Building Efficient LLM Pipeline for Human Mobility Prediction

Chenhao Wang
University of Electronic Science and Technology of China
Chengdu, China
chenhao.wang@std.uestc.edu.cn

Silin Zhou
University of Electronic Science and Technology of China
Chengdu, China
zhousilinxy@gmail.com

Lisi Chen
University of Electronic Science and Technology of China
Chengdu, China
lchen012@e.ntu.edu.sg

Shuo Shang
University of Electronic Science and Technology of China
Chengdu, China
jedi.shang@gmail.com

Abstract

Human mobility prediction is a fundamental problem in spatio-temporal data mining with broad applications in urban computing and transportation systems. While large language models (LLMs) have demonstrated strong sequence modeling capabilities, directly adapting them to structured mobility data remains challenging due to long input sequences and efficiency limitations. In this study, we propose ELP-Mob, an efficient framework that reformulates mobility prediction as a language modeling problem. ELP-Mob employs an instruction-style prompt design that incorporates user mobility profiles, historical trajectories, and target future time slots, enabling LLMs to understand mobility patterns and make predictions. To further enhance efficiency, ELP-Mob includes a data selection strategy that reduces redundancy by sampling informative subsets of training users, and a dynamic splitting strategy with token-length control, which scales to long histories while reducing computational overhead. In the GISCUP 2025, ELP-Mob achieved 6th place on the official leaderboard. The source code is publicly available at https://github.com/chwang0721/ELP-Mob.

CCS Concepts

• **Information systems** → **Spatial-temporal systems**; • **Human-centered computing**;

Keywords

Human mobility prediction, Large language models

ACM Reference Format:
Chenhao Wang, Silin Zhou, Lisi Chen, and Shuo Shang. 2025. Building Efficient LLM Pipeline for Human Mobility Prediction. In *The 33rd ACM International Conference on Advances in Geographic Information Systems (SIGSPATIAL '25), November 3–6, 2025, Minneapolis, MN, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3748636.3771314

SIGSPATIAL '25, Minneapolis, MN, USA

ACM ISBN 979-8-4007-2086-4/2025/11
https://doi.org/10.1145/3748636.3771314

1 Introduction

With the rapid growth of location-aware applications, human mobility prediction has emerged as a fundamental task in spatio-temporal data mining, enabling applications such as route planning [3], traffic flow forecasting [6], and next POI recommendation [2]. Traditional mobility prediction approaches typically rely on deep learning models such as recurrent neural networks (RNNs) [1], convolutional neural networks (CNNs) [7], and diffusion models [5]. While effective, these methods are often tailored to specific input formats. In contrast, LLMs can accommodate heterogeneous trajectory representations. Furthermore, pretraining on large, diverse datasets enables reasoning abilities that extend beyond simple pattern recognition. These capabilities allow LLMs to model complex spatio-temporal dependencies and support higher-level decision-making.

Recent advances in LLMs have shown strong sequential reasoning and problem-solving capabilities, and have thus been explored for mobility prediction [4, 9]. Despite this promise, directly applying LLMs to real-world human mobility is non-trivial: trajectories are often long, noisy, and heterogeneous across users and time. Moreover, naively encoding entire histories easily overwhelms context length and computation power, undermining both scalability and efficiency. In practice, fine-tuning on large-scale mobility datasets may even require several months [9].

To move beyond these limitations, two key challenges must be addressed. **(1) Representation and personalization.** Mobility trajectories need to be reformulated into textual forms that preserve spatio-temporal dependencies while embedding user-specific priors (e.g., top-k frequently visited locations) to capture individual heterogeneity. **(2) Efficiency at scale.** The length of user histories and the number of users increase token budgets and computational overhead. Even with a modest number of training trajectories, fine-tuning is time-consuming because each trajectory is lengthy.

To tackle these challenges, we design **ELP-Mob** (Efficient LLM Pipeline for Human Mobility Prediction), an end-to-end pipeline that reformulates mobility forecasting as a language modeling problem. ELP-Mob encodes each prediction instance as a coherent textual prompt integrating user priors, standardized historical trajectories, and future time slots. The pipeline enhances efficiency through two main strategies: (1) compressing and prioritizing informative training signals so that the model concentrates on trajectories most relevant to the target distribution, and (2) enforcing token-length control via adaptive partitioning of long sequences, thereby preserving spatio-temporal coherence while staying within context

limits. Finally, ELP-Mob fine-tunes the LLM on the mobility objective using LoRA, enabling accurate and personalized predictions with modest computational and memory cost.

Our main contributions are summarized as follows:

- We introduce ELP-Mob, a novel framework that reformulates human mobility prediction as a language modeling problem by leveraging instruction-style prompt design.
- We develop two efficiency-oriented strategies: a data selection strategy for reducing data redundancy, and a dynamic splitting strategy with token-length control for scalable modeling.
- We evaluate ELP-Mob in the GISCUP 2025[1], where it achieved 6th place, demonstrating both its effectiveness and efficiency in large-scale mobility forecasting.

2 Problem Statement

We proceed to present relevant definitions and formally state the human mobility prediction problem.

Definition 1 (Trajectory). A trajectory $T = \langle p_1, p_2, \ldots, p_n \rangle$ represents the movement of an object as an ordered sequence of points with strictly increasing timestamps. Each point is defined as a pair $p_i = (t_i, l_i)$, where t_i is a discrete timestamp and l_i denotes the spatial location of the object at time t_i.

Definition 2 (Mobility Sequence). In this work, we adopt a grid–timeslot discretization of space and time. Let the geographical space be partitioned into a two-dimensional grid, where each grid cell corresponds to a fixed-size region, and let the temporal axis be divided into uniform time slots of length Δt. We define two mappings: (i) a spatial mapping that assigns each continuous location l_i to its grid cell (x_i, y_i), and (ii) a temporal mapping that decomposes each timestamp t_i into day and timeslot indices (d_i, τ_i). The resulting mobility sequence is the discrete spatio-temporal record:

$$S = \big\langle (d_1, \tau_1, x_1, y_1),\ (d_2, \tau_2, x_2, y_2),\ \ldots,\ (d_n, \tau_n, x_n, y_n) \big\rangle. \quad (1)$$

Problem Statement (Human Mobility Prediction). Given a user u with a historical mobility sequence S^{H} observed over the time horizon $\{(d_1, \tau_1), \ldots, (d_m, \tau_m)\}$, we are provided with a sequence of known future day–timeslot pairs:

$$\mathcal{I}^{\mathrm{F}} = \langle (d_{m+1}, \tau_{m+1}), \ldots, (d_{m+n}, \tau_{m+n}) \rangle, \quad (2)$$

the goal is to learn a predictive function

$$f_\theta :\ (S^{\mathrm{H}}, \mathcal{I}^{\mathrm{F}}) \ \mapsto\ \hat{S}^{\mathrm{F}}, \quad (3)$$

such that $\hat{S}^{\mathrm{F}}$ approximates the ground truth S^{F}.

3 Methodology

The proposed ELP-Mob is designed as a four-stage framework, as shown in Figure 1. Specifically, the pipeline consists of: (1) a *data selection* strategy that filters and prioritizes informative users to enhance both efficiency and generalization; (2) a *prompt construction* module that reformulates spatio-temporal trajectories into instruction-style textual prompts, incorporating user profiles, historical trajectories, and future slots; (3) a *dynamic splitting* strategy that partitions long mobility sequences into manageable segments under token-length constraints while preserving prediction fidelity;

[1]https://sigspatial2025.sigspatial.org/giscup/index.html

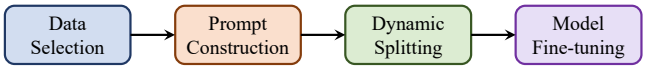

Figure 1: An overview of the pipeline.

and (4) a *model fine-tuning* module, where LoRA-based adaptation is applied to align the LLM with the mobility prediction objective.

3.1 Data Selection Strategy

Human mobility datasets typically encompass a large number of users, but not all of them contribute equally to the prediction task. Incorporating data from irrelevant or weakly related users can introduce noise and increase computational overhead. To this end, we propose a user-level data selection strategy that explicitly identifies and prioritizes training users whose mobility patterns exhibit strong relevance to the test set, thereby improving both the efficiency and the effectiveness of LLM fine-tuning for mobility prediction.

For each user u, we construct the set of visited grid cells from their historical mobility sequence:

$$G(u) = \{(x_i, y_i) \ \mid\ (d_i, \tau_i, x_i, y_i) \in S_u^{\mathrm{H}}\}. \quad (4)$$

Let $\mathcal{U}^{\mathrm{test}}$ denote the set of test users. The relevance of a candidate training user u_{train} is quantified by an overlap score defined as

$$\mathrm{Score}(u_{\mathrm{train}}) = \frac{1}{|\mathcal{U}^{\mathrm{test}}|} \sum_{u_{\mathrm{test}} \in \mathcal{U}^{\mathrm{test}}} \frac{|G(u_{\mathrm{train}}) \cap G(u_{\mathrm{test}})|}{|G(u_{\mathrm{test}})|}, \quad (5)$$

which measures the average spatial coverage of test users explained by a candidate training user. Intuitively, higher values indicate stronger alignment of spatial behaviors with the test set, making the candidate more informative for training.

Finally, candidate training users are ranked by their overlap scores, and the top-K are selected. This procedure ensures that fine-tuning emphasizes the most representative trajectories of the test set, thereby improving training efficiency and model generalization.

3.2 Prompt Construction

To adapt LLMs for human mobility prediction, we reformulate the task as an instruction-following problem by constructing structured prompts. Each prompt instance consists of three key components: the instruction block, the input block, and the output block.

Instruction block. This block defines the model's role, the spatial–temporal environment, the trajectory representation, the prediction task, and the required output. Specifically, it describes the city partitioned, the discretization of time , the record format and the requirement to fill in missing future locations.

Input block. The input consists of three subcomponents:

- **[User Profile]**, which summarizes the most frequently visited locations and their proportions;
- **[Historical Trajectory]**, which records the user's past movements in a standardized format;
- **[Future Time Slots]**, which specify the time points requiring location prediction, with missing values marked by placeholders.

Together, these components provide both personalized prior knowledge and task-specific context for prediction.

Output block. The output is the completed list of records corresponding to the future slots, with each entry expressed in the same

Instruction

[Role]
You are a human mobility prediction assistant.
[Environment]
The city is divided into a 200 × 200 grid, each cell representing a 500m × 500m area. The top-left corner is (1,1); the bottom-right is (200,200). Time is discretized into 30-minute intervals, giving 48 time slots per day.
[Trajectory Format]
Each record is formatted as: <day_id> <timeslot_id> <x> <y>. For example: `12 16 103 88` means on day 12, at time slot 16 (7:30am–8:00am), the person was at cell (103,88).
[Task]
You will receive: 1. [User Profile]: Top-K most frequently visited locations with their proportions. 2. [Trajectory History]: Known locations from day 1 to day 60. 3. [Future Time Slots]: From day 61 to day 75, with missing locations (represented as `999 999`).
Your task: Predict the missing locations for all [Future Time Slots], leveraging both the [User Profile] and the [Trajectory History].
[Output]
Return a list of records `<day_id> <timeslot_id> <x> <y>`. Predictions must correspond exactly to the missing entries in [Future Time Slots]. Maintain the same order as they appear in [Future Time Slots].

Input

[User Profile]
TOPK=5: (112,86)@12.02%,(86,41)@11.24%,(111,86)@5.04%,(87,40)@3.1%,(87,41)@2.33%
[Trajectory History]
1 35 106 69\n1 36 107 70\n1 38 112 86\n ... \n60 40 106 55\n60 41 106 56\n60 42 112 86
[Future Time Slots]
61 15 999 999\n61 16 999 999\n61 17 999 999\n ... \n75 31 999 999\n75 35 999 999\n75 36 999 999

Output

[Prediction]
61 15 102 73\n61 16 90 56\n61 17 86 42\n ... \n75 31 127 79\n75 35 124 86\n75 36 112 86

Figure 2: Prompt example used in ELP-Mob.

standardized format as the input. This structured design ensures direct alignment between queries and predictions, enabling effective supervision during fine-tuning.

3.3 Dynamic Splitting Strategy

While the data selection strategy reduces the number of training instances, the mobility sequences themselves remain lengthy. Directly encoding these sequences could exceed LLMs' token limits, and the efficiency of LLMs' training and inference is also constrained by token length. To this end, we introduce a dynamic splitting strategy that partitions trajectories into manageable segments while retaining the necessary information for accurate prediction. The process of the dynamic splitting strategy is shown in Algorithm 1.

Specifically, the historical and future mobility sequences are partitioned into interleaved subsets through interval sampling, ensuring that each segment remains within the token limit while preserving temporal coverage of the entire trajectory. During inference, a complete mobility sequence can be divided into parallel segments for prediction and then reassembled in chronological order, thereby improving overall efficiency.

3.4 Model Fine-tuning

To adapt large language models (LLMs) to the human mobility prediction task, we apply parameter-efficient fine-tuning using Low-Rank Adaptation (LoRA) within the LLaMA Factory framework [11][2]. LoRA substantially reduces the number of trainable parameters while preserving model capacity, making it well suited for large-scale mobility datasets.

[2]https://github.com/hiyouga/LLaMA-Factory

Algorithm 1: Dynamic Splitting Strategy

Input: Historical trajectory S^H, future sequence S^F, cutoff length $L_{\max}$, maximum partitions $N_{\max}$
Output: Set of valid prompts $\mathcal{P}$

for $n \leftarrow 1$ **to** $N_{\max}$ **do**
  Partition S^H and S^F into n interleaved subsets $\{S_i^H\}, \{S_i^F\}$ by interval sampling;
  $\mathcal{P} \leftarrow \{(S_i^H, S_i^F)\}_{i=1}^n$;
  if $\max_i length(P_i) \le L_{\max}$ **then**
    return $\mathcal{P}$;
return $\emptyset$;

Given a set of prompt–response pairs $\{(X_i, Y_i)\}_{i=1}^N$ generated by our data pipeline, the fine-tuning objective is to minimize the negative log-likelihood (NLL) of the target sequences:

$$\mathcal{L}(\theta) = -\sum_{i=1}^{N} \log P_\theta(Y_i \mid X_i), \tag{6}$$

where θ denotes the trainable LoRA parameters. This objective encourages the model to generate accurate mobility predictions conditioned on the instruction-style prompts.

In practice, we adopt mixed-precision training and gradient accumulation to improve efficiency. By combining parameter-efficient adaptation with our prompt construction strategy, the proposed approach achieves strong predictive performance while maintaining low computational and memory overhead.

4 Experiments

4.1 Experimental Setup

4.1.1 Datasets. Our study is based on the dataset provided by the GISCUP 2025, which contains large-scale synthetic mobility traces derived from anonymized smartphone application data [10]. The dataset spans four Japanese cities (A, B, C, and D), with each individual's movement represented as a sequence of spatio-temporal records. Spatial locations are discretized into a 200×200 grid, where each cell corresponds to a 500m × 500m area, and the temporal dimension is divided into 48 half-hour slots per day.

The data includes movement histories for 150,000 users in city A, 30,000 in city B, 25,000 in city C, and 20,000 in city D. Among these, the prediction task focuses on 3,000 target individuals in each city, whose trajectories during days 61–75 are partially masked by the placeholder value "999". Notably, while cities A and B feature continuous temporal coverage from days 1–75, cities C and D present a temporal gap, where the prediction period (days 61–75) occurs months after the training period (days 1–60), posing an additional challenge for generalization.

For each dataset, 15,000 users are chosen for training via the proposed data selection strategy, and 500 users are randomly sampled for validation. The sequences from the training, validation, and test sets are then used to construct prompts using the dynamic splitting strategy. A summary of dataset statistics is provided in Table 1.

Table 1: Dataset Statistics.

Dataset	A	B	C	D
# Users	150,000	30,000	25,000	20,000
# Train users	15,000	15,000	15,000	15,000
# Validate users	500	500	500	500
# Test users	3,000	3,000	3,000	3,000
# Train prompts	40,889	37,670	27,645	28,619
# Validate prompts	1,153	1,196	795	818
# Test prompts	6,589	7,149	4,341	4,597

Table 2: Performance comparison.

Methods	A	B	C	D	Mean
Per-User Mode	0.07984	0.08116	0.10789	0.10296	0.09296
Bigram Model	0.04687	0.05492	0.06249	0.06212	0.05660
Bigram Model (top-p 0.7)	0.09384	0.09232	0.07039	0.07997	0.08413
ELP-Mob	**0.13719**	**0.13286**	**0.16767**	**0.16237**	**0.15002**

4.1.2 Baselines. We compare ELP-Mob against several officially implemented baselines[3], including Per-User Mode, Bigram Model, and Bigram Mode (top-p = 0.7). In addition, we evaluate its efficiency relative to Llama-3-8B-Mob [9], the leading approach in the Human Mobility Prediction Challenge 2024.

4.1.3 Evaluation Metric. We evaluate model performance using GEO-BLEU [8], a similarity measure designed for geospatial sequences. GEO-BLEU extends the widely used BLEU score from natural language processing by introducing the notion of spatial n-grams. Instead of requiring exact matches between predicted and reference trajectories, GEO-BLEU assigns higher scores to n-grams that are geographically close, thereby accounting for spatial proximity. This property makes GEO-BLEU particularly suitable for trajectory prediction tasks.

4.1.4 Implementation Details. All experiments are conducted on a server with 4 NVIDIA GeForce RTX 4090 GPUs. We adopt Llama-3.2-3B-Instruct[4] as the backbone LLM, with a cutoff length of 4,096 tokens. For fine-tuning, we employ LoRA with a rank of 16, with a per-device batch size of 1 and gradient accumulation steps of 8, yielding an effective batch size of 32. The learning rate is set to 5×10^{-4} with a cosine scheduler, warmup ratio of 0.05, and a total of 3 epochs. During inference, we use a per-device batch size of 4, with temperature set to 0.1, top p set to 1, and top k set to 200.

4.2 Effectiveness Evaluation

As shown in Table 2, ELP-Mob consistently outperforms all baselines across datasets A–D. Compared with the strongest baseline (Bigram Model with top-p = 0.7), ELP-Mob achieves absolute improvements of up to 0.074 in accuracy (Dataset C) and delivers the highest mean score of 0.15002, which is substantially higher than the second-best mean (0.09296). These results demonstrate that ELP-Mob captures mobility patterns more effectively than both unigram- and bigram-based methods.

[3]https://sigspatial2025.sigspatial.org/giscup/problem.html
[4]https://huggingface.co/meta-llama/Llama-3.2-3B-Instruct

Table 3: Efficiency comparison on city A.

Methods	GPU Mem. (Training)	Training Time	Inference Time
Llama-3-8B-Mob	23.93 GiB	74.68 h (estimated)	10.84 s/user
ELP-Mob	14.81 GiB	6.27 h	2.52 s/user

4.3 Efficiency Evaluation

Table 3 further highlights the efficiency advantages of ELP-Mob. It reduces GPU memory usage from 23.93 GiB to 14.81 GiB, requiring only 6.27 h for training compared with 74.68 h for Llama-3-8B-Mob. In inference, ELP-Mob achieves a speedup of more than 4× (2.52 s vs. 10.84 s). This significant improvement in both computational efficiency and scalability makes ELP-Mob highly suitable for real-world human mobility prediction tasks.

5 Conclusion

In this work, we proposed ELP-Mob, an efficient LLM-based framework for human mobility prediction. By reformulating trajectory forecasting as a language modeling task, ELP-Mob integrates per-user mobility profiles, historical trajectories, and target future slots into instruction-style prompts. To further improve scalability, we designed a trajectory selection strategy and a dynamic splitting strategy with token-length control, which together reduce computational overhead while preserving predictive accuracy. Extensive experiments on the GISCUP 2025 datasets demonstrate that ELP-Mob significantly outperforms official baselines, while achieving faster training and inference compared with other LLM-based methods such as Llama-3-8B-Mob. In future work, we plan to explore cross-city generalization to further enhance the applicability of LLMs in real-world mobility prediction tasks.

References

[1] 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 *WWW*. 1459–1468.

[2] Shanshan Feng, Feiyu Meng, Lisi Chen, Shuo Shang, and Yew Soon Ong. 2024. ROTAN: A Rotation-based Temporal Attention Network for Time-Specific Next POI Recommendation. In *KDD*. 759–770.

[3] Ke Li, Lisi Chen, and Shuo Shang. 2020. Towards Alleviating Traffic Congestion: Optimal Route Planning for Massive-Scale Trips. In *IJCAI*. 3400–3406.

[4] Yuebing Liang, Yichao Liu, Xiaohan Wang, and Zhan Zhao. 2024. Exploring large language models for human mobility prediction under public events. *Comput. Environ. Urban Syst.* 112 (2024), 102153.

[5] Qingyue Long, Yuan Yuan, and Yong Li. 2025. A Universal Model for Human Mobility Prediction. In *KDD*. 894–905.

[6] Xuan Rao, Lisi Chen, Yong Liu, Shuo Shang, Bin Yao, and Peng Han. 2022. Graph-Flashback Network for Next Location Recommendation. In *KDD*. 1463–1471.

[7] Bilong Shen, Xiaodan Liang, Yufeng Ouyang, Miaofeng Liu, Weimin Zheng, and Kathleen M. Carley. 2018. StepDeep: A Novel Spatial-temporal Mobility Event Prediction Framework based on Deep Neural Network. 724–733.

[8] Toru Shimizu, Kota Tsubouchi, and Takahiro Yabe. 2022. GEO-BLEU: similarity measure for geospatial sequences. In *SIGSPATIAL*. 17:1–17:4.

[9] 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, HuMob2024*. 1–4.

[10] 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.

[11] Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, and Yongqiang Ma. 2024. LlamaFactory: Unified Efficient Fine-Tuning of 100+ Language Models. *CoRR* abs/2403.13372 (2024).