Proceedings Paper:

Sun, T., Chen, M., Zhang, B. et al. (3 more authors) (2025) SILO: Semantic Integration for Location Prediction with Large Language Models. In: KDD '25: Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2. KDD '25: The 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 03-07 Aug 2025, Toronto, Canada. Association for Computing Machinery, pp. 2756-2767. ISBN: 979-8-4007-1454-2.

https://doi.org/10.1145/3711896.3737129

SILO: Semantic Integration for Location Prediction with Large Language Models

Tianao Sun
School of Software
Shandong University
Jinan, China
tianaosun@mail.sdu.edu.cn

Meng Chen*
School of Software
Shandong University
Jinan, China
mchen@sdu.edu.cn

Bowen Zhang
College of Big Data and Internet
Shenzhen Technology University
Shenzhen, China
zhang_bo_wen@foxmail.com

Genan Dai
College of Big Data and Internet
Shenzhen Technology University
Shenzhen, China
daigenan@sztu.edu.cn

Weiming Huang
School of Geography
University of Leeds
Leeds, UK
W.Huang@leeds.ac.uk

Kai Zhao
AI Lab
Walmart AI
Sunnyvale, USA
kaizhaofrank@gmail.com

Abstract

Next location prediction is a critical task in human mobility modeling, with broad applications in personalized recommendation, urban planning, and location-based services. Recently, researchers have used prompt-based large language models (LLMs) to improve next location prediction with pre-trained knowledge. However, they face inherent challenges in bridging the gap between textual prompts for semantic contextual understanding and human mobility data for transition pattern modeling. In this paper, we introduce **SILO**, a framework designed for **S**emantic **I**ntegration in **LO**cation prediction via LLMs. We first construct a hybrid semantic space that seamlessly integrates ID-based embeddings, text-derived semantics, and auxiliary contextual information, enabling comprehensive modeling of sequential mobility patterns alongside contextual nuances. We then propose user-centric prompts that specify the prediction task for LLMs while embedding user context within a special token. Further, we utilize LLMs as the prediction backbone to process both user-specific prompts and hybrid ID-context embeddings of location sequences. To enhance predictive performance, we finally introduce a dual-logits strategy, combining sequential transition logits with user profile-guided semantic preference logits. Extensive experiments on two large-scale real-world mobility datasets demonstrate that SILO significantly outperforms state-of-the-art baselines, validating its effectiveness in modeling complex mobility patterns through semantic integration using LLMs.

CCS Concepts

• **Information systems → Spatial-temporal systems**; **Data mining**.

*Corresponding Author.

KDD '25, August 3–7, 2025, Toronto, ON, Canada.

ACM ISBN 979-8-4007-1454-2/25/08
https://doi.org/10.1145/3711896.3737129

Keywords

Next Location Prediction; Large Language Models; Human Mobility

ACM Reference Format:
Tianao Sun, Meng Chen, Bowen Zhang, Genan Dai, Weiming Huang, and Kai Zhao. 2025. SILO: Semantic Integration for Location Prediction with Large Language Models. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2 (KDD '25), August 3–7, 2025, Toronto, ON, Canada.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3711896.3737129

KDD Availability Link:
The source code of this paper has been made publicly available at https://doi.org/10.5281/zenodo.15489696.

1 Introduction

Human mobility modeling has become a fundamental research topic due to its wide-ranging applications. Understanding human mobility patterns holds the key to a wide range of applications, from traffic and urban planning [3, 6, 18, 42] to location-based services [2, 14, 16, 37, 43]. Despite the surge of large-scale mobility datasets, accurately predicting the next location remains elusive. This complexity arises from the dynamic interplay of sequential behaviors, evolving user preferences, and contextual factors that are often subtle yet significant.

Traditional deep approaches have attempted to untangle this web through sequential models like Recurrent Neural Networks (RNNs) [22, 41, 44] and Transformer architectures [30, 35, 40], which excel at capturing mobility transitions (see Figure 1a). However, these models often treat user trajectories as mere sequences of discrete IDs, overlooking the rich semantic context embedded in human movements—the very essence that makes a journey meaningful.

The emergence of Large Language Models (LLMs) opens new avenues for bridging this gap. Researchers have experimented with two main paradigms: prompt engineering and embedding integration. The former reframes next location prediction as a text generation task, feeding mobility records as prompts into pre-trained LLMs [1, 8, 17, 34] (see Figure 1b). While innovative, this method struggles to reconcile the inherent differences between natural language and symbolic mobility data. A more refined approach, the embedding integration paradigm (see Figure 1c), attempts to bridge

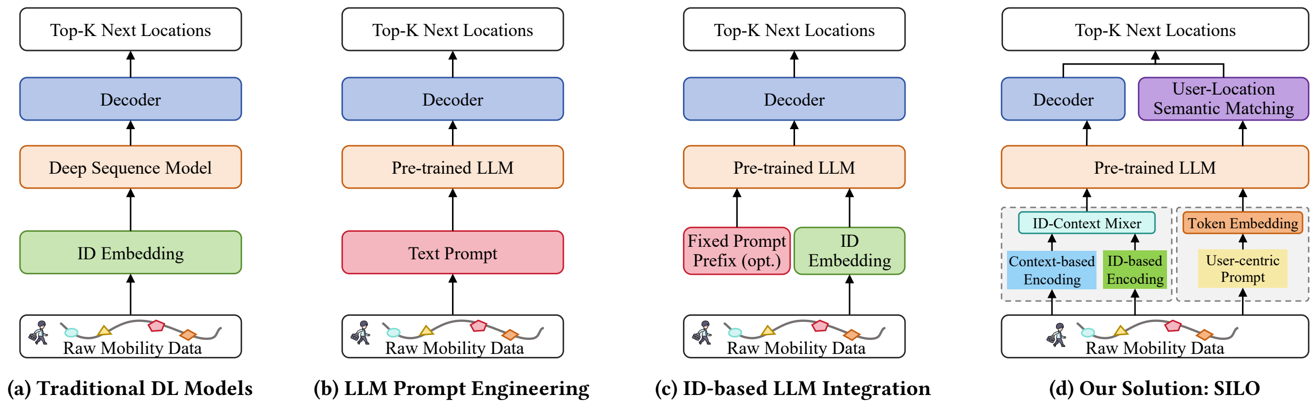


Figure 1: Comparison of various location prediction paradigms.

this gap by incorporating learnable ID-based embeddings while still leveraging LLMs for sequence processing [10, 21, 25]. This approach enhances accuracy, but it still relies on numeric IDs that are randomly initialized and trained on comparatively small datasets. Unlike natural language tokens in LLMs, which acquire rich semantics during large-scale pre-training, these newly introduced IDs are often trained only based on co-occurrence patterns and lack inherent semantics. As a result, such methods still primarily function as sequence learners rather than utilizing LLMs' semantic understanding capabilities.

These limitations underscore a fundamental challenge in LLM-driven mobility modeling: **how to effectively integrate structured mobility transitions with the rich semantics inherent in human movements?** As illustrated above, neither deep sequence models nor existing LLM-based methods can effectively unify sequential mobility patterns with rich semantic reasoning. An ideal approach should not only preserve the structured transition modeling strengths of deep sequence architectures but also harness LLMs' powerful contextual understanding.

In this paper, we introduce **SILO**, a novel framework that harmonizes the structured world of mobility data with the rich semantic landscapes navigated by LLMs (see Figure 1d). SILO creates a hybrid semantic space through the ID-Context Mixer, which combines ID-based embeddings, textual semantics, and additional contextual information. Central to our approach is *user-centric prompts*, which specify the prediction task for LLM and embed user context within a special token. This design enables our model to trace past behaviors and anticipate future actions with contextual awareness. We then utilize an LLM as the prediction backbone to process both user-specific prompts and hybrid ID-context embeddings of activity location sequences. Finally, we propose a dual-logits strategy that merges sequential transition logits (computed via Decoder) with semantic preference logits (computed via User-Location Semantic Matching), allowing SILO to capture both population-level mobility trends and individual behavioral nuances, thus balancing generalization with personalization.

Our main contributions are summarized as follows:

- We propose **SILO**, an LLM-based framework that integrates context-based and ID-based semantics with user-specific behavioral patterns for predicting the next location. This establishes a new paradigm for location prediction using LLMs, bridging the gap between sequence transition modeling and semantic integration.
- We construct a **hybrid semantic space** that combines ID-based embeddings, context-based semantics, and auxiliary contextual information. This fusion enables comprehensive modeling of sequential mobility patterns and rich contextual semantics. Additionally, we introduce **user-centric prompts** for dynamically encoding evolving user preferences into a specialized LLM token, facilitating personalized and context-aware location prediction.
- We design a **dual-logits strategy** that integrates sequential transition logits with user profile-guided semantic preference logits, improving location prediction accuracy by capturing both individual-level nuances and shared behavioral patterns.
- We conduct **comprehensive evaluations** on two real-world mobility datasets, demonstrating that SILO significantly outperforms state-of-the-art baselines, showcasing notable improvements in prediction accuracy. Our codes are available at https://github.com/AIMUrban/SILO.

2 Related Work

In this section, we focus on recent developments in next location prediction methods and the emerging role of LLMs in this domain.

2.1 Next Location Prediction Using Machine Learning

Early approaches to location prediction focused on mining spatial-temporal patterns from human mobility data using statistical models [5, 15, 28] and sequence modeling techniques [4, 9, 19, 20]. With the advent of deep learning, models such as RNNs [7, 39] and Long Short-Term Memory Networks (LSTMs) [23, 29, 33] became widely used for capturing short-term and long-range sequential dependencies in mobility data. In addition to recurrent architectures, Convolutional Neural Networks (CNNs) have also been explored for mobility prediction. Convolutional models [24, 36] mainly capture local spatial-temporal correlations in mobility data, providing an alternative to recurrent structures with improved parallelization and efficiency. More recently, Transformer-based architectures have been introduced, which achieve state-of-the-art prediction performance through adaptive attention mechanisms and multi-context integration [12, 26, 30, 40].

Despite these advancements, most existing methods rely heavily on the intrinsic properties of human mobility data while neglecting external semantic information. Pre-trained knowledge from LLMs, which naturally resides in a different semantic vector space, remains largely underutilized. This gap between the mobility embedding space and the semantic vector space limits the ability of current models to leverage rich contextual semantics, ultimately constraining their accuracy and generalization capabilities for next location prediction.

2.2 Next Location Prediction Using LLMs

Recent work has explored the use of LLMs for next location prediction, taking advantage of their semantic reasoning and contextual processing capabilities. These efforts generally fall into two paradigms: (1) **Prompt Engineering**: Early attempts [8, 34] frame mobility prediction as text completion tasks via designed templates. While demonstrating the feasibility of mapping mobility sequences into a textual format, these zero-shot approaches suffer from the inherent mismatch between natural language inputs and trajectory data, leading to suboptimal performance. (2) **Embedding Integration**: More recent approaches [10, 25] inject learnable ID-based embeddings into LLMs and fine-tune the model partially. While this improves accuracy by exploiting the LLM's sequence-processing capacity, it effectively turns the LLM into a high-capacity RNN/Transformer that still relies heavily on ID tokens rather than rich language understanding and knowledge distillation.

A plausible factor for this limitation is the lack of semantic grounding in mobility-specific identifiers. Unlike natural language tokens whose embeddings are refined on massive corpora, mobility numeric IDs (e.g., loc_11, user_42) are introduced only during fine-tuning on relatively small datasets. Consequently, the model may learn transition probabilities but struggles to capture the behavioral intent behind movements.

In this context, our work seeks to combine the strengths of previous paradigms. ID-based spatiotemporal embeddings excel at modeling transition patterns, whereas context-based representations tap into the LLM's vast pre-trained knowledge encoded in LLMs. To integrate these complementary perspectives effectively, we employ LLMs as the predictive backbone, enabling a natural fusion of multiple semantic sources to enhance next location prediction.

3 Problem Statement

Definition 1 (Trajectory). *Given a user u, a mobility record is represented as a spatial-temporal tuple (l, t), where l denotes the visited location and t is the corresponding timestamp. A sequence of such records over time forms the user's **trajectory** T^u.*

Definition 2 (Activity Sequence). *Given a user trajectory T^u, we extract all significant activity locations where the user stays beyond a threshold duration (e.g., 60 minutes) to form an **activity sequence** S^u. This sequence focuses on user activities rather than intermediate route details.*

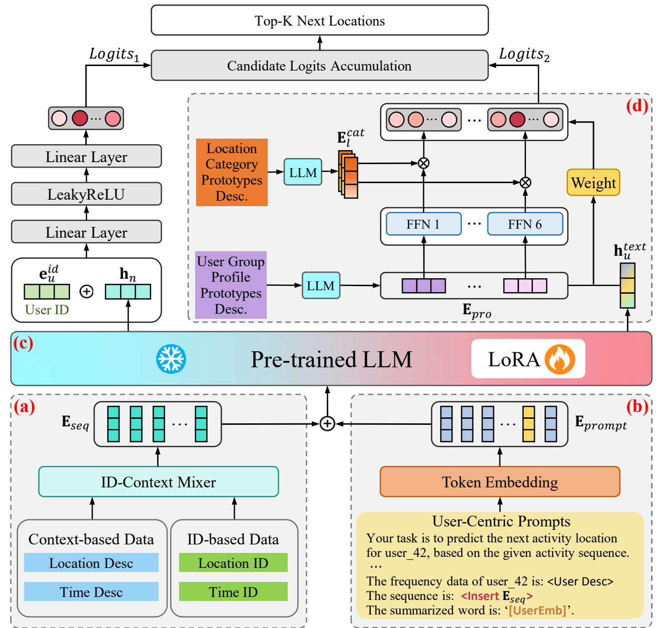


Figure 2: The framework of SILO: (a) ID-Context Mixer for hybrid representation learning, (b) User-Centric Prompts for evolving preferences capturing, (c) LLM-Based Prediction Backbone, and (d) User-Location Semantic Matching.

Definition 3 (Next Location Prediction). *Given a user's activity sequence $S_i^u = \{(l_{n-m+1}, t_{n-m+1}), \cdots, (l_n, t_n)\}$ over a specific time window, the goal is to predict the next activity location l_{n+1} at the upcoming time step. This task is formulated based **solely on sequential mobility data**, i.e., users' spatial-temporal tuples.*

4 Methodology

The framework for SILO is illustrated in Figure 2, which consists of four key components:

(a) **ID-Context Mixer for Hybrid Representation Learning**. We integrate ID-based embeddings and context-based semantics to construct unified representations of location and time. The hybrid representations capture both transition patterns and contextual semantic information.

(b) **User-Centric Prompts for Evolving Preferences Capturing.** We design user-centric prompts to dynamically encode evolving user preferences into a special LLM token (`[UserEmb]`). This mechanism allows the model to capture both long-term behavioral trends and short-term contextual shifts, enhancing its adaptability to user-specific mobility patterns.

(c) **LLM-Based Prediction Backbone.** We employ an LLM as the backbone to process both user-specific prompts and activity sequences. The LLM models sequential transition patterns and evolving user preferences, generating the first set of logits, referred to as *sequential transition logits*.

(d) **User-Location Semantic Matching.** To model user-specific preferences for location semantics, we introduce profile-specific expert modules guided by the semantic representation of evolving user behaviors. This component generates the second set of logits, noted as *semantic preference logits*, capturing high-level shared user mobility preferences.

Finally, the next location prediction is obtained using the dual-logits strategy, which combines sequential transition logits and semantic preference logits. This strategy enables the model to simultaneously consider both individual-level nuances and population-level behavioral patterns, leading to more accurate and context-aware location predictions.

4.1 ID-Context Mixer for Hybrid Representation Learning

We first construct unified representations of location and time by leveraging both context-based and ID-based embeddings, along with pre-defined location category information. On the one hand, ID-based embeddings capture direct spatiotemporal patterns, focusing on transition dynamics. On the other hand, context-based embeddings enrich these representations with semantic insights derived from LLMs. Additionally, we incorporate category-level semantics to capture location-specific nuances grounded in user activity patterns. By fusing these complementary embeddings, our model not only benefits from the explicit identifiers of locations and times but also gains a richer semantic understanding, ultimately leading to more accurate predictions.

4.1.1 Conetxt-based Location and Time Embeddings. For context-based location embeddings, we derive textual descriptions for each location based on historical visit frequencies over different time segments. This design captures subtle user-location interactions from a textual semantic perspective, leveraging the representational power of pre-trained LLMs to enrich the location semantic information.

First, to better capture daily and weekly patterns, we partition each day into four time segments: morning (6:00–12:00), afternoon (12:00–18:00), evening (18:00–24:00), and night (0:00–6:00). This segmentation aligns with typical human activity patterns and mitigates data sparsity. Over a week, time segments are ordered sequentially from "Sunday morning" to "Saturday night", with the time index ranging from 0 to 27. Next, we compute how often each location is visited during these time segments and convert the statistics into a textual description $\mathcal{D}_l$ (see Appendix Section A.1 for implementation details). We then feed it into a frozen LLM and apply mean pooling on the final-layer outputs, followed by a linear transformation, to obtain the semantic embedding $\mathbf{e}_l^{text} \in \mathbb{R}^{d_{llm}}$, where d_{llm} denotes the LLM's default dimension. The process is as follows:

$$\mathbf{H}_l = \textit{Frozen-LLM}(\mathcal{D}_l), \quad \mathbf{e}_l^{text} = \textit{Linear}(\textit{Mean-Pooling}(\mathbf{H}_l)), \quad (1)$$

where $\mathbf{H}_l$ represents the hidden states of a given description $\mathcal{D}_l$ after passing through the frozen LLM. The function $\textit{Linear}(\cdot)$ projects the mean-pooled embedding from the hidden space into the token embedding space.

Meanwhile, we generate a textual-semantic time embedding by combining the day of the week with the time segment (e.g., "Sunday morning") and feed this description $\mathcal{D}_t$ into the frozen LLM to obtain $\mathbf{e}_t^{text} \in \mathbb{R}^{d_{llm}}$,

$$\mathbf{H}_t = \textit{Frozen-LLM}(\mathcal{D}_t), \quad \mathbf{e}_t^{text} = \textit{Linear}(\textit{Mean-Pooling}(\mathbf{H}_t)). \quad (2)$$

4.1.2 ID-based Location and Time Embeddings. While context-based embeddings provide rich semantic insights, we also employ ID-based embeddings to capture discrete identifiers for each location and time. This approach ensures that the model has a direct pattern learning of every unique location or time, which can be useful for representing transitions in sequential data. Formally, we use standard embedding operations,

$$\mathbf{e}_l^{id} = \textit{Embedding}(l), \quad \mathbf{e}_t^{id} = \textit{Embedding}(t), \quad (3)$$

where $\textit{Embedding}(\cdot)$ is a randomly initialized embedding layer. The term $\mathbf{e}_l^{id} \in \mathbb{R}^{d_l}$ and $\mathbf{e}_t^{id} \in \mathbb{R}^{d_t}$ represent the ID-based vectorized embeddings of location and time, respectively, with d_l and d_t denoting their corresponding dimensions.

4.1.3 ID-Context Mixer. We concatenate the ID-based embeddings and context-based embeddings and pass them through a dense layer to enhance their combined representation. This process is formalized as:

$$\begin{aligned} \mathbf{e}_l &= \text{LeakyReLU}(\mathbf{W}_l \cdot (\mathbf{e}_l^{id} \oplus \mathbf{e}_l^{text}) + b_l), \\ \mathbf{e}_t &= \text{LeakyReLU}(\mathbf{W}_t \cdot (\mathbf{e}_t^{id} \oplus \mathbf{e}_t^{text}) + b_t), \end{aligned} \quad (4)$$

where $\oplus$ denotes the concatenation operation. $\mathbf{W}_l \in \mathbb{R}^{d_{llm} \times (d_l + d_{llm})}$, $\mathbf{W}_t \in \mathbb{R}^{d_{llm} \times (d_t + d_{llm})}$ and $b_l, b_t \in \mathbb{R}^{d_{llm}}$ represent learnable parameters, and LeakyReLU is the activation function.

4.1.4 Location Category Prototypes Addition. Recall that we constructed a context-based semantic embedding $\mathbf{e}_l^{text}$ for locations based on the textual description $\mathcal{D}_l$ in Section 4.1.1. However, different locations hold varying meanings depending on the user and the temporal context. For example, a location may represent a "Work" area for one group of users during weekday mornings, as it includes office spaces they frequently visit. In contrast, the same location could represent an "Entertainment" zone for another group of users in the evening, as it might also contain popular restaurants or theaters. Failure to incorporate this variability limits the model's ability to adapt to real-world user activity patterns.

To account for this, we introduce auxiliary location category embeddings that reflect user activity semantics and integrate them with the original location embeddings. These auxiliary embeddings allow the model to dynamically enrich the location representations, adjusting for differences in how various user groups perceive and interact with a given location over time. Based on prior work [25] and domain knowledge, we first construct a description pool with M_c location prototypical categories from the perspective of temporal activity patterns. We then use a frozen LLM to extract the semantic vectors $\mathbf{E}_{cat} \in \mathbb{R}^{M_c \times d_{llm}}$ for each category. The entire process is formalized as follows:

$$\mathbf{H}_{cat} = \textit{Frozen-LLM}(\mathcal{D}_{cat}), \quad \mathbf{E}_{cat} = \textit{Mean-Pooling}(\mathbf{H}_{cat}), \quad (5)$$

where $\mathcal{D}_{cat}$ is the category name and its description. The detailed procedure for this operation is provided in Appendix Section A.2.

To dynamically assign category semantics to each location, we compute the correlation between location embeddings and category embeddings using a cross-attention mechanism,

$$\begin{aligned} \mathbf{Q} &= (\mathbf{e}_l)^{\mathrm{T}} \cdot \mathbf{W}_q + b_q, \\ \mathbf{K} &= \mathbf{E}_{cat} \cdot \mathbf{W}_k + b_k, \\ \mathbf{V} &= \mathbf{E}_{cat} \cdot \mathbf{W}_v + b_v, \\ \mathbf{e}_l^{cat} &= softmax(\mathbf{Q}\mathbf{K}^{\mathrm{T}}/\sqrt{d})\mathbf{V}, \end{aligned} \tag{6}$$

where $\mathbf{W}_q$, $\mathbf{W}_v$, and $\mathbf{W}_k \in \mathbb{R}^{d_{llm} \times d_{llm}}$ are learnable projection matrices. b_q, b_k, and $b_v \in \mathbb{R}^{1 \times d_{llm}}$ are bias terms.

In the end, we add location category embeddings to the initial location embeddings. The unified representation of location and time for a mobility record, denoted as $\mathbf{e}_{mr}$, is computed as follows:

$$\mathbf{e}_l = \mathbf{e}_l + e_l^{cat}, \quad \mathbf{e}_{mr} = LeakyReLU(\mathbf{W} \cdot (\mathbf{e}_l \oplus \mathbf{e}_t) + b), \tag{7}$$

where $\mathbf{W} \in \mathbb{R}^{d_{llm} \times 2 \cdot d_{llm}}$ and $b \in \mathbb{R}^{d_{llm}}$ are learnable parameters. Given an activity sequence, we concatenate each mobility record, $\mathbf{e}_{mr}$, into a single matrix $\mathbf{E}_{seq} \in \mathbb{R}^{(n-m+1) \times d_{llm}}$.

4.2 User-Centric Prompts for Evolving Preferences Capturing

Intuitively, users usually exhibit diverse and evolving preferences that a single ID-based user embedding may fail to capture. To address this limitation, we design a user-centric prompt that not only specifies the prediction task for LLM but also embeds a user's context-dependent information into a special token. Since LLM effectively integrates and prioritizes relevant context, this token encodes each individual's historical activities and contextual cues, thus representing a dynamic user preference. By leveraging this design, LLM can produce more accurate predictions that consider evolving user preferences.

4.2.1 Prompt Design. Traditional prompts in prior works are typically static, serving only to specify the prediction task and remaining unchanged across different sequences, which may fail to account for the evolving nature of user behavior and contextual variations. We advance this concept by proposing a user-centric prompt tailored to reflect an individual's historical habits and contextual information. Our prompt guides the model to dynamically summarize a user's historical behaviors (i.e., <User Desc>) and contextual information from the sequence to be predicted, into a special token (i.e., `[UserEmb]`). This token represents both long-term user preferences and short-term sequence-specific semantics, enabling the model to capture evolving user preferences across different sequences.

For example, for a user with ID 42, the prompt template is:

> **Prompt:** Your task is to predict the next activity location for user_42, based on the given activity sequence.
> Additional requirements:
> 1. Incorporate the frequency data of this user's historical activity times, and summarize it into one word along with the following sequence.
> The frequency data of user_42 is: {<User Desc>}.
> The sequence is: <Insert $\mathbf{E}_{seq}$ >
> The summarized word is: '`[UserEmb]`'.

Here, <Insert> is a placeholder for inserting the activity sequence embedding $\mathbf{E}_{seq}$. We likewise use a frozen LLM to encode each user description <User Desc> and refine it with a dense layer. An illustrative <User Desc> is shown as follows:

> 'Sunday morning': 0.002,
> 'Sunday afternoon': 0.014,
> ...
> 'Saturday night': 0.007

The above user description outlines how frequently the user appears within the historical data. When combined with the sequence to be predicted, such historical information provides valuable insights into the user's potential activity patterns, effectively capturing a dynamic user preference.

4.2.2 Prompt Encoding. Once the prompt is finalized, it is tokenized, resulting in token embeddings, which serve as one part of the input to the LLM,

$$\mathbf{E}_{prompt} = Token\text{-}Embedding(Tokenize(\text{Prompt})), \tag{8}$$

where $Token\text{-}Embedding(\cdot)$ represents the pre-trained token embedding layer of LLM, and $Tokenize(\cdot)$ denotes the tokenization process. In the token embedding space, the special token `[UserEmb]` is highlighted in yellow in Figure 2.

4.3 LLM-Based Prediction Backbone

We illustrate how to form the input of the LLM, and how to extract key hidden states and obtain the sequential transition logits.

4.3.1 LLM's Input Embeddings. The LLM input is prepared by inserting the sequence embedding $\mathbf{E}_{seq}$ into the user prompt embedding $\mathbf{E}_{prompt}$ at the designated position <Insert>, as shown in the prompt template. The LLM processes this input to generate $\mathbf{H}_{final}$, the final layer's hidden states, which have the same shape as the input. We use GPT [27] as the backbone LLM and apply Low-Rank Adaptation (LoRA) [13], which introduces lightweight trainable matrices to enable effective fine-tuning while keeping most of the original parameters frozen.

4.3.2 Extraction in Hidden States. Due to the LLM's attention mechanisms and deep contextual encoding capabilities, the hidden states provide a rich representation of both the input prompt and the activity sequence. We next extract the hidden states corresponding to the special token `[UserEmb]` and the last position of the sequence from $\mathbf{H}_{final}$. These states encode both the user's dynamic preferences and the sequential patterns. Specifically, the state at `[UserEmb]` is denoted by $\mathbf{h}_u^{text} \in \mathbb{R}^{d_{llm}}$, encapsulating the user's evolving preference based on the prompt context (e.g., <User Desc>) and the sequence. Meanwhile, the last position of the sequence in hidden state $\mathbf{H}_{final}$, denoted as $\mathbf{h}_n$, captures sequential patterns.

4.3.3 Logits based on Sequential Transitions. To capture the cumulative influence of sequential patterns, we combine $\mathbf{h}_n$ with the ID-based user embedding $\mathbf{e}_u^{id} \in \mathbb{R}^{d_u}$, and feed the concatenation into an MLP decoder to generate the next location logits:

$$Logits_1 = MLP(\mathbf{h}_n \oplus \mathbf{e}_u^{id}), \tag{9}$$

where $Logits_1 \in \mathbb{R}^{|\mathcal{P}|}$ refers to the prediction scores over all candidate locations $\mathcal{P}$. The MLP consists of two linear layers with LeakyReLU activation in between.

4.4 User-Location Semantic Matching

After undergoing deep encoding and attention mechanisms in the LLM, the special token [UserEmb] encapsulates both the historical information embedded in <User Desc> and the transition patterns of the sequences. This token serves as a dynamic representation of a user's evolving preference. However, its high specificity to individual users and sequences makes direct location prediction challenging. To address this, we align [UserEmb] with profile-specific experts, grouping users into broader behavioral patterns. This abstraction facilitates prediction by capturing shared mobility trends while maintaining user- and sequence-specific nuances, ultimately enhancing both generalization and prediction accuracy.

4.4.1 User Group Profile Prototypes Construction and Representation. To capture diverse user mobility patterns, we define M_u user group profiles based on temporal behavioral patterns [10, 31] and expert-defined categorizations. These profiles capture broad user mobility characteristics, such as "Early Birds, who are active in the morning on weekdays, shifting to the afternoon on weekends.", and "Night Owls, who are active at night—whether for work, study, or leisure—across both weekdays and weekends.". By incorporating these profiles, the model aligns user activities with time-sensitive preferences (e.g., weekdays vs. weekends), improving generalization while preserving individual-level nuances via [UserEmb]. Following a similar process as in Section 4.1.4, we construct a profile description pool and extract the semantic vector for each user group profile using a frozen LLM, denoted as $\mathbf{E}_{pro} \in \mathbb{R}^{M_u \times d_{llm}}$. Formally,

$$\mathbf{H}_{pro} = \textit{Frozen-LLM}(\mathcal{D}_{pro}),\ \mathbf{E}_{pro} = \textit{Mean-Pooling}(\mathbf{H}_{pro}), \quad (10)$$

where $\mathcal{D}_{pro}$ is the profile name along with its description, and a dense layer is applied to refine $\mathbf{E}_{pro}$ before the gating network. The detailed procedure is provided in Appendix Section A.3.

4.4.2 User-Specific Group Profile Learning. To adaptively assign profile features to each user, we introduce a gating mechanism that dynamically adjusts contributions from profile-specific experts. Each expert is trained to specialize in modeling preferences for location categories under a specific profile. By weighting these experts based on a user's semantic vector $\mathbf{h}_u^{text}$, the model tailors predictions to both user- and sequence-specific contexts.

Since profile vectors and $\mathbf{h}_u^{text}$ reside in the same embedding space (both derived from LLM's hidden states), we compute profile-expert weights using a cosine similarity-based gating network:

$$g(\mathbf{h}_u^{text}, \mathbf{E}_{pro}) = \textit{softmax}\left(\frac{\mathbf{h}_u^{text} \cdot (\mathbf{E}_{pro})^{\mathrm{T}}}{\|\mathbf{h}_u^{text}\| \cdot \|\mathbf{E}_{pro}\|}\right), \quad (11)$$

where the output $g(\mathbf{h}_u^{text}, \mathbf{E}_{pro}) \in \mathbb{R}^{M_u}$ is used to indicate each profile expert's relative importance.

4.4.3 Logits based on User-Location Matching. Each expert $f(\cdot)$ is implemented as a feedforward network (FFN) with its own set of learnable parameters, enabling it to focus on the preferences for location category information $\mathbf{E}_l^{cat}$ within its respective user group. The logits for location prediction using the user-location matching module are computed as:

$$Logits_2 = g(\mathbf{h}_u^{text}, \mathbf{E}_{pro}) \sum_{m_u=1}^{M_u} f(\mathbf{E}_{pro_{m_u}})(\mathbf{E}_l^{cat})^{\mathrm{T}}, \quad (12)$$

where $Logits_2 \in \mathbb{R}^{|\mathcal{P}|}$, and for the entire location set $\mathcal{P}$, the location category embeddings $\mathbf{e}_l^{cat}$ are aggregated into $\mathbf{E}_l^{cat} \in \mathbb{R}^{|\mathcal{P}| \times d_{llm}}$.

4.5 Training Objective

The task of next location prediction is formulated as a classification problem over the entire location set $\mathcal{P}$, leveraging a dual-logits strategy that integrates both $Logits_1$ and $Logits_2$ for enhanced predictive performance. For a given user u and their activity sequence S_i^u, we compute the probability distribution over all possible locations for the next activity location l_{n+1}. The predicted probability for each location j is computed as:

$$P(\hat{l}_{n+1})_j = \textit{softmax}(Logits_1 + Logits_2)_j, \quad (13)$$

where $Logits_1$ captures sequential patterns in the user's activity history (Section 4.3.3) and $Logits_2$ models the user-specific shared preferences for location categories (Section 4.4.3).

The training objective is to minimize the multi-class cross-entropy loss, defined as:

$$\mathcal{L}_{S_i^u} = -\sum_{j=1}^{|\mathcal{P}|} P(l_{n+1})_j \log P(\hat{l}_{n+1})_j, \quad (14)$$

where $P(l_{n+1})_j$ represents the ground truth in one-hot encoding, with $P(l_{n+1})_j = 1$ if the next activity location corresponds to the j-th location. Meanwhile, $P(\hat{l}_{n+1})_j$ denotes the model's predicted probability for activity location j.

5 Experiments

5.1 Experiment Setup

5.1.1 Datasets. We utilize public human mobility datasets from two metropolitan areas [38], referred to as Metropolitan A (10,000 users, 22,383 activity locations and 2,320,997 records) and Metropolitan B (8,000 users, 13,869 activity locations and 1,436,346 records). The datasets have been gridded and anonymized to ensure privacy. We apply a temporal split, using the first 80% of the data (e.g., the first 48 days in a 60-day dataset) for training and the remaining 20% (e.g., the last 12 days) for testing.

5.1.2 Baselines. We compare SILO with the following baselines:

- **Statistical Methods:** 1-MMC [9], FPMC [28].
- **Deep Learning Methods:** DeepMove [7], Flashback [39], GETNext [40], C-MHSA [12], CSLSL [31], MCLP [30].
- **LLM-Based Methods:** LLM-Mob [34], Mobility-LLM [10], NextLocLLM [25].

Detailed descriptions of these methods are available in Appendix Section A.4.

5.1.3 Evaluation Metrics. We employ the following metrics to assess the performance of various methods: (1) **Accuracy (Acc@K)** measures the proportion of times that the correct next location appears within top-K predicted candidates. (2) **Mean Reciprocal Rank (MRR)** computes the average reciprocal of the rank at which the correct location is found among the predicted candidates.

5.1.4 Model Settings. We train the model for 15 epochs using the Adam optimizer, with a learning rate of $5e^{-4}$ and an L2 penalty of $1e^{-5}$. We determine model parameters via a grid search with an adaptive step size strategy. The embedding dimensions are set

Table 1: The performance of all methods for next location prediction, where the best and second performing results are represented in bold and <u>underlined</u>, respectively.

Methods	Metropolitan A					Metropolitan B				
	Acc@1	Acc@3	Acc@5	Acc@10	MRR	Acc@1	Acc@3	Acc@5	Acc@10	MRR
1-MMC	15.16	26.68	31.75	38.32	22.68	16.31	28.49	34.49	42.59	24.60
FPMC	17.35	32.77	39.34	47.64	27.81	18.48	34.34	41.29	50.18	29.33
DeepMove	18.48	35.09	42.34	50.89	29.56	20.28	36.78	43.43	51.92	31.31
Flashback	18.26	36.09	44.33	54.21	30.33	19.51	37.15	45.03	55.02	31.49
GETNext	19.91	37.18	44.68	54.52	31.68	20.69	39.08	47.03	56.92	33.03
C-MHSA	19.73	37.80	45.68	55.10	31.74	20.81	39.14	47.42	57.84	33.25
CSLSL	21.28	39.10	45.42	51.54	32.14	22.65	41.45	48.88	57.80	34.86
MCLP	20.58	39.65	47.74	57.32	33.06	21.84	41.41	49.76	59.78	34.75
LLM-Mob	20.54	39.11	45.64	52.64	-	21.73	41.38	48.56	56.26	-
Mobility-LLM	20.87	40.49	<u>49.22</u>	<u>59.28</u>	33.78	22.93	42.91	51.02	60.36	35.79
NextLocLLM	<u>22.25</u>	<u>40.97</u>	48.12	56.42	<u>34.23</u>	<u>23.06</u>	<u>43.01</u>	<u>51.09</u>	<u>60.51</u>	<u>36.04</u>
SILO	**23.79**	**43.86**	**51.71**	**60.78**	**36.58**	**24.85**	**45.75**	**54.09**	**63.62**	**38.22**

as follows: $d_l = 128$, $d_t = 128$, and $d_u = 768$, while the remaining dimensions follow the default LLM hidden size configuration.

5.2 Performance Comparison

Table 1 presents the overall performance of SILO compared to baseline methods on the two metropolitan datasets. Each model runs five times, and the mean performance metrics are reported.

- Traditional statistical models (1-MMC and FPMC) exhibit significantly lower performance than deep learning-based approaches, as they struggle to capture complex mobility patterns, highlighting the limitations of Markovian assumptions in modeling long-range dependencies.
- Deep sequence models (DeepMove, Flashback, GETNext, C-MHSA, CSLSL, and MLCP) achieve competitive results but remain inferior to LLM-based methods. These models effectively capture long-term sequential dependencies via recurrent or attention-based architectures. However, they rely primarily on ID-based transition semantics, making it difficult to integrate external context-based semantic knowledge. As a result, their improvements remain incremental. Notably, MCLP performs better than other deep sequence models in mid-to-high-range predictions (Acc@5 and Acc@10), likely due to its multi-context learning mechanism, which captures user preferences and temporal regularities, thereby refining ranked predictions.
- LLM-based approaches (LLM-Mob, Mobility-LLM, and NextLocLLM) exhibit stronger generalization but remain constrained by ID-based representations. LLM-Mob highlights the potential of pre-trained LLMs in mobility prediction, but its zero-shot setting leads to weaker results. Moreover, LLM-Mob cannot inherently rank all candidate locations, since it formulates next location prediction as a text generation task, limiting its effectiveness in scenarios where a complete ranked prediction list is required. NextLocLLM and Mobility-LLM improve performance by incorporating learnable ID-based embeddings, yet they still suffer from a lack of alignment between mobility sequences and LLMs' textual context understanding capabilities, limiting their ability to fully leverage external knowledge.
- SILO achieves the best performance among all, consistently outperforming all baselines across all evaluation metrics. Specifically, SILO improves Acc@1 by 6.92% and MRR by 6.42% over the best baseline on Metropolitan A, and by 7.76% and 6.86%, respectively, on Metropolitan B. The paired t-test results confirm that these improvements are statistically significant, with $p < 0.01$, demonstrating the effectiveness of SILO's bi-semantic modeling. By integrating ID-based and context-based semantics, SILO enhances LLM-driven mobility prediction through hybrid semantic integration, user-centric prompts, and the dual-logits strategy, enabling more effective reasoning over both mobility transitions and contextual semantics.

Table 2: Ablation study on the Metropolitan B data.

Variant	Acc@1	Acc@5	Acc@10	MRR
w/o Context-based	23.08	51.99	61.93	36.38
w/o ID-based	21.78	46.13	53.60	33.09
w/o $Logits_2$	23.08	49.86	58.79	35.54
w/o Prompt	23.70	52.96	62.25	37.02
w/o LLM	22.12	49.47	56.49	34.33
SILO	24.85	54.09	63.62	38.22

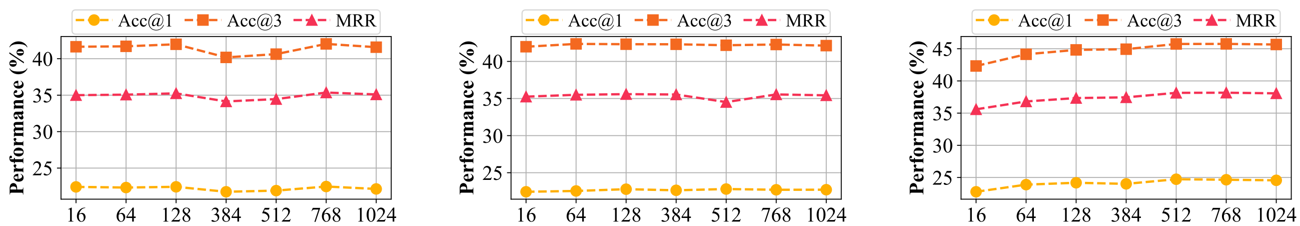
5.3 Ablation Study

To evaluate the contribution of each component to the overall performance, we conduct an ablation study (see Table 2) by testing the following model variants:

- ***w/o* Context-based**: We remove context-based semantics, relying only on ID-based embeddings. The results indicate that

Table 3: Comparative analysis of different LLMs on the Metropolitan B data.

Model	Epoch Time (s)	GPU Mem. (GB)	Best Epoch	Acc@1	Acc@3	Acc@5	Acc@10	MRR
GPT-2-124M	~ 89	6.94	15	24.85	45.75	54.09	63.62	38.22
Llama 3.2-1B	~ 278	14.26	7	22.90	43.64	51.99	61.34	36.14
Phi-2-2.7B	~ 328	15.97	8	23.39	44.23	52.76	62.34	36.74
Llama 3.2-3B	~ 469	17.51	6	22.67	43.97	52.52	61.64	36.08
Llama 2-7B	~ 632	22.64	5	22.69	43.73	52.23	61.25	35.94



(a) Location Dimension **(b) Time Dimension** **(c) User Dimension**

Figure 3: Effect of the ID-based embedding dimensions.

textual semantics provide valuable contextual knowledge for mobility modeling when using LLMs.

- ***w/o* ID-based**: We remove ID-based embeddings, relying solely on context-based representations. The performance degradation suggests that ID-based embeddings play a crucial role in capturing mobility transitions, as they inherently encode structured movement patterns.
- ***w/o*** $Logits_2$: We remove the module of user-location semantic matching, using only $Logits_1$ for prediction. The results confirm that removing this module, which captures shared mobility trends, primarily degrades mid-to-high-range accuracy (Acc@5 and Acc@10), while having a smaller impact on Acc@1.
- ***w/o* Prompt**: We eliminate the user-centric prompt module, replacing `[UserEmb]` with standard mean pooling over the input sequence in hidden states. The results indicate that removing prompts weakens the model's ability to incorporate historical user behaviors with activity sequences through evolving user preferences. Notably, profile experts still provide reasonable improvements, but their full potential is not realized without user-centric prompts.
- ***w/o* LLM**: We substitute the LLM-based prediction backbone with a standard Transformer architecture. The results reveal that deep sequence models remain insufficient to fully incorporate the rich semantic knowledge provided by LLMs, as they do not naturally share a vector space between sequential transitions and semantic embeddings. This highlights the unique role of LLMs in bridging structured mobility representations with external semantic information.

5.4 Comparison of Different LLMs

To explore the impact of different LLM backbones on next location prediction, we compare multiple open-source LLMs under the same experimental setup. We apply LoRA to enhance training efficiency. All experiments are conducted on AMD EPYC 7742 64-core Processor CPUs and NVIDIA A100 GPUs. We evaluate GPT-2-124M [27], Llama 3.2-1B, Phi-2-2.7B [11], Llama 3.2-3B, and Llama 2-7B [32]. To ensure stable convergence and prevent overfitting, we employ an early stopping mechanism.

Our evaluation considers both predictive performance and computational efficiency, such as training time per epoch, and GPU memory consumption, as shown in Table 3. The results reveal a diminishing return in performance improvement as model size increases, leading to a clear trade-off between model complexity and computational efficiency. Larger models (Llama 3.2-3B and Llama 2-7B) require more memory and longer training times provide only marginal gains. The second-best Phi-2 requires nearly 3.7× the training time per epoch compared to GPT-2, yet fails to surpass it in key metrics like Acc@1 and MRR.

Conversely, GPT-2 achieves the best efficiency-performance balance, converging in only 15 epochs with minimal GPU memory (6.94GB) while attaining the highest Acc@1 (24.85) and MRR (38.22). Based on these findings, we adopt GPT-2 as the backbone model for SILO, as it offers the most practical balance between accuracy, efficiency, and resource constraints.

5.5 Parameter Sensitivity Analysis

Because context-based information inherits the LLM's dimension, we focus on analyzing the sensitivity of ID-based embedding dimensions: location (d_l), time (d_t), and user (d_u). Figure 3 shows that performance remains stable across different location embedding dimensions, with $d_l = 128$ providing a slight advantage by balancing expressiveness and efficiency. Time embedding performance is also stable; we select $d_t = 128$ for its competitive results and computational efficiency. Higher-dimensional user embeddings consistently improve performance, indicating that complex user mobility patterns benefit from richer representations. Therefore, we choose $d_u = 768$ for improved personalization.

5.6 Few-shot Study

We follow the protocol in Mobility-LLM by training on only the first 5% of the Metropolitan B training set to simulate a few-shot scenario. As reported in Table 4, SILO consistently outperforms both DL- and LLM-based baselines under limited supervision.

Table 4: Few-shot performance on the Metropolitan B data.

Method	Acc@1	Acc@5	Acc@10	MRR
GETNext	6.66	18.53	26.97	13.35
CSLSL	8.67	17.17	21.62	13.23
MCLP	8.74	20.02	26.28	14.66
Mobility-LLM	12.27	21.45	23.33	16.67
NextLocLLM	6.87	13.68	17.66	10.75
SILO	13.94	24.38	25.37	19.05

These results indicate that, compared to approaches that rely on randomly initialized symbolic IDs, integrating textual semantics grounded in pre-trained knowledge enables SILO to leverage the semantic understanding capabilities of LLMs, thereby enhancing prediction accuracy under limited data conditions.

5.7 Showcases of User-Location Semantic Mapping

To better understand how SILO leverages expert specialization for personalized prediction, we conduct two case studies. Figure 4a reveals that users are dynamically assigned to different combinations of experts, reflecting the model's ability to adapt to diverse individual mobility patterns. Some users rely heavily on a single expert, while others show more distributed weights, indicating mixed behaviors.

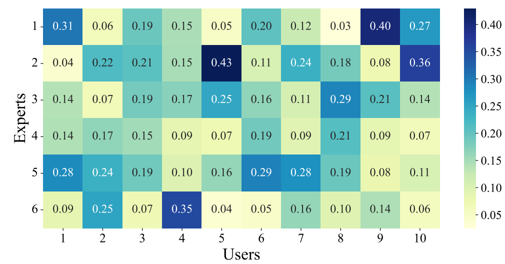
Moreover, Figure 4b analyzes the location category attention of two experts over the top-5 predicted locations for a representative user. The results show that each expert develops distinct semantic preferences, some focusing on daily routine categories, others on leisure-related ones, demonstrating their specialization in modeling different aspects of human mobility.

These observations confirm that SILO effectively leverages expert specialization not only to personalize predictions but also to generalize across common mobility trends shared among user groups.

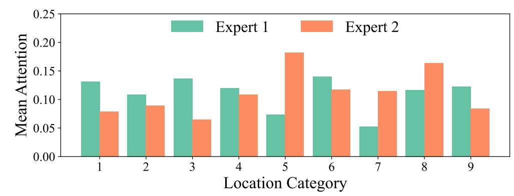
6 Conclusion

In this paper, we propose SILO, a novel framework for next location prediction that leverages Large Language Models to integrate multiple levels of semantic information. We first construct a hybrid semantic space by combining ID-based and context-based embeddings with auxiliary contextual information, enabling it to jointly capture sequential mobility transitions and high-level semantic nuances. We then employ user-centric prompts, embedding user context within a special token, to specify the prediction task for large language models. Using LLMs as the prediction backbone, we process these user-specific prompts alongside hybrid ID-context embeddings of location sequences. Finally, we improve prediction accuracy through a dual-logits strategy that combines sequential transition logits with semantic preference logits. Extensive experiments on real-world mobility datasets demonstrate that SILO consistently outperforms state-of-the-art methods, highlighting the advantages of semantic integration in mobility prediction.

Our findings suggest that incorporating structured mobility patterns and contextual semantics can improve next location prediction, providing a new perspective on leveraging LLMs for spatiotemporal modeling. Future work may explore more efficient adaptation of LLMs to mobility tasks, as well as strategies to enhance personalization and interpretability in user trajectory modeling.



(a) Experts gating weight distribution for ten random users.



(b) Category attentions of two experts on a user's top-5 prediction.

Figure 4: Visualization of expert routing and semantic preferences in user-location mapping.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant No. 61906107, the Key Scientific and Technological Innovation Project of Shandong Province under Grant No. 2024CXGC010113 and 2024CXG010213, and the Young Scholars Program of Shandong University.

References

[1] Ciro Beneduce, Bruno Lepri, and Massimiliano Luca. 2024. Large Language Models are Zero-Shot Next Location Predictors. *arXiv preprint arXiv:2405.20962* (2024).

[2] Junxiang Bing, Meng Chen, Min Yang, Weiming Huang, Yongshun Gong, and Liqiang Nie. 2022. Pre-trained Semantic Embeddings for POI Categories based on Multiple Contexts. *IEEE Transactions on Knowledge and Data Engineering* 35, 9 (2022), 8893–8904.

[3] Meng Chen, Zechen Li, Weiming Huang, Yongshun Gong, and Yilong Yin. 2024. Profiling Urban Streets: A Semi-supervised Prediction Model based on Street View Imagery and Spatial Topology. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 319–328.

[4] Meng Chen, Qingjie Liu, Weiming Huang, Teng Zhang, Yixuan Zuo, and Xiaohui Yu. 2021. Origin-aware Location Prediction based on Historical Vehicle Trajectories. *ACM Transactions on Intelligent Systems and Technology* 13, 1 (2021), 1–18.

[5] Meng Chen, Yan Zhao, Yang Liu, Xiaohui Yu, and Kai Zheng. 2020. Modeling Spatial Trajectories with Attribute Representation Learning. *IEEE Transactions on Knowledge and Data Engineering* 34, 4 (2020), 1902–1914.

[6] Yile Chen, Weiming Huang, Kaiqi Zhao, Yue Jiang, and Gao Cong. 2025. Self-supervised Representation Learning for Geospatial Objects: A Survey. *Information Fusion* (2025), 103265.

[7] 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 2018 World Wide Web Conference.* 1459–1468.

[8] Shanshan Feng, Haoming Lyu, Fan Li, Zhu Sun, and Caishun Chen. 2024. Where to Move Next: Zero-Shot Generalization of LLMs for Next POI Recommendation. In *IEEE Conference on Artificial Intelligence.* 1530–1535.

[9] 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.

[10] Letian Gong, Yan Lin, Xinyue Zhang, Yiwen Lu, Xuedi Han, Yichen Liu, Shengnan Guo, Youfang Lin, and Huaiyu Wan. 2024. Mobility-LLM: Learning Visiting Intentions and Travel Preference from Human Mobility Data with Large Language Models. In *The 38th Annual Conference on Neural Information Processing Systems.*

[11] Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, et al. 2023. Textbooks are All You Need. *arXiv preprint arXiv:2306.11644* (2023).

[12] Ye Hong, Yatao Zhang, Konrad Schindler, and Martin Raubal. 2023. Context-aware Multi-head Self-attentional Neural Network Model for Next Location Prediction. *Transportation Research Part C: Emerging Technologies* 156 (2023), 104315.

[13] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations.*

[14] Hongwei Jia, Meng Chen, Weiming Huang, Kai Zhao, and Yongshun Gong. 2024. Learning Hierarchy-enhanced POI Category Representations using Disentangled Mobility Sequences. In *Proceedings of the 33rd International Joint Conference on Artificial Intelligence.* 2090–2098.

[15] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix Factorization Techniques for Recommender Systems. In *Computer* (2009), 30–37.

[16] Xiaoliang Lei, Hao Mei, Bin Shi, and Hua Wei. 2022. Modeling Network-Level Traffic Flow Transitions on Sparse Data. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining.* 835–845.

[17] Peibo Li, Maarten de Rijke, Hao Xue, Shuang Ao, Yang Song, and Flora D Salim. 2024. Large Language Models for Next Point-of-Interest Recommendation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval.* 1463–1472.

[18] Zechen Li, Weiming Huang, Kai Zhao, Min Yang, Yongshun Gong, and Meng Chen. 2024. Urban Region Embedding via Multi-view Contrastive Prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 8724–8732.

[19] Defu Lian, Cong Zhao, Xing Xie, Guangzhong Sun, Enhong Chen, and Yong Rui. 2014. GeoMF: Joint Geographical Modeling and Matrix Factorization for Point-of-Interest Recommendation. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* 831–840.

[20] Dawen Liang, Laurent Charlin, James McInerney, and David M Blei. 2016. Modeling User Exposure in Recommendation. In *Proceedings of the 25th International Conference on World Wide Web.* 951–961.

[21] Jiayi Liao, Sihang Li, Zhengyi Yang, Jiancan Wu, Yancheng Yuan, Xiang Wang, and Xiangnan He. 2024. Llara: Large language-Recommendation Assistant. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval.* 1785–1795.

[22] Nicholas Lim, Bryan Hooi, See-Kiong Ng, Yong Liang Goh, Renrong Weng, and Rui Tan. 2022. Hierarchical Multi-task Graph Recurrent Network for Next POI Recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval.* 1133–1143.

[23] 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*, Vol. 35. 4241–4248.

[24] Chi Harold Liu, Yu Wang, Chengzhe Piao, Zipeng Dai, Ye Yuan, Guoren Wang, and Dapeng Wu. 2020. Time-aware Location Prediction by Convolutional Area-of-Interest Modeling and Memory-Augmented Attentive LSTM. *IEEE Transactions on Knowledge and Data Engineering* 34, 5 (2020), 2472–2484.

[25] Shuai Liu, Ning Cao, Yile Chen, Yue Jiang, and Gao Cong. 2024. NextLocLLM: Next Location Prediction Using LLMs. *arXiv preprint arXiv:2410.09129* (2024).

[26] Yanjun Qin, Yuchen Fang, Haiyong Luo, Fang Zhao, and Chenxing Wang. 2022. Next Point-of-Interest Recommendation with Auto-correlation Enhanced Multi-modal Transformer Network. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval.* 2612–2616.

[27] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language Models are Unsupervised Multitask Learners. *OpenAI Blog* 1, 8 (2019), 9.

[28] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing Personalized Markov Chains for Next-basket Recommendation. In *Proceedings of the 19th International Conference on World Wide Web.* 811–820.

[29] Ke Sun, Tieyun Qian, Tong Chen, Yile Liang, Quoc Viet Hung Nguyen, and Hongzhi Yin. 2020. Where to Go Next: Modeling Long-and Short-term User Preferences for Point-of-Interest Recommendation. In *Proceedings of the AAAI conference on Artificial Intelligence*, Vol. 34. 214–221.

[30] Tianao Sun, Ke Fu, Weiming Huang, Kai Zhao, Yongshun Gong, and Meng Chen. 2024. Going Where, by Whom, and at What Time: Next Location Prediction Considering User Preference and Temporal Regularity. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining.* 2784–2793.

[31] Tianao Sun, Kai Zhao, and Meng Chen. 2024. Human-AI Interaction: Human Behavior Routineness Shapes AI Performance. *IEEE Transactions on Knowledge and Data Engineering* 36, 12 (2024), 8476–8487.

[32] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open Foundation and Fine-tuned Chat Models. *arXiv preprint arXiv:2307.09288* (2023).

[33] Chen Wang, Mengting Yuan, Yang Yang, Kai Peng, and Hongbo Jiang. 2024. Revisiting Long-and Short-term Preference Learning for Next POI Recommendation with Hierarchical LSTM. *IEEE Transactions on Mobile Computing* (2024).

[34] Xinglei Wang, Meng Fang, Zichao Zeng, and Tao Cheng. 2023. Where Would I Go Next? Large Language Models as Human Mobility Predictors. *arXiv preprint arXiv:2308.15197* (2023).

[35] Jiayi Xie and Zhenzhong Chen. 2023. Hierarchical Transformer with Spatio-temporal Context Aggregation for Next Point-of-Interest Recommendation. *ACM Transactions on Information Systems* 42, 2 (2023), 1–30.

[36] Chengfeng Xu, Pengpeng Zhao, Yanchi Liu, Jiajie Xu, Victor S Sheng S. Sheng, Zhiming Cui, Xiaofang Zhou, and Hui Xiong. 2019. Recurrent Convolutional Neural Network for Sequential Recommendation. In *The World Wide Web Conference.* 3398–3404.

[37] Ronghui Xu, Meng Chen, Yongshun Gong, Yang Liu, Xiaohui Yu, and Liqiang Nie. 2023. TME: Tree-guided Multi-task Embedding Learning towards Semantic Venue Annotation. *ACM Transactions on Information Systems* 41, 4 (2023), 1–24.

[38] 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.

[39] Dingqi Yang, Benjamin Fankhauser, Paolo Rosso, and Philippe Cudre-Mauroux. 2020. Location Prediction over Sparse User Mobility Traces Using RNNs: Flashback in Hidden States!. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence.* 2184–2190.

[40] Song Yang, Jiamou Liu, and Kaiqi Zhao. 2022. GETNext: Trajectory Flow Map Enhanced Transformer for Next POI Recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval.* 1144–1153.

[41] Feiyu Yin, Yong Liu, Zhiqi Shen, Lisi Chen, Shuo Shang, and Peng Han. 2023. Next POI Recommendation with Dynamic Graph and Explicit Dependency. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37. 4827–4834.

[42] Dabin Zhang, Meng Chen, Weiming Huang, Yongshun Gong, and Kai Zhao. 2024. Exploring Urban Semantics: A Multimodal Model for POI Semantic Annotation with Street View Images and Place Names. In *Proceedings of the 33rd International Joint Conference on Artificial Intelligence.* 2533–2541.

[43] Shiyuan Zhang, Tong Li, Shuodi Hui, Guangyu Li, Yanping Liang, Li Yu, Depeng Jin, and Yong Li. 2023. Deep Transfer Learning for City-Scale Cellular Traffic Generation through Urban Knowledge Graph. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining.* 4842–4851.

[44] Kangzhi Zhao, Yong Zhang, Hongzhi Yin, Jin Wang, Kai Zheng, Xiaofang Zhou, and Chunxiao Xing. 2021. Discovering Subsequence Patterns for Next POI Recommendation. In *Proceedings of the 29th International Conference on International Joint Conferences on Artificial Intelligence.* 3216–3222.

A Appendix

A.1 Location Description

This example illustrates a location description where, in the absence of historical data, the visit frequency for all time segments is set to 0. The time segments are sequentially ordered from "Sunday morning" to "Saturday night", with indices ranging from 0 to 27. All frequencies are rounded to three decimal places.

Given the historical visitation data of a location in JSON format, where each key indicates a specific time period (e.g., Sunday morning), and each value represents how frequently the location is visited during that time.
{'Sunday morning': 0.072, 'Sunday afternoon': 0.064,
'Sunday evening': 0.080, 'Sunday night': 0.040,
'Monday morning': 0.048, 'Monday afternoon': 0.040,
'Monday evening': 0.040, 'Monday night': 0.032,
'Tuesday morning': 0.048, 'Tuesday afternoon': 0.008,
'Tuesday evening': 0.040, 'Tuesday night': 0.024,
'Wednesday morning': 0.024, 'Wednesday afternoon': 0.040,
'Wednesday evening': 0.040, 'Wednesday night': 0.000,
'Thursday morning': 0.032, 'Thursday afternoon': 0.032,
'Thursday evening': 0.024, 'Thursday night': 0.024,
'Friday morning': 0.032, 'Friday afternoon': 0.024,
'Friday evening': 0.040, 'Friday night': 0.016,
'Saturday morning': 0.016, 'Saturday afternoon': 0.024,
'Saturday evening': 0.048, 'Saturday night': 0.048}.
Please summarize this location.

A.2 Generating Semantic Vectors for Pre-defined Location Categories

To generate the semantic vector for each location category, we input the category name along with its description into the LLM. Based on prior studies [25] and expert definitions, we define $M_c = 9$ location categories (see Table 7), ensuring a structured representation of mobility semantics.

For example, we provide the sentence: "*Residential: Locations where people typically spend weekday mornings and evenings, with longer stays on weekends.*". After processing this text via LLM, we extract the hidden state of the last layer corresponding to the input. To generate a compact semantic representation, we apply mean pooling over the hidden states, resulting in the final category embedding e_{cat}. **Note that all category embeddings are generated as a preprocessing step, adding no runtime overhead during model training and inference.**

A.3 Generating Semantic Vectors for Pre-defined User Group Profiles

To generate the semantic embedding for each user group profile, we input the profile name along with its description into the LLM. Based on prior work [31] and domain knowledge, we define $M_u = 6$ user prototypical group profiles (see Table 6) to cover the main behavioral patterns present in our data.

For example, the input sentence for a specific user group profile is: "*Night Owl: Users who are active at night—whether for work, study, or leisure—across both weekdays and weekends.*". The model processes this input, extracting the hidden state of the last layer corresponding to the provided text. To generate a compact and meaningful representation, we apply mean pooling over the hidden states, resulting in the final profile embedding e_{pro}. **Note that all profile embeddings are generated as a preprocessing step, adding no runtime overhead during model training and inference.**

A.4 Baselines Description

To evaluate the effectiveness of SILO, we compare it with a diverse set of baselines spanning statistical models, deep learning approaches, and recent LLM-based methods:

- **1-MMC** [9]: A first-order Markov model that predicts the next location based on the last visited location.
- **FPMC** [28]: A personalized Markov chain model that integrates sequential patterns with user-specific preferences via factorization.
- **DeepMove** [7]: A RNN-based model that captures mobility transitions with an attention mechanism to emphasize key historical locations.
- **Flashback** [39]: Enhances RNN-based modeling by dynamically weighting past check-ins using spatial and temporal decay functions.
- **GETNext** [40]: A graph-enhanced model that learns spatiotemporal dependencies by modeling location transitions as a graph structure.
- **C-MHSA** [12]: Introduces context-aware multi-head self-attention mechanisms for sequential modeling.
- **CSLSL** [31]: Introduces causal sequence modeling with a multi-task consistency constraint, explicitly structuring the "when → what → where" decision process.
- **MCLP** [30]: A transformer-based multi-context location prediction framework that integrates user preferences, temporal regularities, and sequential transitions for mobility prediction.
- **LLM-Mob** [34]: Applies pre-trained LLMs directly to mobility data by formulating location prediction as a sequence-to-text prompt-based task.
- **Mobility-LLM** [10]: Leverages LLMs with a visiting intention memory network and human travel preference for mobility pattern extraction.
- **NextLocLLM** [25]: Incorporates LLMs with enhanced POI embeddings to improve next location prediction.

A.5 Few-shot Learning Results

Under a 10% training data setting, SILO continues to outperform baselines, confirming its generalization ability in few-shot scenarios.

Table 5: Few-shot performance on Metropolitan B with 10% training data.

Method	Acc@1	Acc@5	Acc@10	MRR
GETNext	12.86	27.78	34.64	20.33
CSLSL	14.34	28.88	35.19	21.58
MCLP	13.76	30.77	38.84	22.17
Mobility-LLM	15.32	32.19	34.76	22.62
NextLocLLM	12.53	27.96	35.33	20.30
SILO	16.58	35.19	37.41	24.51

Table 6: Predefined user group profile and its descriptions.

User Profile	Detail Description
Night Owl	*Users who are active at night—whether for work, study, or leisure—across both weekdays and weekends.*
Early Bird	*Users who are active in the morning on weekdays, shifting to the afternoon on weekends.*
Evening Enthusiast	*Users who are most active during the evening, often showing a significant increase on weekends.*
Weekday Regular	*Users who consistently engage in morning and evening activities on weekdays, with reduced activity on weekends.*
Daytime Dweller	*Users who remain primarily active during the afternoon across all days.*
All-Day Active	*Users who are frequently active throughout the day, from morning to evening, reflecting a busy schedule.*

Table 7: Predefined location category and its description.

Location Category	Detail Description
Residential	*Locations where people typically spend weekday mornings and evenings, with longer stays on weekends.*
Work	*Locations primarily active during weekday mornings and afternoons, with limited activity on weekends.*
Leisure/Recreation	*Locations often visited on weekday afternoons, extending into mornings and evenings on weekends.*
Entertainment	*Locations frequently attended on Friday evenings and weekend nights, with fewer weekday morning visits.*
Shopping/Commercial	*Locations often visited during weekday afternoons and evenings, shifting to weekend mornings and afternoons.*
Education	*Locations attended mainly during weekday mornings and afternoons, with little to no presence on weekends.*
Healthcare	*Locations usually visited on weekday mornings and afternoons, with occasional visits on weekend mornings.*
Transportation	*Locations busier on weekday mornings and evenings, with more varied patterns on weekends.*
Public/Social	*Locations visited across all periods on both weekdays and weekends, though intensity varies.*