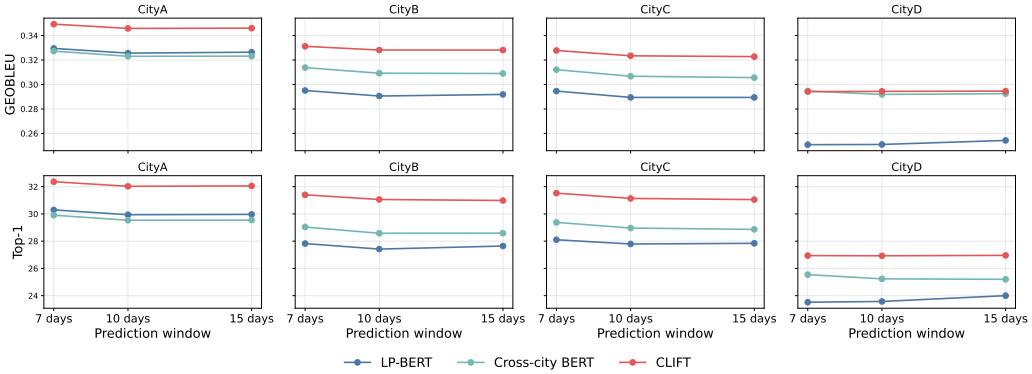

Fig. 6. Prediction accuracy under different prediction windows

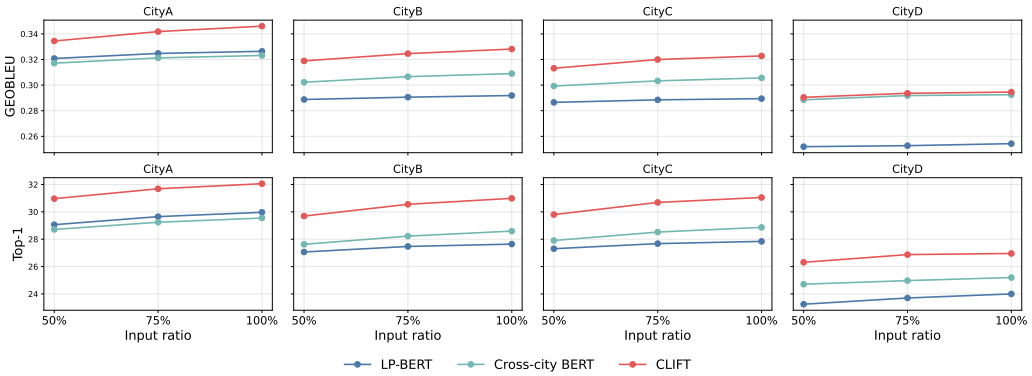

Fig. 7. Prediction accuracy under different input trajectory ratios

— Trained on users from all four cities (baseline: full multi-city training as in the experiments of Section 5).
— Trained on 100% of users in City A (100,000 users).
— Trained on 50% of users in City A (50,000 users).
— Trained on 25% of users in City A (25,000 users).
— Trained on all users in City B (20,000 users).
— Trained on all users in City C (15,000 users).

In all cases, the city-specific mobility pattern encoder and LP-BERT were trained with the same configurations as in Section 5. The experimental results are shown in Figure 8, where all scores are normalized so that the encoder trained on users from all four cities (full multi-city training) has a relative value of 1.0. The best performance was achieved when the general lifestyle pattern encoder was trained on data from all cities, indicating that learning from diverse cross-city behaviors improves its generalization ability. In City A, accuracy remained high even with reduced training data, since abundant mobility records captured local mobility patterns and mitigated the effect of reduced encoder capacity. In the medium-scale and small-scale cities (Cities B, C, and D), performance declined more clearly as the training data decreased. These results indicate that the representational strength of the general lifestyle pattern encoder grows with both the scale and

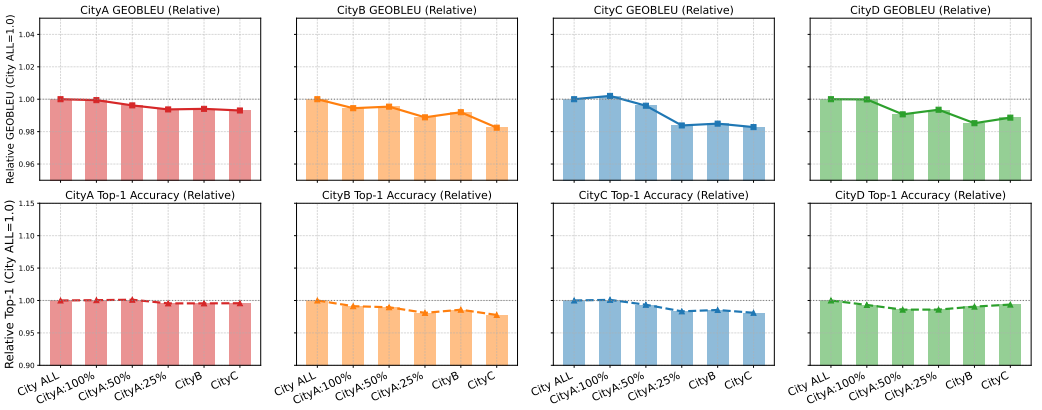

Fig. 8. Comparison of general lifestyle pattern encoders trained with varying datasets. The upper row shows GEOBLEU scores and the lower row shows Top-1 accuracy. All scores are shown as relative performance, normalized so that the encoder trained on all users from all cities in the experiments of Section 5 (the baseline: full multi-city training) has a value of 1.0.

diversity of its training data, making aggregated multi-city datasets effective for learning robust lifestyle representations. Moreover, training the general lifestyle pattern encoder on data from a single source city still improves prediction in another target city, suggesting that the learned lifestyle patterns are transferable across cities. Nevertheless, the current approach does not explicitly model factors such as seasonal variations or long-term shifts in mobility behavior. In addition, since the model is trained solely on data from Japanese cities, it may not generalize to cities in other countries where mobility behaviors differ substantially. Incorporating more diverse datasets spanning longer time periods and multiple countries to address these limitations remains an important direction for future work.

7 Conclusion

In this study, we proposed **CLIFT**, a novel human mobility prediction framework that simultaneously captures both general lifestyle patterns shared across cities and city-specific mobility patterns. CLIFT employs two complementary pre-trained encoders—one learns transferable lifestyle patterns across cities and the other captures local mobility tendencies—and integrates them through the downstream predictor LP-BERT. This design enables CLIFT to jointly leverage global and local characteristics. Experimental results on the real-world multi-city human mobility dataset show that CLIFT consistently outperforms existing baseline models across multiple evaluation metrics. On average, CLIFT improves GEOBLEU from 0.2905 to 0.3229 and Top-1 accuracy from 27.36% to 30.27%, corresponding to absolute gains of 0.032 and 2.91 percentage points, respectively, over the single-city baseline. Compared to the multi-city baseline, CLIFT improves GEOBLEU from 0.3076 to 0.3229 and Top-1 accuracy from 28.05% to 30.27%, corresponding to absolute gains of 0.015 and 2.21 percentage points, respectively. The ablation study further confirms the contribution of each component, demonstrating the overall effectiveness of the CLIFT architecture. Moreover, CLIFT outperformed the top-ranked teams in the international competition *HuMob Challenge 2024*, demonstrating superior predictive performance under the same dataset and task setting.

As future work, we aim to extend the framework in three directions. First, while the current anonymized dataset does not disclose city identities and thus prevented us from integrating external geographic resources, we plan to extend our experiments to datasets with known city identities,

incorporating richer urban context information such as street-view images, POI metadata, official land-use maps, and textual descriptions of individual locations, which are expected to capture cross-city similarities in greater detail and further improve prediction accuracy. Second, we will study how to adapt the learned lifestyle and mobility representations to unseen cities, including zero-shot and few-shot settings under distribution shifts such as seasonal changes and long-term trends. Third, we will examine its computational efficiency and practical applicability, for example, in large-scale demand forecasting and anomaly detection in real-world urban mobility systems. Through these extensions, our ultimate goal is to develop a multimodal foundation model for human mobility prediction that can robustly generalize to unseen cities by leveraging knowledge from diverse heterogeneous modalities.

References

[1] Carlo Ratti, Dennis Frenchman, Riccardo Maria Pulselli, and Sarah Williams. 2006. Mobile landscapes: Using location data from cell phones for urban analysis. *Environment and Planning B: Planning and Design* 33, 5 (2006), 727–748.

[2] Shan Jiang, Joseph Ferreira, and Marta C. Gonzalez. 2017. Activity-based human mobility patterns inferred from mobile phone data: A case study of Singapore. *IEEE Transactions on Big Data* 3, 2 (2017), 208–219.

[3] Takahiro Yabe, Nicholas K. W. Jones, P. Suresh C. Rao, Marta C. Gonzalez, and Satish V. Ukkusuri. 2022. Mobile phone location data for disasters: A review from natural hazards and epidemics. *Computers, Environment and Urban Systems* 94 (2022).

[4] Zipei Fan, Chuang Yang, Zhiwen Zhang, Xuan Song, Yinghao Liu, Renhe Jiang, Quanjun Chen, and Ryosuke Shibasaki. 2022. Human mobility based individual-level epidemic simulation platform. *ACM Transactions on Spatial Algorithms and Systems* 8, 3 (2022), 1–16.

[5] Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. 2016. Predicting the next location: A recurrent model with spatial and temporal contexts. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 194–200.

[6] Dejiang Kong and Fei Wu. 2018. HST-LSTM: A hierarchical spatial-temporal long-short term memory network for location prediction. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*. 2341–2347.

[7] Ye Hong, Henry Martin, and Martin Raubal. 2022. How do you go where? Improving next location prediction by learning travel mode information using transformers. In *Proceedings of the ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. 1–10.

[8] Kazuyuki Shoji, Shunsuke Aoki, Takuro Yonezawa, and Nobuo Kawaguchi. 2024. Area modeling using stay information for large-scale users and analysis for influence of COVID-19. *IPSJ Journal* 62, 10 (2024), 1644–1657.

[9] Haru Terashima, Naoki Tamura, Kazuyuki Shoji, Shin Katayama, Kenta Urano, Takuro Yonezawa, and Nobuo Kawaguchi. 2023. Human mobility prediction challenge: Next location prediction using spatiotemporal BERT. In *Proceedings of the 1st ACM SIGSPATIAL International Workshop on Human Mobility Prediction Challenge*. 1–6.

[10] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the International Conference on Neural Information Processing Systems* (2017), 5998–6008.

[11] Takahiro Yabe, Kota Tsubouchi, Toru Shimizu, Yoshihide Sekimoto, Kaoru Sezaki, Esteban Moro, and Alex Pentland. 2024. YJMob100K: City-scale and longitudinal dataset of anonymized human mobility trajectories. *Scientific Data* 11, 1 (2024), 397.

[12] Retrieved from https://wp.nyu.edu/humobchallenge2024/. (2024).

[13] Massimiliano Luca, Gianni Barlacchi, Bruno Lepri, and Luca Pappalardo. 2021. A survey on deep learning for human mobility. *ACM Computing Surveys* 55, 1, Article 7 (Nov. 2021), 44 pages.

[14] Luca Pappalardo, Ed Manley, Vedran Sekara, and Laura Alessandretti. 2023. Future directions in human mobility science. *Nature Computational Science* 3, 7 (2023), 588–600.

[15] Mohamed Mokbel, Mahmoud Sakr, Li Xiong, Andreas Züfle, Jussara Almeida, Taylor Anderson, Walid Aref, Gennady Andrienko, Natalia Andrienko, Yang Cao, et al. 2024. Mobility data science: Perspectives and challenges. *ACM Transactions on Spatial Algorithms and Systems* 10, 2, Article 10 (2024), 35 pages.

[16] Kai Zhao, Sasu Tarkoma, Siyuan Liu, and Huy Vo. 2016. Urban human mobility data mining: An overview. In *Proceedings of the 2016 IEEE International Conference on Big Data (Big Data)*. 1911–1920.

[17] Marta C. Gonzalez, Cesar A. Hidalgo, and Albert-Laszlo Barabasi. 2008. Understanding individual human mobility patterns. *Nature* 453 (2008), 779–782.

[18] Soto Anno, Kota Tsubouchi, and Masamichi Shimosaka. 2024. Forecasting lifespan of crowded events with acoustic synthesis-inspired segmental long short-term memory. *IEEE Access* 12 (2024), 87309–87322.

[19] Maria Luisa Damiani, Andrea Acquaviva, Fatima Hachem, and Matteo Rossini. 2020. Learning behavioral representations of human mobility. In *Proceedings of the ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. 367–376.

[20] Junjun Si, Jin Yang, Yang Xiang, Hanqiu Wang, Li Li, Rongqing Zhang, Bo Tu, and Xiangqun Chen. 2023. TrajBERT: BERT-based trajectory recovery with spatial-temporal refinement for implicit sparse trajectories. *IEEE Transactions on Mobile Computing* 23, 5 (2023), 4849–4860.

[21] Shang-Ling Hsu, Emmanuel Tung, John Krumm, Cyrus Shahabi, and Khurram Shafique. 2024. TrajGPT: Controlled synthetic trajectory generation using a multitask transformer-based spatiotemporal model. In *Proceedings of the ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. 362–371.

[22] Zijun Yao, Yanjie Fu, Bin Liu, Wangsu Hu, and Hui Xiong. 2018. Representing urban functions through zone embedding with human mobility patterns. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*.

[23] Kazuyuki Shoji, Haru Terashima, Naoki Tamura, Shin Katayama, Kenta Urano, Takuro Yonezawa, and Nobuo Kawaguchi. 2025. Life pattern-based human attribute estimation using only GPS data. *IEEE Access* 13 (2025), 131803–131822.

[24] Yinyin Zhang, Yongjun Li, and Wenli Ji. 2023. A trajectory-based user movement pattern similarity measure for user identification. *IEEE Transactions on Network Science and Engineering* 10, 6 (2023), 3834–3845.

[25] Shenglin Zhao, Tong Zhao, Irwin King, and Michael R. Lyu. 2017. Geo-teaser: Geo-temporal sequential embedding rank for point-of-interest recommendation. In *Proceedings of the World Wide Web conference (WWW)*. 153–162.

[26] Huaiyu Wan, Yan Lin, Shengnan Guo, and Youfang Lin. 2021. Pre-training time-aware location embeddings from spatial-temporal trajectories. *IEEE Transactions on Knowledge and Data Engineering* 34, 11 (2021), 5510–5523.

[27] Toru Shimizu, Takahiro Yabe, and Kota Tsubouchi. 2020. Enabling finer grained place embeddings using spatial hierarchy from human mobility trajectories. In *Proceedings of the ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. 187–190.

[28] Yingtao Luo, Qiang Liu, and Zhaocheng Liu. 2021. STAN: Spatio-temporal attention network for next location recommendation. In *Proceedings of the World Wide Web conference (WWW)*. 2177–2185.

[29] Amiya Bhattacharya and Sajal K. Das. 1999. LeZi-update: An information-theoretic approach to track mobile users in PCS networks. In *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*. 1–12.

[30] Sébastien Gambs, Marc-Olivier Killijian, and Miguel Núñez del Prado Cortez. 2012. Next place prediction using mobility markov chains. In *Proceedings of the 1st Workshop on Measurement, Privacy, and Mobility*. 1–6.

[31] Jianwei Chen, Jianbo Li, and Ying Li. 2020. Predicting human mobility via long short-term patterns. In *Proceedings of the Computer Modeling in Engineering & Sciences*, Vol. 124. 847–864.

[32] Dingqi Yang, Benjamin Fankhauser, Paolo Rosso, and Philippe Cudre-Mauroux. 2020. Location prediction over sparse user mobility traces using RNNs. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*. 2184–2190.

[33] Jie Feng, Yong Li, Chao Zhang, Funing Sun, Fanchao Meng, Ang Guo, and Depeng Jin. 2018. Deepmove: Predicting human mobility with attentional recurrent networks. In *Proceedings of the World Wide Web conference (WWW)*. 1459–1468.

[34] Shengren Ke, Meiyi Xie, Hong Zhu, and Zhongsheng Cao. 2022. Group-based recurrent neural network for human mobility prediction. *Neural Computing and Applications* 34, 12 (2022), 9863–9883.

[35] Bangchao Deng, Bingqing Qu, Pengyang Wang, Dingqi Yang, Benjamin Fankhauser, and Philippe Cudre-Mauroux. 2025. REPLAY: Modeling time-varying temporal regularities of human mobility for location prediction over sparse trajectories. *IEEE Transactions on Mobile Computing* 24, 10 (2025), 9428–9440.

[36] Meisaku Suzuki, Yusuke Fukushima, Ryo Koyama, Hayato Kumagai, Tomohiro Mimura, and Keiichi Ochiai. 2024. Cross-city-aware spatiotemporal BERT. In *Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Human Mobility Prediction Challenge*. 33–36.

[37] Jonas Gunkel, Andrea Tundis, and Max Mühlhäuser. 2024. The story of mobility: Combining state space models and transformers for multi-step trajectory prediction. In *Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Human Mobility Prediction Challenge*. 19–24.

[38] Yan Lin, Huaiyu Wan, Shengnan Guo, and Youfang Lin. 2021. Pre-training context and time aware location embeddings from spatial-temporal trajectories for user next location prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 4241–4248.

[39] David Arthur and Sergei Vassilvitskii. 2007. k-means++: The advantages of careful seeding. In *Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms*. 1027–1035.

[40] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (2018), 4171–4186.

[41] Toru Shimizu, Kota Tsubouchi, and Takahiro Yabe. 2022. GEO-BLEU: Similarity measure for geospatial sequences. In *Proceedings of the ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. 1–4.

[42] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*. 311–318.

[43] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation* 9, 8 (1997), 1735–1780.

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

[45] Pavel Senin. 2008. Dynamic time warping algorithm review. *Information and Computer Science Department University of Hawaii at Manoa Honolulu, USA* 855 (2008), 1–23.

[46] Haru Terashima, Shun Takagi, Naoki Tamura, Kazuyuki Shoji, Tahera Hossain, Shin Katayama, Kenta Urano, Takuro Yonezawa, and Nobuo Kawaguchi. 2024. Time-series stay frequency for multi-city next location prediction using multiple BERTs. In *Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Human Mobility Prediction Challenge*. 5–9.

[47] Peizhi Tang, Chuang Yang, Tong Xing, Xiaohang Xu, Renhe Jiang, and Kaoru Sezaki. 2024. Instruction-tuning Llama-3-8B excels in city-scale mobility prediction. In *Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Human Mobility Prediction Challenge*. 1–4.

[48] Haoyu He, Haozheng Luo, and Qi R. Wang. 2024. ST-MoE-BERT: A spatial-temporal mixture-of-experts framework for long-term cross-city mobility prediction. In *Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Human Mobility Prediction Challenge*. 10–15.

[49] Yuki Imai, Takuya Tokumoto, Kohei Koyama, Tomoko Ochi, Shogo Imai, Tomoyuki Mori, Tomohiro Nakao, and Kenta Maruyama. 2024. Urban human mobility prediction using support vector regression: A classical data-driven approach. In *Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Human Mobility Prediction Challenge*. 37–41.

Appendices

A Dataset Description

Figure 9 and Figure 4 present visualizations of the dataset used in this study. In Section 5, we use Version 2 of the "LYMob-4Cities: Multi-City Human Mobility Dataset",[2] whereas Section 6 employs the full version that was released after the competition. Figure 9 summarizes the data volume in terms of the number of users, days, and time slots. To infer the corresponding weekdays, we analyze periodic patterns in the daily data volume: days with characteristically low volume are identified as weekends, with day 0 aligned to Saturday and subsequent weekdays assigned sequentially.

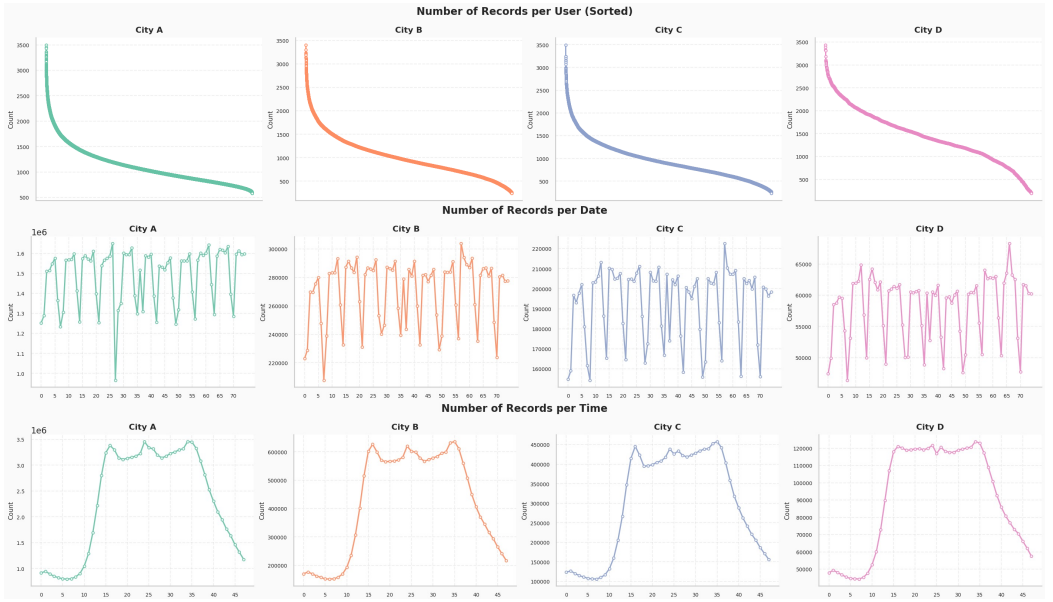

Fig. 9. Data volume per user, day, and time slot in the dataset.

B Comparison of the Number of Clusters for Urban Functions

We estimate the urban function of each location using Area2Vec-based representations: we first learn Area2Vec embeddings from mobility histories and then perform clustering on these embeddings using the k-means++ algorithm. To identify the optimal number of clusters, we conducted comparative experiments with three different cluster settings. In all settings, locations with sparse mobility data were assigned to an additional cluster. The results are summarized in Table 9. Since the 36-cluster setting achieved the highest average score, we therefore adopt 36 clusters in this study.

Table 9. Prediction Results of Different Numbers of Clusters [GEOBLEU (GEO)↑, Top-1↑, Top-5↑]

Model	*City A (100,000)*			*City B (20,000)*			*City C (15,000)*			*City D (3,000)*		
	GEO	Top-1	Top-5	GEO	Top-1	Top-5	GEO	Top-1	Top-5	GEO	Top-1	Top-5
CLIFT (16 clusters)	0.3463	32.09	58.51	0.3281	30.71	57.27	0.3226	30.87	55.54	0.2923	26.63	53.23
CLIFT (36 clusters)	0.3460	32.06	58.53	0.3281	30.99	57.49	0.3228	31.05	55.62	0.2946	26.96	53.55
CLIFT (64 clusters)	0.3467	31.97	58.48	0.3274	30.87	57.32	0.3223	30.99	55.62	0.2948	26.73	53.84

[2] https://zenodo.org/records/13237029

C Training Parameters of Baseline Models

Table 10 summarizes the hyperparameters used for the baseline models. For each baseline, we tuned hyperparameters on the available training-period data. We first adjusted the learning rate while monitoring overfitting, and then tuned the number of layers. The final configuration for each model was selected based on the Top-1 accuracy for the target users. For City D, we used a batch size of 8 during training due to its relatively small data volume.

Table 10. Hyperparameters of Baseline Models

	Layers	*Attention Heads*	*Learning Rate*	*Batch Size*	*Embed Size*	*Epochs*
LSTM	1	-	1e-4	16 (8)	128	100
ST-LSTM	1	-	1e-4	16 (8)	128	100
STGN	1	-	1e-4	16 (8)	128	100
Transformer	3	8	1e-4	16 (8)	128	100
LP-BERT	4	8	1e-4	16 (8)	128	100
Cross-city BERT	4	8	1e-4	16	128	100

Received 13 December 2025; revised 5 April 2026; accepted 8 April 2026