Secrets Everywhere: Auditing Memorization in Mobility Prediction Models

Anne Josiane Kouam
Inria & TU Berlin
Palaiseau, France

Hristo Boyadzhiev
BIFOLD & TU Berlin
Berlin, Germany

Konrad Rieck
BIFOLD & TU Berlin
Berlin, Germany

Abstract

Human mobility prediction models, which forecast the next location in a user's trajectory, are increasingly deployed in urban analytics, navigation, and personalized services. Yet, little is known about their potential to memorize and expose sensitive user trajectories from training data. While memorization has been extensively studied in language models, mobility prediction poses unique challenges: training sequences encode human behavior at various spatial and temporal scales, creating privacy risks at different granularities.

In this paper, we conduct the first systematic audit of memorization in mobility prediction models. While prior work has shown that privacy leaks can arise from such models, we systematically assess and quantify memorization risks at scale. We identify key challenges, including the lack of a randomness space, the multi-scale structure of trajectories, and user-specific behavioral diversity. To address these challenges, we introduce a framework to quantify mobility memorization at different levels of granularity: individual locations, anchor pairs, and subtrajectory segments. We also develop user-grounded reference sets to assess how likely a model is to prefer training data over realistic alternatives. Our evaluation across multiple models and datasets reveals pervasive memorization patterns that correlate with user regularity and increase the risk of data extraction at inference time. Our findings call for mandatory privacy auditing in mobility prediction models.

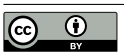
CCS Concepts

• **Information systems → Location based services**; • **Security and privacy → Privacy-preserving protocols**; • **Computing methodologies → Machine learning**.

Keywords

Location Privacy; Spatiotemporal Data; Sequence Modeling; Trajectory Extraction

ACM Reference Format:
Anne Josiane Kouam, Hristo Boyadzhiev, and Konrad Rieck. 2026. Secrets Everywhere: Auditing Memorization in Mobility Prediction Models. In *Proceedings of the 2026 ACM SIGSAC Conference on Computer and Communications Security (CCS '26), November 15–19, 2026, The Hague, Netherlands.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3830454.3832707



CCS '26, The Hague, Netherlands

ACM ISBN 979-8-4007-2871-6/2026/11
https://doi.org/10.1145/3830454.3832707

1 Introduction

Human mobility prediction plays a central role in modern urban analytics and decision-making, powering applications from smart transportation and crowd management to resource planning and personalized services. These models learn from historical location data to forecast individuals' next positions, enabling governments to optimize public transit and reduce congestion [23], while companies like Uber and mobile platforms anticipate demand and provide real-time, location-based recommendations [7, 41].

This rise in practical applications has been mirrored by a growing body of research seeking to understand and model human mobility [6, 20]. Early approaches relied on statistical formulations such as Markov chains or probabilistic transition matrices [29], while more recent efforts have explored a range of sequential deep learning architectures, including RNNs, LSTMs, attention-based models [9, 14, 31], and combinations with convolutional approaches [28]. These models typically adapt architectures from language modeling to geospatial data, achieving notable performance with accuracies sometimes exceeding 90% [10].

However, this predictive power might hide a privacy risk that has received little attention in the context of mobility prediction: Carlini et al. [4] demonstrates that sequential models can unintentionally store and reproduce sequences from their training data; a behavior that persists even in the absence of classical overfitting. This phenomenon, known as *memorization*, implies that models can regurgitate specific training samples verbatim under the right inputs. The risk is particularly acute for mobility sequences, where every data point corresponds to the real-world movement of an individual and thus carries intrinsic sensitivity.

We consider a realistic setting in which an adversary can interact with a trained mobility prediction model through its standard prediction interface (e.g., by querying location prefixes) and attempts to infer or reconstruct sensitive portions of training trajectories. For instance, a recommender system trained on commuters' GPS traces, when prompted with the known workplace of an individual, might continue the sequence with the exact locations that person typically visits after work. The returned data may include sensitive locations, such as home address or a medical clinic, as well as trajectory subsegments such as particular routes through the city.

In deep learning, concerns about unintended memorization are typically addressed through *privacy auditing*, which systematically evaluates how much information from the training data can be inferred from a model, enabling practitioners to detect and mitigate potential data leakage during or after training. One of the most influential frameworks in this area is *exposure* [4], which has become a standard diagnostic for large language models, integrated into the TensorFlow Privacy library [33]. Exposure quantifies memorization by testing how much a model assigns unusually high likelihood to

a known "secret" seen during training. It does so by comparing the model's likelihood for that secret against many randomly generated unknown sequences drawn from the same distribution. If the known secret ranks significantly higher than random candidates, the model is said to have memorized it.

In this paper, we establish that the *exposure* metric fails to capture the privacy risks specific to mobility data, for two main reasons:

- **Outliers vs. inliers.** Exposure relies on the assumption that secrets are *outliers*: sequences irrelevant to the learning task and unhelpful for improving accuracy. Assigning high confidence to such an outlier (e.g., a phone number) is clear evidence of memorization. This logic fits natural language models, where there is a clear boundary between what should and should not be learned. In mobility prediction, however, the situation is reversed. Every trajectory and location that improve predictive accuracy are also inherently sensitive: the data needed for generalization and the information that must stay private coincide. Here, secrets are *inliers* and high confidence may reflect the faithful reproduction of common, yet privacy-breaching, behavioral patterns.
- **Isolated vs. pervasive.** Exposure also assumes that secrets are *isolated*: rare, atypical sequences whose presence in the training corpus is limited to a few scattered occurrences. Under this assumption, memorization can be meaningfully summarized by a single global score capturing how the model retains such anomalies in general. In mobility data, however, sensitivity is *pervasive*: every trajectory corresponds to an actual individual's movements and therefore constitutes a potential leak. Consequently, memorization must take into account *every* training trajectory, making leakage intrinsically data-dependent. This shift requires analyzing a distribution of trajectory-level memorization signals, from which model-level summaries such as the maximum or tail percentiles can be derived.

As a result, existing exposure-based audits may substantially underestimate privacy risks in deployed mobility prediction systems, leaving sensitive user trajectories vulnerable to inference or extraction attacks. These fundamental differences call for rethinking how memorization should be measured in mobility prediction models. In this research, we address this gap by designing a framework to assess memorization at the level of individual training trajectories, guided by the following research questions:

(1) **How can we quantify mobility memorization?** We address this challenge by introducing a methodology that separates what a model *should learn*—the underlying mobility patterns and transition structures—from what it *should not memorize*, i.e., the exact coordinates and user-specific locations. This helps identify distinct dimensions of memorization such as sequence-level, anchor pair-level, and localized recall. To quantify these effects, we introduce three privacy metrics for individual trajectories: an adapted *exposure*, the *exposure preference*, and the *exposure magnitude*. Together, these metrics offer different perspectives on how a model memorizes original patterns over a reference set of realistic alternatives.

(2) **Which mobility patterns influence memorization?** Based on this data-centric view, we explore how the statistical properties of individual trajectories correlate with memorization. Through an extensive empirical evaluation on three public mobility datasets, ranging from 6,132 to 99,610 users and exhibiting distinct mobility dynamics, we find that memorization is far from marginal. In some settings, up to 85% of training trajectories show strong memorization signals. Moreover, models systematically memorize more the trajectories of users with routinary and low-entropy movements, revealing a clear link between mobility behavior and privacy risk.

(3) **How does model design impact memorization?** To explore this, we reproduce and compare a representative set of predictive mobility models covering both statistical and deep learning approaches [9, 13, 24, 28], This diversity allowed us to assess memorization across architectures, differing in complexity, temporal modeling capacity, and representational depth. We observe that while all models exhibit memorization, its extent and distribution vary with architectural design, suggesting that specific modeling choices can amplify memorization tendencies. Most notably, we found no correlation between our trajectory-level memorization metrics and Carlini et al. [4]'s model-level exposure, indicating that these measures capture fundamentally different dimensions of privacy risk.

(4) **Does memorization translate into extractability?** Finally, we assess whether highly memorized trajectories are also easier to recover from trained models. Using greedy extractability proxies that simulate an attacker repeatedly querying the model with plausible prefixes, we find a clear correlation: the higher a trajectory's memorization scores, the fewer attempts are needed to reconstruct it. This suggests that memorization not only reflects internal model behavior but also predicts real extraction risk at inference time.

Taken together, our work thus makes three main contributions. (1) We introduce a methodology for identifying unintended memorization in mobility prediction models. (2) We define complementary trajectory-level metrics that capture distinct memorization behaviors and enable privacy risk assessment. (3) We conduct a systematic empirical study showing that memorization varies across trajectories, datasets, and model architectures, with direct implications for privacy auditing and extraction risk. Our findings indicate that memorization in mobility prediction models is widespread, underscoring the need for systematic privacy auditing before deployment in real-world mobility services.

Roadmap. We begin in Section 2 with background and related work, followed in Section 3 by the main challenges of memorization in mobility data. Section 4 presents our framework and metrics, and Section 5 describes the experimental setup. Our empirical results are presented in Section 6, with broader implications discussed in Section 7. We conclude in Section 8.

Reproducibility. We publicly release the implementation of our memorization auditing framework. Details are provided in the Open Science appendix (Section A). Additional analyses and experimental results are available in the full version of the paper [16].

2 Preliminaries

We start by outlining the foundations of human mobility prediction and then introduce the exposure framework, a widely adopted tool for quantifying memorization in language models.

2.1 Mobility Prediction Models

A human mobility trajectory is represented as an ordered sequence of locations

$$T_u = (l_1, l_2, \ldots, l_n),$$

where each l_i denotes the latitude–longitude coordinate visited by user u at the i-th time interval.

In most datasets, time is discretized into regular slots such as 30 minutes or one hour, making the temporal ordering implicit in the index i and allowing trajectories to be treated as fixed-grid sequences of numerical coordinates. We focus on this representation, although our approach can also be extended to trajectories with non-uniform time steps.

The goal of mobility prediction modeling is to forecast the next location an individual will visit, given their historical mobility data. Formally, this entails learning a function

$$f_\theta : (l_1, \ldots, l_k) \longmapsto l_{k+1},$$

that maps a prefix of the trajectory to a predicted next location l_{k+1} in the time grid. This sequential prediction task is analogous to language modeling, treating locations like words and trajectories like sentences, such that mobility patterns are modeled probabilistically in temporal order.

Over the past decade, a wide spectrum of model architectures has been explored for next-location prediction [6, 20, 29]. Early approaches relied on *Markov chains* and other probabilistic transition models to capture short-range dependencies between locations [26, 27, 30]. While effective with limited data, these traditional methods struggled to capture long-range temporal patterns. The advent of deep learning brought recurrent neural networks (RNNs) and their gated variants (LSTMs, GRUs), which enabled learning longer-term dependencies from raw trajectories [14, 15]. Subsequent advances introduced attention mechanisms and spatio-temporal embedding techniques to better model periodicity, context, and user-specific preferences.

As an example, the *DeepMove* model augments a GRU-based sequence predictor with an attention layer to capture both long-term periodic behaviors and short-term sequential patterns [9]. Similarly, *LSTPM* model employs a context-aware network (with geo-dilated LSTM components) to integrate users' long-term and short-term location preferences [31]. Other innovations include models like *Graph-Flashback*, incorporating graph neural networks over a spatial-temporal graph [28]. Recently, large language models (LLMs) have also been applied to mobility prediction in a zero-shot or few-shot setting [35, 43]. However, since these models are not fine-tuned on target mobility datasets, evaluating their memorization remains out of this study's scope.

Overall, deep sequential models have achieved strong performance in next-location prediction, often exceeding 80–90% accuracy in urban datasets [6]. For broader coverage of statistical and learning approaches, we refer the reader to recent surveys [6, 20].

2.2 Memorization and Exposure

In language models, *memorization* refers to the model unintentionally preserving specific training examples that are irrelevant to its general task [4]. Carlini et al. [4] formalizes a way to quantify such memorization with an *exposure* metric. They insert known unique sequences called *canaries* into the training data (e.g., a random ID number within a sentence) and then measure how likely the trained model is to produce that exact sequence compared to all other possible sequences of the same format. This likelihood is measured via the log-perplexity, a proxy for the model's confidence in generating a given string. The set of all possible canary values defines a *randomness space R*.

By ranking all candidate canaries by their model likelihood, the exposure of a specific secret canary is defined as:

$$\text{exposure}(s) = \log_2 |R| \; - \; \log_2\big(\,\text{rank}\,(L(s))\,\big) \tag{1}$$

where $rank(L(s))$ is the position of the true secret in the model's likelihood ranking. Intuitively, this value represents how much the model reduces the uncertainty of guessing the secret. An exposure of 0 means no memorization, while a higher exposure indicates that the model has significantly memorized the canary.

The study by Carlini et al. [5] has prompted the development of different defenses. For example, deduplicating text corpora can reduce memorized outputs [18], while post-hoc approaches such as machine unlearning aim to remove memorized examples without retraining [19]. Similarly, the exposure metric has also influenced industry practices, including audits of production systems and the adoption of privacy-preserving tools like the TensorFlow Privacy library [33]. Despite this impact, research on memorization has remained concentrated on natural language [36] and, more recently, on image generation models [12, 17]. Comparable evaluations in other domains, particularly mobility data, are still scarce, which motivates the design of our auditing framework.

2.3 Privacy in Mobility Prediction Models

Mobility prediction models, especially those based on deep learning, have been recognized as potential vectors of privacy leakage. While raw and pseudonymized mobility traces have long been known to be highly identifying [8], only recently has the literature begun to uncover that predictive models trained on such data can also encode and leak sensitive user information.

Several studies have demonstrated various forms of leakage: model inversion attacks can reconstruct parts of users' trajectories from personalized predictors [2], while membership inference attacks can determine whether a user's data or a specific location was used in training, either at the individual [3] or aggregate level [25]. These findings reveal that predictive models can become as privacy-sensitive as the mobility data they are trained on. This realization is especially critical given the growing deployment of such models in real-world systems, where they can be probed or exploited without directly accessing raw data.

Positioning. In this paper, we build on existing work on privacy risks in mobility prediction models but go beyond showing that (and how) leakage can occur. Instead, we introduce the first framework for providing a quantitative and thus measurable account of this leakage. As inference-time leakage stems from models internalizing sensitive training data, memorization provides the ideal signal through which we can examine this behaviour. By quantifying how much user-specific information is preserved, memorization offers a general-purpose diagnostic tool useful both for anticipating leakage and for evaluating the effectiveness of privacy-preserving

techniques, establishing a foundation for privacy auditing in the domain of mobility prediction.

3 Challenges in Mobility Memorization

Diagnosing memorization in mobility models poses several unique challenges due to the structure of the data, the user-specific nature of mobility, and the complexity of model behavior. Unlike prior work in text domains, where canaries can be synthetically generated and inserted at scale, mobility traces require careful design of user-grounded references that preserve realism, structure, and spatial semantics. We outline the key conceptual and technical obstacles that shaped our auditing framework.

3.1 Multiple Scales of Memorization

Mobility trajectories are not flat sequences; they are structured behaviors composed of recurring locations, characteristic transitions, and sub-trajectories that reflect personal routines. This means that memorization can arise at several levels: a model may not need to retain an entire trajectory to compromise privacy; it may instead retain a specific location, an association between two points, or a movement pattern (e.g., home → gym → café). As a consequence, sensitive information can occur at any scale, from highly local to globally scattered across the trajectory.

This multi-layered structure is not fully captured by exposure, which tracks only one form of memorization: whether a model assigns disproportionately high likelihood to an *exact* sequence or token compared to plausible alternatives. In mobility data, however, privacy leakage can also stem from information beyond exact sequences that may still be sufficient to identify a user. This demands an assessment of how different patterns within a trajectory are retained and reemerge in the model's behavior.

3.2 Complex Randomness Space

Exposure quantifies memorization by comparing a known training sequence against many left-out candidates drawn from a randomness space, typically random strings that are meaningless to the model. This works for language models, where secrets are rare, context-independent outliers (e.g., ID numbers). In mobility prediction, however, we are concerned with the memorization of inlier sequences, i.e., realistic user trajectories that the model must learn from, yet should not regurgitate. Hence, the equivalent of a randomness space becomes much harder to define. Randomly sampling locations produces implausible trajectories that the model easily rejects by giving lower likelihood, not because it memorizes, but because the comparison set is behaviorally incoherent.

To measure whether a model memorizes a user's trajectory, we thus need reference sequences that are both *unseen* and *plausible*, capturing realistic patterns like commuting rhythms, temporal regularity, and geographic coherence. Moreover, because mobility behavior varies from person to person, these references must be constructed individually to reflect each user's movement style without replicating their true path.

3.3 User-Centric Evaluation

Finally, in mobility data, each training example corresponds to a specific individual, making memorization inherently user-specific. A model that stores part of one user's routine poses a privacy risk to that individual, even if it generalizes well overall. This user-centered nature of mobility brings added complexity: the number of plausible alternatives against which memorization should be assessed is not the same for every user. Differences in mobility entropy, visited locations, and routine imply that the randomness space R used in exposure (Eq. 1) varies across users.

Hence, exposure values are not directly comparable in the context of mobility: a high rank (strong memorization) with a small $|R|$ yields lower exposure than the same rank in a large reference set. Evaluating memorization, therefore, requires metrics that remain comparable despite heterogeneous random spaces, enabling a fair assessment of how much the model preserves data.

4 Methodology

To address the challenges in mobility memorization, we present a framework for quantifying how much a mobility prediction model memorizes individual training trajectories. The core idea is to distinguish what the model is expected to learn from what it could but should not memorize. Our evaluation proceeds in three conceptual steps, illustrated in Figure 1.

(1) **Behavioral abstraction:** For a given training trajectory, we define an abstraction that captures the structural pattern the model is expected to generalize from, such as mobility routines, temporal rhythms, or spatial transition types, while deliberately discarding user-specific identifiers (→ Section 4.1).

(2) **Reference set construction:** From the abstraction, we generate a reference dataset that reflects the distribution of plausible, behaviorally consistent alternatives. These reference trajectories are sampled from other users who exhibit similar patterns, ensuring they share high-level mobility structure but differ in sensitive content (→ Section 4.2).

(3) **Memorization quantification:** We measure how much more likely the model is to assign high confidence to the target trajectory compared to its reference set. Intuitively, a non-memorizing model should treat the original and reference trajectories similarly. A strong preference for the original trajectory indicates memorization (→ Section 4.3).

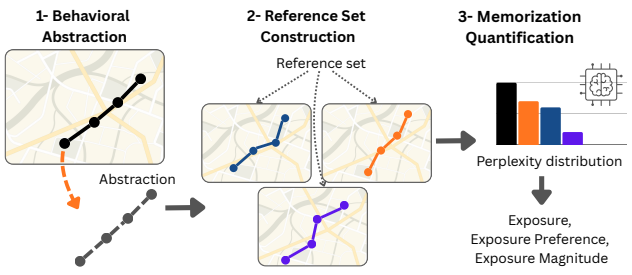


Figure 1: Overview of our memorization auditing framework.

4.1 Behavioral Abstraction

Mobility trajectories encode diverse facets of human behavior in the same data: a single sequence simultaneously reflects a user's

global movement style (e.g., distance range, regularity), anchor points such as home and work, and fine-grained routines embedded within specific time windows. This hierarchical structure makes it difficult to define a single notion of memorization (see Section 3.1). Indeed, a model may successfully generalize across some behavioral aspects, such as displacement rhythms, while memorizing others, such as locally repeated sub-sequences.

To address this, we express memorization in mobility prediction as a spectrum of risks rather than a binary property. Precisely, we distinguish several *memorization risks*, each targeting a specific type of behavioral information conveyed by the trajectory. For each risk, we specify what the model is expected to generalize versus what would constitute memorization. This decomposition allows us to isolate how models behave across different behavioral layers. To formalize each risk, we define a corresponding *trajectory abstraction* as a simplified representation of the signal we expect the model to generalize from, not memorize.

Definition 4.1 (Trajectory Abstraction). Let T_u denote a discretized trajectory of user u. A *trajectory abstraction* is a mapping

$$A : T_u \mapsto Z_u \in \mathbb{R}^d$$

that projects the raw trajectory into a fixed-size representation Z_u capturing the mobility patterns that the model is expected to generalize from rather than memorize.

Based on this abstraction, the memorization risk can be measured by how strongly the model distinguishes the specific training trajectory T_u from other sequences T'_u that satisfy the same abstraction, that is, $A(T'_u) = A(T_u) = Z_u$. The particular definition of A allows us to encode different types of memorization and quantify their presence in a model. In the following, we introduce three memorization risks and corresponding definitions of A, spanning from the retention of locations to leakage of movement routines. Although not exhaustive, these provide a structured lens to assess memorization across mobility properties.

4.1.1 Location memorization. This memorization risk arises when a model learns the specific locations visited by a user, rather than generalizing from the behavioral mobility patterns. For example, consider a user who remains stationary all day while teleworking. The desired behavior from the model is to learn that full-day stationarity is realistic in residential areas, not to memorize the specific coordinates of the user's home. Likewise, when people commute from home to work in the morning, the model should learn the pattern of commuting from residential to business zones, not the precise route or destinations of an individual.

From the perspective of abstraction, the model should thus generalize over *how* people move rather than *where*. The relevant pattern here is the user's displacement dynamics, expressed as the sequence of motion vectors over time, independent of absolute location. Let $T_u = (l_1, \ldots, l_n)$ denote the trajectory of user u, where l_i is the location visited at time step i. Consequently, we define the abstraction A as a sequence of first-order displacements:

$$A_{\text{motion}}(T_u) = (\Delta l_2, \ldots, \Delta l_n) \text{ with } \Delta l_i = l_i - l_{i-1}.$$

This abstraction removes absolute positioning, retaining only the user's trajectory shape. It captures whether the user remains stationary, their typical movement range, and directional tendencies, while eliminating specific locations that could be memorized. Each time step is treated uniformly, ensuring that mobility flow is preserved but spatial identity is removed. In other words, we can measure memorization in a model when it associates concrete locations with a trajectory T_u, even though learning $A(T_u)$ would have been sufficient for generalization.

4.1.2 Anchor-pair memorization. This risk occurs when a model memorizes the *association* between salient anchor locations frequently visited by a user, typically their home and workplace. Unlike the previous risk, which already captures the leakage of individual locations, here we specifically assess whether the model learns to link locations together in a way that can reveal the user's routines. Such anchor pairings are particularly sensitive because they form a behavioral signature: once one location is known, the model may trivially infer the other, even if not memorized independently. In this setting, however, we do not aim to prevent the model from learning that users exhibit anchor-based routines but we do expect it to generalize across users with similar patterns, rather than memorizing specific pairs.

To capture this expected behavior, we represent anchor usage through a coarse abstraction: the location of the primary anchor such as home or work and the commuting radius d associated with it. In the home-based variant, we define

$$A_{\text{home}}(T_u) = (l_{\text{home}}, d), \quad A_{\text{work}}(T_u) = (l_{\text{work}}, d),$$

where $l_{\text{home}} \in \mathbb{R}^2$ is the centroid of early morning locations, such as those before 6am, and $d \in \mathbb{R}_+$ is the average distance to the user's daytime activity region. Similarly, l_{work} denotes the centroid of mid-day locations, for example, between 10am and 6pm. While we focus on the two most common anchors, namely home and workplace, the approach also applies to any salient location that structures daily mobility, such as transport hubs, dinner spots, or activity centers.

4.1.3 Segment-level Memorization. The final memorization risk targets localized patterns within a user's trajectory, that is, short fragments that may reflect short routines or temporally localized behaviors. Unlike memorization of locations or anchor pairs, segment-level memorization concerns the model's tendency to rigidly learn specific short-term movement traces, even when they are not semantically essential. For instance, a user may often visit different places in a particular order around lunchtime. While such a loop is behaviorally plausible, the model should not memorize the exact sequence of locations if the underlying routine can be expressed more flexibly. Memorization here would mean that the model fails to generalize across slight variations of this local pattern, treating one specific realization as canonical.

To capture this risk, we define the abstraction over a localized segment of the trajectory treated as an interchangeable placeholder. Specifically, for a continuous subsequence $T_u^{[s:e]} = (l_s, \ldots, l_e)$, we abstract it as a masked slot within the full sequence:

$$A_{\text{segment}}(T_u) = (l_1, \ldots, l_{s-1}, \star, l_{e+1}, \ldots, l_n),$$

where $\star$ denotes the masked segment that can be substituted by any plausible sub-trajectory.

This abstraction reflects the intuition that users may vary their short-term routines (e.g., different lunch routes or minor errand

sequences) while preserving overall behavior. The goal is to ensure the model generalizes across such variations, without retaining one specific realization of a sub-pattern.

4.2 Reference Set Construction

An ideal model without memorization should generalize beyond specific trajectories to capture the underlying behavioral patterns rather than memorize where or when they occurred. For each trajectory T_u used in training, we seek to define a reference set $\mathcal{R}(T_u)$ of unseen trajectories that express the same behavioral pattern, such that a model generalizing correctly should assign them comparable likelihoods. However, the construction of this set critically depends on how it abstracts from the particular target, and so we introduce an ideal and an approximate construction strategy.

4.2.1 Ideal reference construction. In the ideal case, we are able to synthesize new trajectories for the reference set that are (almost) indistinguishable from real behavior and originate from the same data distribution as the target trajectory.

Definition 4.2 (Ideal Reference Set). Given a behavioral abstraction A and a training trajectory T_u, the ideal reference set is the collection of held-out trajectories (i.e., not seen during training) that satisfy the same abstraction as T_u:

$$\mathcal{R}(T_u) = \left\{ T'_v \in \mathcal{T}_{\text{held-out}} \mid A(T'_v) = A(T_u) \right\}.$$

Constructing such ideal reference sets, however, is only possible in very specific cases, as their feasibility depends strongly on the abstraction A in question. While it might be possible to synthesize a trajectory T'_v satisfying $A(T'_v) = A(T_u)$, we have limited control over whether it is truely realistic. For example, a commute between two residential areas may be plausible in one neighborhood but infeasible elsewhere because of barriers.

Fortunately, for *segment-level memorization* (Section 4.1.3), constructing realistic alternatives is feasible. We can replace the segment with a plausible alternative from the same user. In particular, we replace masked subsequences with alternatives drawn from the same user, including: (i) *substitute*, where the segment is replaced with an alternative observed in the same time window on different days; (ii) *shuffle*, where the same locations are reordered; and (iii) *stationary*, where a single randomly chosen location is repeated across the entire segment duration. Although synthetic, these segments are derived from real behavior to preserve local temporal and spatial plausibility.

4.2.2 Approximate reference construction. For several abstractions, an ideal construction is not possible. To address this, we employ an approximate approach that leverages the diversity of real trajectories. Our method begins by applying the abstraction A to each trajectory, producing a feature representation that captures the intended behavioral signal. We then cluster these abstractions using an appropriate distance metric (such as the Euclidean distance), grouping users who exhibit similar mobility behaviors. From each cluster, we then select a representative trajectory T_u to serve as the training instance, while all other trajectories in the same cluster form its reference set.

Definition 4.3 (Approximate Reference Set). Let $\{A(T_i)\}_{i=1}^{N}$ be the abstractions computed over all trajectories in a population. Let $\text{cluster}(\cdot)$ assign trajectories to clusters in the abstraction space. Then, for each medoid trajectory T_u selected from a cluster C, its approximate reference set is defined as:

$$\tilde{\mathcal{R}}(T_u) = \left\{ T'_v \in \mathcal{T}_{\text{held-out}} \mid \text{cluster}(A(T'_v)) = \text{cluster}(A(T_u)) \right\}.$$

This clustering based approach naturally adapts to the population distribution. In our framework, we use it to construct approximate reference sets for the risks of *location memorization* (Section 4.1.1) and *anchor pair memorization* (Section 4.1.2).

When constructing these sets, we observe that common mobility patterns lead to larger clusters and thus richer reference sets. In contrast, users with more unique patterns may fall into smaller clusters, resulting in limited references and potentially biased memorization scores. To mitigate this, we introduce a cluster enrichment strategy that leverages the fact that trajectories from the held out set can appear in multiple reference sets. Our algorithm (see Appendix 1) augments small clusters by incorporating nearby neighbors in the abstraction space, subject to a minimum similarity threshold.

4.3 Memorization Quantification

Equipped with appropriate reference sets, we can now turn to the measurement of memorization. For this, we follow the design of the exposure metric but reformulate it over the reference set and derive more fine grained statistics from it. Given a trajectory T_u from the training dataset $\mathcal{T}_{\text{train}}$ of a model and its corresponding reference set $\mathcal{R}(T_u)$, we define three complementary memorization metrics based on model likelihood $\mathcal{L}(T)$, computed as the negative log-perplexity assigned by the model to T_u.

(1) Exposure (Carlini et al. [4]). As introduced earlier, exposure measures how highly T_u ranks within its reference set. We can thus formulate it as follows for mobility memorization:

$$\text{exposure}(T_u) = \log_2 |\mathcal{R}(T_u)| - \log_2 \text{rank}(\mathcal{L}(T_u)).$$

In contrast to the original formulation for text, however, the exposure now depends directly on the reference set size $|\mathcal{R}(T_u)|$, making its scale non-uniform across users. As a result, quantitatively comparing memorization for different abstractions and trajectories becomes impossible.

(2) Exposure Preference. This metric captures where T_u stands within the likelihood distribution of its reference set by normalizing its rank into a value in $[0, 1]$:

$$\text{preference}(T_u) = \frac{\text{rank}(\mathcal{L}(T_u))}{|\mathcal{R}(T_u)|}.$$

Values close to 1 indicate that the model does not favor the training trajectory over its references (no memorization), whereas values near 0 signal that the model assigns unusually high preference to T_u, consistent with memorization. Unlike exposure, the preference is unaffected by the absolute size of the reference set and therefore supports population-level comparison.

(3) Exposure Magnitude. To capture the *magnitude* of memorization, we compute how much higher the model's likelihood for T_u is compared to the average alternative:

$$\text{magnitude}(T_u) = \mathcal{L}(T_u) - \mathbb{E}_{T'_v \sim \mathcal{R}(T_u)}[\mathcal{L}(T'_v)].$$

Since $\mathcal{L}$ denotes negative log-perplexity, positive values indicate that the model assigns an advantage to the exact training trajectory, consistent with memorization. Values near zero reflect appropriate generalization, whereas negative values reflect an absence of memorization and a preference for more typical behavior.

Together, these metrics offer complementary perspectives on memorization: ordinal exposure, normalized position, and magnitude, providing a multifaceted evaluation framework.

5 Experimental Setup

We continue to describe the empirical setup used to evaluate memorization risks in mobility prediction models. This includes the datasets, preprocessing steps, and model configurations employed throughout our experiments. Our aim is to assess memorization across different urban environments and model architectures using real-world human mobility traces.

5.1 Mobility Datasets

For our experiments, we consider three public mobility datasets, which are summarized in Table 1. These datasets span diverse urban contexts and differ in sampling modalities, spatial granularity, and population scale, providing a comprehensive setup for our investigation of mobility memorization.

Shanghai Telecom [22]. This dataset records anonymized mobile phone activity from 9,481 users in Shanghai, China, collected via device associations with 3,233 base station. Each entry corresponds to a timestamped connection event (i.e., mobile data), enabling the reconstruction of user mobility over periods ranging from two weeks to six months. The data reflects natural usage patterns in a cellular network and supports long-range trajectory analysis.

YJMob100K [37]. This dataset captures fine-grained GPS mobility traces for 100,000 users in a Japanese urban region over a continuous 75-day period. Location points are sampled every 30 minutes and discretized over a uniform 500-meter grid. Each grid cell is enriched with POI category annotations, providing both spatial and semantic context. The data was collected through a smartphone application with opt-in user consent.

Shenzhen Urban [39, 40]. Collected from cellular tower logs in Shenzhen, China, this dataset contains over 38 million events for approximately 414,000 users. Each record corresponds to a timestamped network event (call/text message) between a user and one of 1,090 cell towers. Although explicit dates are anonymized, we assume each user is active for a distinct number of days and that the dataset is chronologically ordered per user, following the approach of Yonga et al. [38]. This results in a highly skewed distribution of active days, with a median of approximately 9 days, while a minority of users remain active for up to 949 days.

5.2 Preprocessing

To standardize and structure the raw datasets for predictive modeling, we design a uniform preprocessing pipeline that transforms the heterogeneous dataset records into a common sequence format. Summary statistics for this preprocessing are reported in Table 1. The process comprises four main stages:

Harmonization. All datasets are first converted into a normalized format with four fields: *UserID*, *Latitude*, *Longitude*, and *Timestamp*. This ensures compatibility across datasets with varying formats. For sources that only provide time-of-day information (*Shenzhen Urban* and *YJMob100K*), artificial dates are assigned per user starting from day 0 and incremented chronologically.

Discretization. In general, spatial discretization is a necessary step for mobility analysis, particularly when working with raw GPS trajectories, where exact location matches are unlikely due to measurement noise and floating-point precision. Without discretization, trajectories may reflect artificial movement rather than meaningful user behavior, making it difficult to identify recurring locations or periods of stationarity. Therefore, it is standard to apply noise filtering and spatial discretization techniques, such as grid-based tessellation or hierarchical spatial indexing (e.g., H3 [34]), as discussed in [42]. In our case, the datasets already provide discrete spatial representations (e.g., cell towers or grid-based locations), and no additional spatial discretization is required.

We then perform temporal discretization to align records to a fixed sampling frequency. All trajectories are resampled at uniform *30-minute intervals*, consistent with the resolution of *YJMob100K*. For each interval, the most frequent location is retained, while missing intervals are handled in a later preprocessing step.

Anchor inference and completion. Each user's home and work locations are inferred as the most frequently visited places during night hours (23:00–06:00) and work hours (09:00–12:00 and 14:00–18:00), respectively. These inferred anchors are then used to complete missing values during their corresponding time windows. Intervals outside these periods, such as lunch breaks or evening activities, remain unfilled.

User and day filtering. We keep only users with sufficient activity, defined as having at least 5 valid days, where a day is considered valid if it contains less than 30% missing intervals. From these, we select each user's 14 best days (i.e., those with the lowest proportion of missing data) and segment them into non-overlapping trajectories of 3 consecutive days. This choice balances two goals: (i) providing sufficiently long sequences for effective training of mobility models, and (ii) increasing the number of trajectory samples for downstream memorization analysis, particularly approximate reference set construction.

Table 1: Summary of the mobility datasets used in our experiments, including statistics before and after filtering.

Feature	**Shanghai Telecom**	**YJMob100K**	**Shenzhen Urban**
City	Shanghai, China	Japan	Shenzhen, China
#Records (raw)	7.2M	111.5M	38.2M
#Users (raw)	9,481	100,000	414,271
#Locations (raw)	3,233	34,032	1,090
Temporal Span	up to 180 days	75 days	up to 949 days
#Users (filtered)	6,132	99,610	78,591
#Locations (filtered)	2,939	30,785	965
#Trajectories	22,993	303,779	398,440

5.3 Mobility Prediction Models

To evaluate memorization across different setups, we select one representative model from each major architecture used in mobility modeling. These include both classical probabilistic models and state-of-the-art deep learning architectures.

- *Markov model [24].* We implement a classic second-order Markov chain that predicts the next location from the two most recent locations in a user's trajectory. This provides a strong probabilistic baseline capturing short-term transition dynamics.
- *Long Short-Term Memory (LSTM) [13].* We implement a standard LSTM-based prediction model (LSTM-simple) that operates on 24-hour input sequences. Locations and timestamps are embedded, concatenated, and passed through the LSTM, followed by a softmax output layer. A variant, LSTM-long, processes the full 72-hour history to examine the effect of longer context.
- *DeepMove [9].* This attention-based model splits the trajectory into a long-term history and a current-day segment, using an encoder–decoder structure to capture periodic cross-day patterns. We evaluate two variants: *DM-locallong*, which attends locally from the current-day decoder to a 48h history encoder, and *DM-avglonguser*, which instead pools the entire history into a global context vector and includes a user embedding.
- *LSTPM [31].* This model captures both long-term and short-term mobility preferences. It aggregates trajectories into 48 temporal bins (hour-of-day × weekday/weekend), computes inter-bin similarities to derive long-term preferences, and combines them with a geo-dilated RNN for short-term modeling. The resulting context is used to generate personalized next-location predictions.
- *Graph-Flashback [28].* This model represents mobility trajectories as a Point-of-Interest (POI) transition graph. A graph convolutional network (GCN) learns POI embeddings based on structural and temporal context. These embeddings are then integrated with user histories to enhance next-location prediction.

We use official implementations whenever available[1]. Hyperparameters are left largely unchanged from the original repositories; only dataset-dependent parameters (e.g., number of locations) are adapted. All models are trained using a time-based 80–20 train–validation split with early stopping based on validation loss. A complete summary of hyperparameter settings is provided in the full version of this paper [16] (Table 4).

5.4 Experimental Protocol

For the measurement of memorization with our framework, we construct reference sets tailored to each of the three abstractions introduced in Section 4.1. In all cases, we fix the training set size to 2,000 trajectories, which provides sufficient behavioral diversity for model training while leaving the remaining trajectories available to form large, diverse reference sets for each training sample.

[1]DeepMove: https://github.com/vonfeng/DeepMove; LSTPM: https://github.com/NLPWM-WHU/LSTPM; Graph-Flashback: https://github.com/kevin-xuan/Graph-Flashback.

Location and anchor-pair memorization. For the first two memorization risks, we construct approximate reference sets by partitioning user behaviors using k-means clustering, where we fix the number of clusters to 2,000. Each cluster is represented by one trajectory used for training, while the remaining trajectories form its reference set. To ensure that memorization metrics are computed over sufficiently large and stable samples, we apply a post-processing enrichment step (see Algorithm 1). This step is governed by two parameters: the target minimum reference set size $k_{\min}$ and a distance threshold τ controlling similarity. The parameter $k_{\min}$ determines the minimum number of trajectories required in each reference set to obtain statistically reliable estimates, while τ limits how dissimilar additional trajectories can be when enriching a cluster. These parameters induce a trade-off: smaller τ values preserve behavioral coherence but may prevent reaching $k_{\min}$, whereas larger values facilitate reaching $k_{\min}$ at the cost of reduced similarity.

In practice, τ is inherently dataset- and abstraction-dependent, as it depends on the scale and variability of trajectory features, and is therefore not fixed globally. Instead, trajectories are added in order of increasing distance until the target size $k_{\min}$ is reached, resulting in an implicit threshold. In contrast, $k_{\min}$ should be chosen relative to the dataset size: it must remain small compared to the total number of trajectories to preserve behavioral coherence within each reference set, while being large enough to ensure stable estimation. In our datasets (ranging from ∼23K to ∼398K trajectories), we set $k_{\min} = 100$ as a rule of thumb that provides a good balance between these objectives. Under this setting, reference set sizes vary significantly across datasets (from 100 to 144,891; see Figure 9 in [16]), and the resulting effective values of τ also differ across memorization risks (e.g., $\tau \approx 1.67$ for location memorization in Shenzhen versus $\tau \approx 45.31$ for anchor-pair normalization in YJ-Mob100K). Finally, to ensure that clusters capture shared behavioral patterns rather than user identity, we analyze the spatial dispersion between cluster representatives and their reference sets (Figure 10 in [16]), confirming that reference trajectories remain sufficiently varied while still reflecting consistent mobility behaviors.

Segment-level memorization. To evaluate segment-level memorization, we generate controlled variants of real trajectories. We again sample 2,000 trajectories for training and construct reference sets by perturbing short temporal segments. Each trajectory is divided into non-overlapping 4-hour windows (00:00–20:00), providing multiple localized positions where alternative behavior can be inserted. A short window is chosen to preserve realism, as larger perturbation spans tend to produce implausible mobility patterns. For each window, we generate synthetic references using three transformations—*shuffle*, *stationary*, and *substitute* (cf. Section 4.2)—and balance them to obtain 300 references per trajectory.

Model performance. The resulting training subsets are then used to train mobility prediction models using the hyperparameter settings reported in Table 4 of the full version of this paper [16]. We report the corresponding predictive performance in Table 2.

Overall, the models exhibit comparable accuracy trends across datasets, with *DM-locallong* achieving the strongest performance

and *Markov* remaining surprisingly competitive despite its simplicity. One exception is *Graph-Flashback*, whose accuracy is substantially lower. This behavior is consistent with the original architecture, which relies on relatively small embedding and hidden dimensions (10 units), limiting its capacity to capture complex mobility patterns. While its performance on smaller datasets remains within the range reported in prior work [28], it degrades on larger and more diverse datasets such as *ShanghaiTel* and *YJMob100K*, where trajectory variability is higher, following the general trend of decreasing accuracy with increasing location entropy (see Table 1). Importantly, this lower predictive performance provides a useful contrast, allowing us to study memorization under a lower-capacity regime where the model captures less global structure and may rely differently on memorization. As such, Graph-Flashback serves as an informative outlier highlighting how memorization behaviors vary across models with different representational capacities.

These results confirm that models capture mobility structure to varying degrees while being far from highly optimized; a setting well suited for analyzing memorization behavior across architectures with different capacities.

Table 2: Average (top-1 and top-5) models' accuracy.

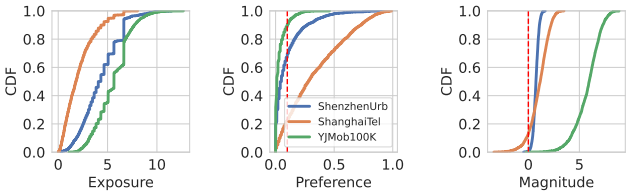
Model	ShenzhenUrb.		ShanghaiTel.		YJMob100K	
	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5
Markov (2nd)	91.3	92.3	76.2	79.7	22.5	38.1
LSTM-simple	89.7	92.8	73.2	80.2	18.2	22.9
LSTM-long	90.0	93.4	78.0	84.8	18.8	24.5
DM-locallong	92.7	96.0	84.6	87.6	20.2	25.3
DM-avglonguser	92.4	95.0	71.4	79.2	18.1	22.9
LSTPM	86.5	92.7	58.6	68.2	15.5	20.4
Graph-Flashback	32.0	57.2	10.4	23.2	00.3	00.7

6 Empirical Findings

We structure our discussing of findings along the research questions formulated in the introduction. In particular, we explore whether memorization occurs (RQ1: Section 6.1), what best explains it (RQ2: Section 6.2), how it varies across models (RQ3: Section 6.3), and how it relates to extractability (RQ4: Section 6.4). Each question is addressed under controlled conditions by fixing key factors such as model, dataset, or memorization risk and varying a single dimension to isolate its effect.

6.1 Detecting Individual Memorization

We begin by assessing whether mobility memorization arises when training on our three datasets. For this experiment, we focus on the model *LSTM-simple*, as it provides a simple and interpretable neural baseline for analyzing memorization behavior. Experiments involving all models are discussed later in Section 6.3. We train LSTM-simple on each dataset and compute the proposed memorization metrics under the *location memorization* abstraction. The training dynamics are shown in the full paper [16, Figure 11], where we observe a continuous decrease in loss, indicating that overfitting has not occurred.



(a) Exposure CDF **(b) Preference CDF** **(c) Magnitude CDF**

Figure 2: Cumulative distribution of mobility memorization metrics for the *LSTM-simple* model across datasets.

Figure 2 shows the distributions of memorization metrics for all training trajectories. Across the three datasets, we observe clear and systematic memorization signals.

Memorization on ShenzhenUrb. Memorization is strongest for this dataset with low mobility entropy. Over 99.4% of trajectories have positive exposure magnitude, with a mean of 0.74. Exposure values are high (median 4.19), and 67% of trajectories stand within the bottom 10% of their reference likelihood distribution, showing strong preferential treatment for true training sequences.

Memorization on ShanghaiTel. For this dataset, patterns are qualitatively similar but attenuated. Exposure magnitude remains positive for 87.1% of trajectories (median 1.16). Exposure scores decrease substantially (median 1.79), and only 21.8% of trajectories appear in the bottom 10% preference region, indicating weaker but still non-negligible memorization.

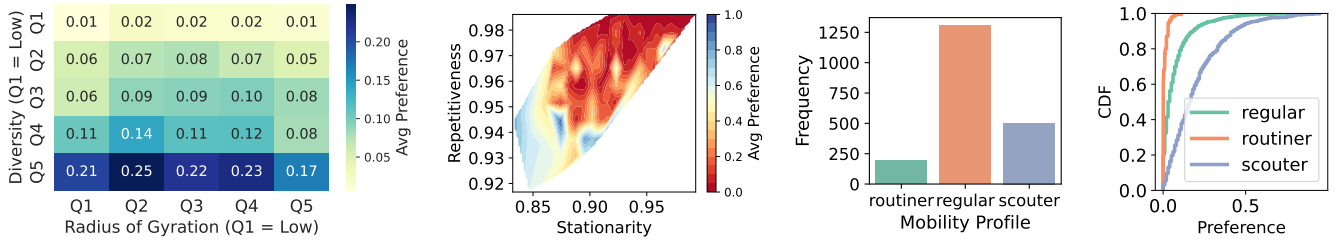
Memorization on YJMob100K. Despite being the most heterogeneous dataset, memorization manifests strongly. Exposure magnitude is positive for 99.95% of trajectories, with a large median of 5.95. Exposure remains high (median 5.64), and a striking 88.9% of trajectories fall within the most confident 10% of preference scores, revealing a strong model tendency to favor true sequences.

Summary: *Our findings demonstrate that mobility memorization arises across datasets of varying scale and entropy. Even without overfitting, prediction models tend to retain user-specific locations rather than generalize from behavior.*

6.2 Who is Memorized and Why

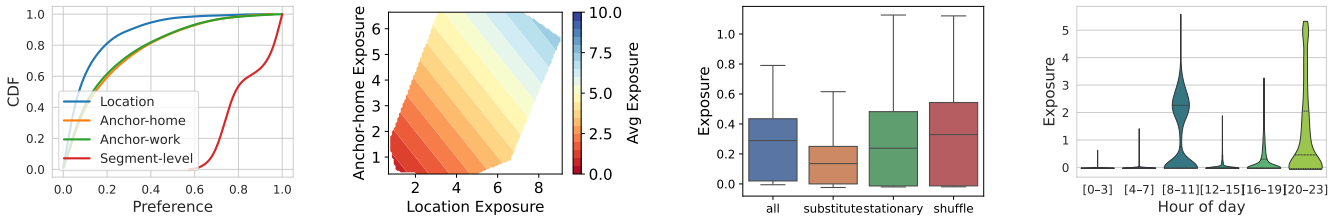
To understand which users are more likely to be memorized by predictive models, we analyze how their individual behavior correlates with the memorization metrics. To obtain a clearer grasp of this behavior, we focus on four well-established features known to capture key properties of human movement [1, 11, 21, 32]:

- rep *repetitiveness*, defined as the fraction of returns to previously visited locations within a trajectory.
- sta *stationarity*, the proportion of time spent at the same location across consecutive intervals, indicating sedentary behavior.
- div *diversity*, measured as the number of distinct sub trajectories, reflecting spatial variability.
- rg *radius of gyration*, which quantifies the spread of movement, with higher values indicating broader coverage.



(a) Preference (div vs. rg) **(b) Preference (sta vs. rep)** **(c) Mobility profiles.** **(d) Preference (profiles).**

Figure 3: Mobility characteristics and memorization patterns for *LSTM-simple* on the *Shenzhen Urban* dataset. (a) Heatmap of memorization preference as a function of diversity (div) and radius of gyration (rg); lighter colors indicate lower preference and thus higher memorization. (b) Contour plot of memorization preference with respect to stationarity (sta) and repetitiveness (rep); cooler (blue) regions correspond to lower memorization (higher preference). (c) Distribution of user mobility profiles (frequency of trajectories per profile). (d) Cumulative distribution of memorization preference across profiles.



(a) Preference across memorization types. **(b) Exposure comparison: Location vs. Anchor-home.** **(c) Exposure variation across transformations (Segment-level).** **(d) Exposure by time-of-day segments (Segment-level).**

Figure 4: Memorization behavior across risks, transformations, and sub-trajectories (*LSTM-simple, Shenzhen Urban*). (a) Cumulative distribution of preference across memorization types. (b) Contour plot comparing exposure between location-level and anchor-home memorization; cooler (blue) regions indicate higher exposure (and thus higher memorization). (c) Boxplot of exposure across segment-level transformations; boxes represent the interquartile range (IQR), the central line denotes the median, and whiskers extend to 1.5×IQR. (d) Violin plot of exposure across time-of-day segments; the width of each violin reflects the density of trajectories, with wider sections indicating more frequent exposure values, while central markers indicate median values. Higher exposure corresponds to higher memorization.

Based on these features, the users are grouped into three behavioral mobility profiles, i.e., *routiner*, *regular*, and *scouter*, which represent decreasing levels of spatial and temporal regularity [1]. Further details on this analysis and the used features are presented in the full paper [16, Section B]. When investigating the Shenzhen Urban dataset with LSTM-simple, we observe consistent and interpretable patterns, as shown in Figure 3. Corresponding analyses for other datasets are discussed in [16, Appendix B, Figure 12], with similar structural trends but differences in magnitude and distribution.

Radius of gyration. First, we find that users with low diversity (Q1 < 0.6) and small radius of gyration (Q1 < 10.73 km) are most memorized, with exposure preferences concentrated near zero (see Figure 3a). Interestingly, we find that the radius of gyration exhibits a U-shaped effect. Users with either very low (Q1) or very high (Q4 > 20.82 km) gyration values tend to be memorized more, while those with mid-range values are less retained. This latter trend, however, is not consistent across all datasets (see [16, Figure 12]).

Stationarity and repetitiveness. Second, we observe that memorization generally increases with both stationarity and repetitiveness. Users who tend to stay in the same place and frequently return to previously visited locations during movememnts are more likely to be memorized compared to users with less stationary and less repetitive activities (see Figure 3b).

Mobility profiles. Finally, similar effects are also reflected in the mobility profiles: Figure 3c and Figure 3d show that *routiners* (9.5%), characterized by high stationarity and low diversity, tend to experience stronger memorization, with memorization exposure preferences tightly concentrated below 0.1. They are followed by *regular* users (65.3%), who show moderate retention. In contrast, *scouters* (25.2%), who demonstrate greater spatial and temporal variability, are memorized far less, with exposure preference distributions skewed higher, indicating lower exposure.

Summary: *Our findings reveal that memorization in predictive mobility models is far from uniform. It tends to disproportionately*

affect users with regular, repetitive, and localized mobility patterns, which are easier to learn and thus more likely to be memorized.

6.3 Memorization Across Risks and Models

To further explore how memorization behaves under different settings, we investigate how our metrics vary across abstractions and model architectures. Using the Shenzhen Urban dataset, we again show our analysis on the LSTM-simple model to isolate the effects of varying abstractions and generalize later across model variants. Equivalent analyses for other model architectures are discussed in [16, Appendix B, Figure 13], with similar trends but variations in magnitude and consistency across models.

Memorization risk varies by abstraction. Figure 4a shows that the same model exhibits different levels of memorization depending on the abstracted mobility pattern. We observe stronger memorization for location sequences, indicated by a highly skewed distribution toward low preference values (with 80% lower than 0.2); more moderate memorization for anchor pairs, with exposure preferences more evenly spread; and much lower memorization for segment-level patterns, where the preference values cluster around 0.6–1.0.

This contrast between abstractions holds despite varying degrees of abstraction, i.e., the extent to which each abstraction modifies the input sequence: full-trajectory changes for location-based memorization, partial windows (e.g., morning or evening hours) for anchor-based memorization, and short sub-sequences (e.g., 4-hour slices) for segment-level analysis. Similar patterns are generally observed across other model architectures ([16, Figure 13]), although the magnitude of the differences varies and segment-level memorization is not uniformly negligible.

Anchor-pair and location memorization are correlated. Figure 4b compares user-level memorization exposure between anchor-pair (home) and location-based settings, focusing on the 109 users present in both training subsets. Despite the limited overlap, we observe a positive relationship: users who are memorized in the location-based setting tend to also be memorized under anchor-based conditions. Comparable cross-abstraction relationships are also observed for other architectures, as reported in [16, Figure 13]. A similar trend holds between home-based and work-based anchor pair memorization, see [16, Figure 14].

Transformation matter in segment-level memorization. We observe that the nature of transformations influences memorization. Figure 4c compares different transformations at the segment level and shows that exposure generally remains low, yet the form of transformation matters: *shuffle* introduces the most deviation, yielding the highest exposure values. This suggests that the model struggles slightly more when sequences are reordered unexpectedly. In contrast, *substitute* and *stationary* modes yield lower exposure, reflecting the model's ability to tolerate more semantically plausible changes, such as swapping a routine with another similar one or remaining at a single location. This ranking is observed in several models, although the differences between transformations can be smaller or partially overlapping depending on the architecture.

In Figure 4d, we further explore how memorization varies by the time of day at which the transformation occurs. The distribution reveals almost no exposure during nighttime hours (0–7h),

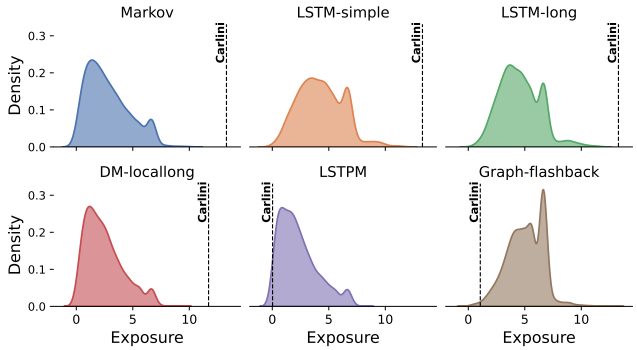


Figure 5: Exposure distributions across models on the *Shenzhen Urban* dataset, with exposure markers from Carlini et al. [4]. Each curve represents a kernel density estimate of exposure values across trajectories. The shape of the density reflects how exposure is distributed within a model: peaks indicate common exposure levels, while spread and tails capture variability and the presence of high memorization.

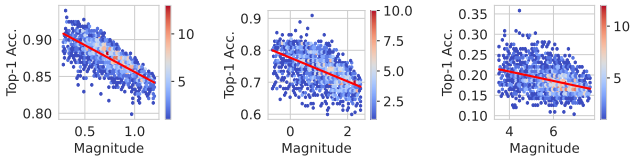
where user activity is minimal and patterns are uniform. By contrast, exposure exposure tends to be higher during two windows: early morning (8–11h) and late evening (20–23h). These intervals often capture transitions such as commuting or post-work activities, where user routines may be less consistent. Interestingly, exposure values during these periods can exceed 5 in some cases, despite segment-level memorization generally being low, indicating that for some users, the model retains highly specific behaviors even in varying, non-routine parts of the day. Similar temporal tendencies can be observed in some architectures (e.g., DM-avglonguser), although the effect is less pronounced or absent in others ([16, Figure 13].

Summary: *Memorization tends to concentrate on long-term and routine-based behaviors, while short segments are generally less memorized. Users memorized under one abstraction often remain vulnerable under others, although the strength of this effect varies across models. Segment-level memorization can still occur during transition periods, where user behavior is more distinctive.*

Model architectures yield distinct memorization patterns. We now examine how memorization varies across model architectures under

Table 3: Memorization metrics (mean ± std) for all models on Shenzhen Urban under the *location* abstraction.

Model	Exposure	Preference	Magnitude
Markov	2.82 ± 1.87	0.25 ± 0.24	2.97 ± 2.10
LSTM-simple	4.31 ± 1.95	0.10 ± 0.13	0.74 ± 0.28
LSTM-long	4.57 ± 1.77	0.07 ± 0.09	0.77 ± 0.24
DM-locallong	2.49 ± 1.68	0.28 ± 0.23	0.47 ± 0.31
DM-avglonguser	3.02 ± 1.87	0.22 ± 0.22	0.49 ± 0.26
LSTPM	2.31 ± 1.69	0.32 ± 0.26	0.63 ± 0.68
Graph-Flashback	5.07 ± 1.57	0.05 ± 0.08	469.15 ± 1230.36



(a) Shenzhen Urban **(b) Shanghai Telecom** **(c) YJMob100K**

Figure 6: Utility (Top-1 Accuracy) vs. Memorization (Magnitude) across datasets using *LSTM-simple*. Each hexagonal bin aggregates trajectories with similar values, and the color scale indicates the number of trajectories within each bin, with warmer colors corresponding to higher concentrations.

the *location* abstraction. Results for *ShenzhenUrban* are shown in Figure 5, with analogous patterns observed for *ShanghaiKaggle* and *YJMob100K* (see [16, Figure 13]).Additionally, Table 3 additionally shows all metrics for all models.

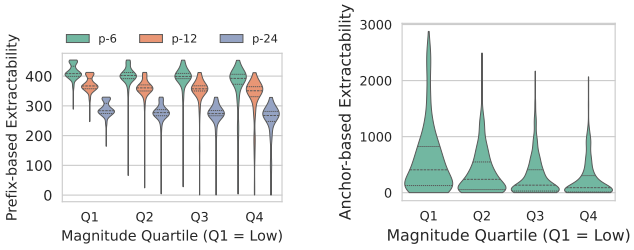
Exposure distributions differ across models in both spread and tail behavior. *LSTPM* and *DM-locallong* exhibit the lowest memorization, with compact distributions concentrated near small exposure values, whereas *Graph-flashback* displays a much heavier tail and substantial mass beyond exposure 5. This is noteworthy given its very small model capacity (10-dimensional embeddings and hidden states), suggesting that reduced representational power does not prevent strong memorization effects. Even minor architectural changes can shift memorization: *LSTM-simple* and *LSTM-long* share similar shapes, yet extending the temporal input from 24h to 72h noticeably increases the upper tail of the distribution, indicating that longer input horizons can raise memorization pressure. The *Markov* baseline falls mid-range between neural models, demonstrating that memorization can arise even in simple models.

Adapted vs. original exposure. In addition, we reproduce the original exposure protocol by Carlini et al. [4] as a benchmark: one synthetic "canary" trajectory is inserted into the training set, while 10,000 others—generated by uniformly sampling locations with equal probability—serve as references during testing. This yields a model-level exposure score based on how strongly the model ranks the true canary over the references. However, as shown in Figure 5, these Carlini scores (dashed lines) do not consistently align with user-level memorization. In some cases (e.g., *LSTPM*), the Carlini score sits near the lower end of the user-level distribution, whereas in others (e.g., *LSTM-simple*, *DM-locallong*), it exceeds real trajectory scores substantially.

Summary: *Memorization depends jointly on architectural design, temporal context, and model capacity. That is, small hyperparameter changes can shift memorization, and small models memorize as well. Moreover, while Carlini et al. [4]'s synthetic canary metric captures extreme cases, it does not align with how real user trajectories are memorized across architectures.*

6.4 Utility and Extractability

We finally investigate how individual-level memorization impacts two downstream dimensions of model behavior: generalization



(a) Prefix-based extractability. **(b) Anchor-based extractability.**

Figure 7: Extractability vs. Memorization (Magnitude) for *LSTM-simple* on the *Shanghai Telecom* dataset. Each violin shows the distribution of memorization magnitudes across extractability levels; width indicates density and central markers denote medians and quartiles statistics. Lower extractability corresponds to higher vulnerability.

performance (utility) and the risk of privacy leakage through extraction (extractability). In this work, we focus on extractability, where an adversary aims to recover sensitive trajectory information from partial observations. This setting captures a strong and practical form of privacy leakage, as it reflects the model's ability to reconstruct previously unseen portions of a user's behavior.

This perspective differs from membership inference, where the adversary is assumed to have access to a complete trajectory and seeks to determine whether it was part of the training set. While the attack objectives differ, both settings rely on a common underlying signal, namely, the model's tendency to assign disproportionately high likelihood to memorized data. Our analysis therefore provides insight into privacy risks beyond a specific attack instantiation, revealing that memorization is closely entangled with how the model learns and what it can leak:

Memorization harms utility. For each training trajectory, we compute the model's average top-1 accuracy on its corresponding reference set, capturing how well the model generalizes to unseen data. [16, Figure 6] and [16, Figure 16] reveal a consistent negative correlation between memorization (magnitude) and top-1 accuracy across datasets, indicating that higher memorization aligns with poorer generalization. Interestingly, we observe that memorization arises from two opposing mechanisms. In high-diversity settings like *YJMob100K*, where 80% of users have a diversity score above 0.95 (cf. [16, Figure 8c]), the model struggles to identify recurring patterns and defaults to memorizing trajectories, explaining both the low accuracy range (10–35%) and the high magnitudes (0 to 7.5). By contrast, in more structured datasets, such as *ShenzhenUrb*, memorization targets more users with low diversity, as shown in Figure 3a. Here, the model captures highly learnable patterns, even when generalization would suffice.

Summary: *Together, these findings highlight that memorization is not simply a byproduct of overfitting, but reflects both data complexity and the model's inductive biases.*

Memorization amplifies extractability. Finally, we assess the risks posed by memorization through two realistic extraction proxies,

each simulating a plausible inference scenario. Full details on attack design and metric formulation are provided in [16, Appendix D].

(a) Prefix-based attack. For this attack, we assume the adversary has partial knowledge of a user's trajectory and seeks to infer the remaining portion. We simulate three levels of attacker knowledge—prefixes of 6h, 12h, and 24h—and decode the rest of the 72h-long trajectory using a perplexity-weighted tree search. Extractability is measured as the logarithm of the number of greedy attempts required to retrieve the ground-truth suffix within the decoding tree. As shown in Figure 7a, trajectories with higher memorization magnitude tend to be easier to extract across prefix lengths, although the strength of this relationship varies across datasets (see [16, Figure 17]).

(b) Anchor-based attack. In this attack, we align with the anchor-pair memorization setting, where an adversary knows one key location and attempts to infer its counterpart. In our simulation, we assume the attacker knows the user's true home location and generates a synthetic trajectory initialized with stationary movement from 0–6 a.m. During the assumed commuting window (6–8 a.m.), we perform beam search to expand the trajectory. We then evaluate extractability over the subsequent work hours (9–17h) by ranking the true work location at each step and averaging these ranks across time. Figure 7b shows that users with higher anchor-pair memorization tend to be more vulnerable to work location inference in the *ShanghaiTel* dataset, although this trend may be weaker across other datasets (see [16, Figure 17]).

Summary: *We find that memorized samples act as inference shortcuts, surfacing more quickly during an attack. While these attacks serve as proxies, they approximate realistic adversarial capabilities and highlight concrete privacy risks.*

7 Discussion and Implications

Our results demonstrate that memorization in mobility models is neither random nor negligible. We discuss below key implications for privacy risk assessment and mitigation, as well as the framework scalability.

Memorization is predictable and user-specific. Our analyses show that memorization risk is not evenly distributed but disproportionately affects users with either highly regular or highly unique patterns. This observation unlocks new possibilities for mitigation. Rather than uniformly regularizing the dataset, one could adopt targeted training interventions, such as noise injection or dropout, applied more aggressively to memorization-prone users. Furthermore, model-level memorization was found to only partially reflect user-level retention in our experiments. This underscores the need for diverse audits using different metrics when deploying mobility models in sensitive contexts.

Memorization covers behavioral patterns. Our findings show that memorization in mobility models is not limited to reproducing exact locations from the training set. Models also retain behavioral structure, such as routine fragments or anchor pairs. These forms of retention are not captured by traditional privacy evaluations that focus on exact memory leakage only. Our analysis therefore supports improved privacy assessments that account for behavioral memorization, where patterns indicative of personal habits are unintentionally stored in mobility models.

Memorization exposes users to inference risks. We show that memorization correlates with how easily information can be extracted during inference, making our framework a useful early-warning tool for privacy vulnerabilities. By identifying trajectories that models treat as unusually likely, we can anticipate where attacks may succeed before they are carried out. This connection is particularly clear when considering membership inference attacks. Such attacks rely on distinguishing whether a trajectory belongs to the training set by comparing its likelihood to that of non-training samples drawn from a similar distribution. In our framework, memorization explicitly captures this likelihood gap: each training trajectory is evaluated against a reference set of behaviorally similar but unseen trajectories, and high memorization corresponds to a significant deviation in likelihood. As a result, highly memorized trajectories are expected to be more easily distinguishable from non-training data, making them inherently more vulnerable to membership inference. This establishes a direct link between memorization and inference risk, based on the same statistical signal.

Cross-user memorization. Our findings in Section 6.2 show that memorization increases with repetitiveness and stationarity, indicating that locations appearing more frequently within a trajectory are more likely to be retained by the model. This observation suggests a broader effect: locations that are frequently observed—either within a single trajectory or across multiple users—are inherently easier for the model to memorize. For instance, popular places (e.g., transport hubs or event venues) may be learned more strongly due to their repeated occurrence in the data. However, such cross-user memorization does not necessarily imply the same level of privacy risk as user-specific memorization. While frequently visited locations may be memorized at the population level, privacy risks arise when these locations can be associated with an individual's trajectory or routine. Overall, analyzing cross-user location memorization, and disentangling global popularity from user-specific patterns, is a promising direction for future work.

Scalability of the framework. The proposed framework evaluates memorization using a representative subset of trajectories rather than requiring exhaustive processing of the full dataset. In our setup, 2,000 medoid trajectories are selected via clustering, capturing diverse mobility behaviors and enabling an efficient approximation of memorization at the dataset level. The main computational cost arises from the reference set construction phase, which includes trajectory abstraction, clustering, and dataset generation. Importantly, this step is performed only once per dataset, and the resulting reference sets can be reused across different models. Consequently, reference set construction cost is primarily driven by dataset size and abstraction complexity (see [16, Table 5]. In contrast, memorization assessment is lightweight (see [16, Table 6]). While perplexity computation depends on the underlying model and dominates the evaluation runtime, the memorization metrics themselves require only a few seconds across all datasets and models. This makes the proposed framework practical for large-scale evaluation, as the additional overhead beyond standard model inference is negligible.

8 Conclusion and Perspective

Unintended memorization is a pervasive problem in mobility prediction models. In our experiments, we show that memorized data is not a simple side effect of overfitting, but arises depending on user behavior and model design. By examining three memorization types (location level, anchor pair, and segment level memorization), we demonstrate that the models unintentionally store sensitive parts of people's movements in their parameters, ranging from personal locations to fragments of daily routes. Unfortunately, these memorized patterns are easy to extract, revealing a concrete privacy risk in current prediction systems.

Our framework provides the first systematic means for assessing this risk, yet it should not be interpreted as the ultimate approach to auditing privacy in mobility models. The abstractions we study and the clustering method we use represent only one possible way to measure memorization, given the varying granularity of privacy leakage. That is, they constitute a best-effort design for quantifying this behavior and should thus rather serve as a template for similar analyses in future research.

Overall, our findings echo a broader privacy problem in generative modeling. Whenever sensitive data are used to train generative models, there is a risk that parts of the training data are memorized and can be exposed by adversaries. This problem is inherent to generative learning, which must strike a balance between capturing general patterns and relying on specific examples in the training data. The resulting risk is especially pronounced for mobility models, which are widely used in practice yet rely on highly sensitive location data. We hope that our framework provides a new practical tool for assessing this risk in deployed systems.

Acknowledgments

Generative AI tools (ChatGPT by OpenAI) were used solely for minor editorial assistance, including improving text flow and correcting typographical and grammatical issues.

This work was supported by the European Research Council (ERC) under the Consolidator Grant MALFOY (Grant Agreement No. 101043410) and by the German Federal Ministry of Research, Technology and Space (BMFTR) under the grant AIgenCY (16KIS2012). The work of Anne Josiane Kouam was supported through the Mob Sci-Dat Factory project (ANR-23-PEMO-0004) under the France 2030 program.

References

[1] L. Amichi, A. C. Viana, M. Crovella, and A. A.F. Loureiro. 2020. Understanding individuals' proclivity for novelty seeking. In *ACM SIGSPATIAL*. doi:10.1145/3397536.3422248

[2] Akanksha Atrey, Prashant J. Shenoy, and David D. Jensen. 2021. Preserving Privacy in Personalized Models for Distributed Mobile Services. In *41st IEEE International Conference on Distributed Computing Systems, ICDCS 2021, Washington DC, USA, July 7-10, 2021*. IEEE, 875–886. doi:10.1109/ICDCS51616.2021.00088

[3] Kunlin Cai, Jinghuai Zhang, Zhiqing Hong, William Shand, Guang Wang, Desheng Zhang, Jianfeng Chi, and Yuan Tian. 2024. Where Have You Been? A Study of Privacy Risk for Point-of-Interest Recommendation. In *Proceedings of the 30th ACM Conference on Knowledge Discovery and Data Mining* (Barcelona, Spain) *(KDD '24)*. ACM, New York, NY, USA, 175–186. doi:10.1145/3637528.3671758

[4] N. Carlini, C. Liu, Ú. Erlingsson, J. Kos, and D. Song. 2019. The Secret Sharer: Evaluating and Testing Unintended Memorization in Neural Networks. In *28th USENIX Security Symposium (USENIX Security 19)*. 267–284.

[5] N. Carlini, F. Tramèr, E. Wallace, M. Jagielski, A. Herbert-Voss, K. Lee, A. Roberts, T. Brown, D. Song, Ú. Erlingsson, A. Oprea, and C. Raffel. 2021. Extracting Training Data from Large Language Models. In *30th USENIX Security Symposium (USENIX Security 21)*. 2633–2650.

[6] A. Gobezie Chekol and M. Sintayehu Fufa. 2022. A survey on next location prediction techniques, applications, and challenges. *EURASIP Journal on Wireless Communications and Networking* 2022, 1 (2022), 29. doi:10.1186/s13638-022-02114-6

[7] dataplor. 2025. Unlocking the Power of Mobility Data: Insights for Localized Strategies. https://www.dataplor.com/resources/blog/mobility-location-data/. Blog post, *dataplor*, 19 May 2025.

[8] Yves-Alexandre de Montjoye, César A. Hidalgo, Michel Verleysen, and Vincent D. Blondel. 2013. Unique in the Crowd: The Privacy Bounds of Human Mobility. *Scientific Reports* 3, 1 (2013), 1376. doi:10.1038/srep01376

[9] 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* (Lyon, France) *(WWW '18)*. 1459–1468. doi:10.1145/3178876.3186058

[10] Hana Gebrie, Hasan Farooq, and Ali Imran. 2019. What Machine Learning Predictor Performs Best for Mobility Prediction in Cellular Networks?. In *2019 IEEE International Conference on Communications Workshops (ICC Workshops)*. 1–6. doi:10.1109/ICCW.2019.8756972

[11] M. C. González, C. A. Hidalgo, and A. Barabási. 2008. Understanding individual human mobility patterns. *Nature* (2008). doi:10.1038/nature06958

[12] Xiangming Gu, Chao Du, Tianyu Pang, Chongxuan Li, Min Lin, and Ye Wang. 2025. On Memorization in Diffusion Models. arXiv:2310.02664 [cs.LG]

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

[14] Renhe Jiang, Xuan Song, Zipei Fan, Tianqi Xia, Quanjun Chen, Qi Chen, and Ryosuke Shibasaki. 2018. Deep ROI-Based Modeling for Urban Human Mobility Prediction. 2, 1, Article 14 (March 2018), 29 pages. doi:10.1145/3191746

[15] Renhe Jiang, Xuan Song, Zipei Fan, Tianqi Xia, Quanjun Chen, Satoshi Miyazawa, and Ryosuke Shibasaki. 2018. DeepUrbanMomentum: an online deep-learning system for short-term urban mobility prediction. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence (AAAI'18/IAAI'18/EAAI'18)*. Article 96.

[16] Anne Josiane Kouam, Hristo Boyadzhiev, and Konrad Rieck. 2026. Secrets Everywhere: Auditing Memorization in Mobility Prediction Models. arXiv:2608.02052 [cs.LG] https://arxiv.org/abs/2608.02052

[17] Nicky Kriplani, Minh Pham, Gowthami Somepalli, Chinmay Hegde, and Niv Cohen. 2025. SolidMark: Evaluating Image Memorization in Generative Models. arXiv:2503.00592 [cs.LG]

[18] Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. 2022. Deduplicating Training Data Makes Language Models Better. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. ACL, Dublin, Ireland, 8424–8445. doi:10.18653/v1/2022.acl-long.577

[19] Sijia Liu, Yuanshun Yao, Jinghan Jia, Stephen Casper, Nathalie Baracaldo, Peter Hase, Yuguang Yao, Chris Yuhao Liu, Xiaojun Xu, Hang Li, Kush R. Varshney, Mohit Bansal, Sanmi Koyejo, and Yang Liu. 2025. Rethinking machine unlearning for large language models. *Nature Machine Intelligence* 7, 2 (2025), 181–194. doi:10.1038/s42256-025-00985-0

[20] Massimiliano Luca, Gianni Barlacchi, Bruno Lepri, and Luca Pappalardo. 2021. A Survey on Deep Learning for Human Mobility. *ACM Comput. Surv.* 55, 1, Article 7 (Nov. 2021), 44 pages. doi:10.1145/3485125

[21] E. M. R. Oliveira, A. C. Viana, C. Sarraute, J. Brea, and I. Alvarez-Hamelin. 2016. On the regularity of human mobility. *Pervasive and Mobile Computing* (2016). doi:10.1016/j.pmcj.2016.04.005

[22] Mexwell. 2020. Telecom Shanghai Dataset. https://www.kaggle.com/datasets/mexwell/telecom-shanghai-dataset. Accessed: 2025-11-13.

[23] Ministerie van Infrastructuur en Waterstaat. 2023. *Exploring new mobility policy-making pathways: Contrasting predict-and-provide and decide-and-provide*. Technical Report TNO Publiek R12676. Dutch Ministry of Infrastructure and Water Management. https://www.government.nl/binaries/government/documenten/reports/2023/12/19/exploring-new-mobility-policy-making-pathways-contrasting-predict-and-provide-and-provide-and-decide/Exploring+new+mobility+policy-making+pathways+-+Contrasting+predict-and-provide+and+provide-and-decide.pdf

[24] James R Norris. 1998. *Markov chains*. Number 2. Cambridge university press.

[25] Apostolos Pyrgelis, Carmela Troncoso, and Emiliano De Cristofaro. 2018. Knock Knock, Who's There? Membership Inference on Aggregate Location Data. In *25th Annual Network and Distributed System Security Symposium, NDSS 2018, San Diego, California, USA, February 18-21, 2018*. The Internet Society.

[26] Shaojie Qiao, Dayong Shen, Xiaoteng Wang, Nan Han, and William Zhu. 2015. A Self-Adaptive Parameter Selection Trajectory Prediction Approach via Hidden Markov Models. *Trans. Intell. Transport. Syst.* 16, 1 (Jan. 2015), 284–296. doi:10.1109/TITS.2014.2331758

[27] Yuanyuan Qiao, Zhongwei Si, Yanting Zhang, Fehmi Ben Abdesslem, Xinyu Zhang, and Jie Yang. 2018. A hybrid Markov-based model for human mobility prediction. *Neurocomputing* 278 (2018), 99–109. Recent Advances in Machine

Algorithm 1 Cluster Enrichment for Reference Set Balancing

Require: Trajectory records with cluster label and abstraction vector; cluster medoids; target minimum reference size $k_{\min}$; distance threshold τ

$k_{\min}$ *defines a target minimum size; reaching it depends on* τ *and the data distribution*
Build mapping from each cluster to its member trajectories
Build mapping from each medoid to its abstraction vector
Compute inter-cluster distances between medoid abstractions
for each cluster C **do**
    Select medoid trajectory T_u in C as training sample
    Initialize reference set $\mathcal{R}_C \leftarrow C \setminus \{T_u\}$
    if $|\mathcal{R}_C| < k_{\min}$ **then**
        Order other clusters by increasing medoid distance to C
        for each neighboring cluster C' in this order **do**
            for each trajectory $T' \in C'$ sorted by distance to T_u **do**
                if $d(T', T_u) \leq \tau$ **then**
                    Add T' to $\mathcal{R}_C$
                end if
                if $|\mathcal{R}_C| \geq k_{\min}$ **then**
                    break out of both loops
                end if
            end for
        end for
    end if
    Set $\tilde{\mathcal{R}}(T_u) \leftarrow \mathcal{R}_C$
end for
return training set $\{T_u\}$ and reference sets $\{\tilde{\mathcal{R}}(T_u)\}$

Learning for Non-Gaussian Data Processing. doi:10.1016/j.neucom.2017.05.101
[28] Xuan Rao, Lisi Chen, Yong Liu, Shuo Shang, Bin Yao, and Peng Han. 2022. Graph-Flashback Network for Next Location Recommendation. In *Proceedings of the 28th ACM Conference on Knowledge Discovery and Data Mining* (Washington DC, USA) *(KDD '22)*. ACM, New York, NY, USA, 1463–1471. doi:10.1145/3534678.3539383
[29] Ragil Saputra, Suprapto, and Agus Sihabuddin. 2024. Mobility Prediction Using Markov Models: A Survey. In *2024 7th International Conference on Informatics and Computational Sciences (ICICoS)*. 508–513. doi:10.1109/ICICoS62600.2024.10636860
[30] Libo Song, D. Kotz, Ravi Jain, and Xiaoning He. 2006. Evaluating Next-Cell Predictors with Extensive Wi-Fi Mobility Data. *IEEE Transactions on Mobile Computing* 5, 12 (2006), 1633–1649. doi:10.1109/TMC.2006.185
[31] 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. *Proceedings of the AAAI Conference on Artificial Intelligence* 34, 01 (Apr. 2020), 214–221. doi:10.1609/aaai.v34i01.5353
[32] D. Teixeira, J. Almeida, and A. C. Viana. 2021. On estimating the predictability of human mobility: the role of routine. *EPJ Data Science* (2021). doi:10.1140/epjds/s13688-021-00304-8
[33] TensorFlow Authors. 2021. TensorFlow Privacy. https://github.com/tensorflow/privacy. Includes exposure metric implementation for Secret Sharer privacy tests.
[34] Uber Technologies. 2018. H3: A Hexagonal Hierarchical Geospatial Indexing System. https://h3geo.org. Accessed: 2026-04-27.
[35] Xinglei Wang, Meng Fang, Zichao Zeng, and Tao Cheng. 2024. Where Would I Go Next? Large Language Models as Human Mobility Predictors. arXiv:2308.15197 [cs.AI]
[36] Jiaheng Wei, Yanjun Zhang, Leo Yu Zhang, Ming Ding, Chao Chen, Kok-Leong Ong, Jun Zhang, and Yang Xiang. 2025. Memorization in Deep Learning: A Survey. *ACM Comput. Surv.* 58, 4, Article 98 (Oct. 2025), 35 pages. doi:10.1145/3769076
[37] Takahiro Yabe, Kota Tsubouchi, Toru Shimizu, Yoshihide Sekimoto, Kaoru Sezaki, Esteban Moro, and Alex Pentland. 2023. *YJMob100K: City-Scale and Longitudinal Dataset of Anonymized Human Mobility Trajectories*. doi:10.5281/zenodo.10142719
[38] Gaelle M. Yonga, Anne J. Kouam, Aline C. Viana, and Auguste V. Noumsi. 2025. On Assessing Usability and Reliability of Anonymized Spatio-Temporal Data. In *2025 21st International Conference on Distributed Computing in Smart Systems and the Internet of Things (DCOSS-IoT)*. 689–696. doi:10.1109/DCOSS-IoT65416.2025.00107
[39] Desheng Zhang, Juanjuan Zhao, Fan Zhang, and Tian He. 2014. Description for Urban Data Release V2. https://people.cs.rutgers.edu/~dz220/data.html. Accessed: 2025-03.
[40] Desheng Zhang, Juanjuan Zhao, Fan Zhang, and Tian He. 2015. UrbanCPS: a cyber-physical system based on multi-source big infrastructure data for heterogeneous model integration. In *Proceedings of the ACM/IEEE Sixth International Conference on Cyber-Physical Systems*. ACM, Seattle Washington, 238–247. doi:10.1145/2735960.2735985
[41] Bob Zheng, Dhruv Ghulati, Manoj Panikkar, and Michael (Yichuan) Cai. 2025. Forecasting Models to Improve Driver Availability at Airports. https://www.uber.com/en-FR/blog/forecasting-models-to-improve-availability-at-airports/. Blog post, *Uber Engineering*, 19 Aug 2025.
[42] Yu Zheng. 2015. Trajectory Data Mining: An Overview. *ACM Trans. Intell. Syst. Technol.* 6, 3, Article 29 (May 2015), 41 pages. doi:10.1145/2743025
[43] Xiaofeng Zhong, Yinfeng Xiang, Fang Yi, Chao Li, and Qinmin Yang. 2024. HMP-LLM: Human Mobility Prediction Based on Pre-trained Large Language Models. In *2024 IEEE 4th International Conference on Digital Twins and Parallel Intelligence (DTPI)*. 687–692. doi:10.1109/DTPI61353.2024.10778764

A Open Science

To support the results presented in this paper and enable reproducibility, we release the full experimental artifact used in our evaluation. The artifact includes the data preprocessing pipeline, reference-set construction procedures, model training and evaluation code, memorization auditing scripts, and result analysis tools.

Datasets. Our experiments rely exclusively on publicly available mobility datasets:

- **Shanghai Telecom**. Available from Kaggle at [22]. The dataset consists of 12 files (XLSX format) containing anonymized cellular activity records, all of which are used in our experiments.
- **YJMob100K**. Available from Zenodo at [37]. We use the file `yjmob100k-dataset1.csv.gz`, which corresponds to the standard mobility period.
- **Shenzhen Urban**. Publicly available at [39]. From the datasets released under *Urban Data Release v2*, we use only the Call Detail Record data. Other data modalities provided on the same page (e.g., smart card, taxi GPS, bus GPS) are not used in this work.

The raw datasets are not redistributed. The artifact provides detailed instructions to download the data from their official sources and to reproduce the preprocessing steps described in the paper.

Repository access. The complete artifact is available at the following repository:

`gitlab.inria.fr/mobleak/mobleak-predictive`

Released artifact. The released artifact contains: (i) preprocessing notebooks implementing all data transformations and trajectory construction steps; (ii) scripts to generate training sets and reference sets for the three memorization abstractions studied (location-level, anchor-pair, and segment-level); (iii) implementations or adapted wrappers for all evaluated mobility prediction models; (iv) an automated pipeline to compute trajectory-level memorization metrics across models and datasets; and (v) analysis scripts to reproduce the figures and empirical findings reported in the paper.

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009