Where Have You Been? A Study of Privacy Risk for Point-of-Interest Recommendation

Kunlin Cai
kunlin96@g.ucla.edu
University of California, Los Angeles
Los Angeles, USA

Jinghuai Zhang
jinghuai1998@g.ucla.edu
University of California, Los Angeles
Los Angeles, USA

Zhiqing Hong
zh252@cs.rutgers.edu
Rutgers University
New Brunswick, USA

William Shand
wss2ec@g.ucla.edu
University of California, Los Angeles
Los Angeles, USA

Guang Wang
guang@cs.fsu.edu
Florida State University
Tallahassee, USA

Desheng Zhang
desheng@cs.rutgers.edu
Rutgers University
New Brunswick, USA

Jianfeng Chi
jianfengchi@meta.com
Meta
New York, USA

Yuan Tian
yuant@ucla.edu
University of California, Los Angeles
Los Angeles, USA

ABSTRACT

As location-based services (LBS) have grown in popularity, more human mobility data has been collected. The collected data can be used to build machine learning (ML) models for LBS to enhance their performance and improve overall experience for users. However, the convenience comes with the risk of privacy leakage since this type of data might contain sensitive information related to user identities, such as home/work locations. Prior work focuses on protecting mobility data privacy during transmission or prior to release, lacking the privacy risk evaluation of mobility data-based ML models. To better understand and quantify the privacy leakage in mobility data-based ML models, we design a privacy attack suite containing data extraction and membership inference attacks tailored for point-of-interest (POI) recommendation models, one of the most widely used mobility data-based ML models. These attacks in our attack suite assume different adversary knowledge and aim to extract different types of sensitive information from mobility data, providing a holistic privacy risk assessment for POI recommendation models. Our experimental evaluation using two real-world mobility datasets demonstrates that current POI recommendation models are vulnerable to our attacks. We also present unique findings to understand what types of mobility data are more susceptible to privacy attacks. Finally, we evaluate defenses against these attacks and highlight future directions and challenges.

CCS CONCEPTS

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

Correspondence to Kunlin Cai, Yuan Tian, and Jianfeng Chi. Work unrelated to Meta.

KDD '24, August 25–29, 2024, Barcelona, Spain

ACM ISBN 979-8-4007-0490-1/24/08
https://doi.org/10.1145/3637528.3671758

KEYWORDS

POI recommendation; privacy-preserving machine learning; data extraction; membership inference

ACM Reference Format:
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 SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '24), August 25–29, 2024, Barcelona, Spain.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3637528.3671758

1 INTRODUCTION

With the development and wide usage of mobile and wearable devices, large volumes of human mobility data are collected to support location-based services (LBS), such as traffic management [3, 34], store location selection [38], and point-of-interest (POI) recommendation [58, 69]. In particular, POI recommendation involves relevant POI suggestions to users for future visits based on personal preferences using ML techniques [27], which has recently gained much research attention[1]. POI recommendation models have also been integrated into popular services such as Yelp and Google Maps to assist users in making informed decisions about the next destination to visit. However, mobility data collected to train POI recommendation models are highly sensitive as they can leak users' sensitive information such as their social relationships, trip purposes, and identities [4].

Although there are a significant number of studies [2, 21, 32, 54] on mobility data privacy, the existing research primarily focuses on analyzing attacks and evaluations within the context of mobility data transmission and release processes. For example, previous studies have demonstrated the linkages of mobility data from various side channels, including social networks [24, 26], open-source datasets [20, 44], and network packets [29, 61]. The linkages between these side channels can lead to the identification of individuals. As a result, efforts to protect mobility data have primarily

[1]From 2017 to 2023, there are more than 111 papers on POI recommendation built upon mobility data collected by location service providers [64].

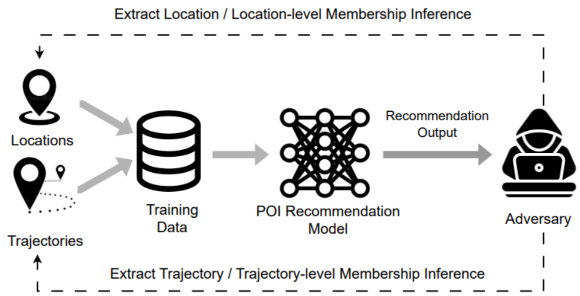


Figure 1: Our attack suite highlights the privacy concerns in POI recommendation models. In particular, we demonstrate that an adversary can extract or infer membership information of locations or trajectories in the training dataset.

concentrated on data aggregations and releases [5, 21, 41]. These studies neglect the risk of adversaries extracting sensitive attributes or properties from the ML models (e.g., POI recommendation models) that use mobility data for training, which are inherently susceptible to privacy attacks [6, 52].

Evaluating privacy risks in POI recommendation models remains challenging because existing attack and defense mechanisms are ineffective due to the unique features of mobility data. Previous privacy attacks have mainly focused on ML models trained with image and text data [8, 18, 52], where each data point can uniquely identify itself. However, mobility data, such as locations, are less semantically unique without the context. Moreover, mobility data is special in that it contains multimodal spatial and temporal information, which describes each individual's movements and behavior patterns over time. All existing attacks fail to construct meaningful context and leverage spatial-temporal information, resulting in their failures when applied to POI recommendations. Furthermore, existing defense mechanisms [1, 50, 51] have mainly been tested on classification models trained with image or text data. Given the task and data are significantly different, the effectiveness of defense mechanisms is unknown when applied to POI recommendation.

In this paper, we design a comprehensive privacy attack suite to study the privacy leakage in POI recommendation models trained with mobility data. Specifically, our privacy attack suite contains the two most popular kinds of privacy attacks on machine learning models, data extraction and membership inference attacks, to assess the privacy vulnerabilities of POI recommendation models at both **location** and **trajectory** levels. In contrast to privacy attacks for image and text data, the attacks in our attack suite are tailored for mobility data and aim to extract different types of sensitive information based on practical adversary knowledge.

We perform experiments on three representative POI recommendation models trained on two mobility datasets. We demonstrate that POI recommendation models are vulnerable to our designed data extraction and membership inference attacks. We further provide an in-depth analysis to understand what factors affect the attack performance and contribute to the effectiveness of the attacks. Based on our analysis, we discover that the effect of data outliers exists in privacy attacks against POI recommendations, making training examples with certain types of users, locations, and trajectories particularly vulnerable to the attacks in the attack suite. Finally, We test several existing defenses and find that they do not effectively thwart our attacks with negligible utility loss, which calls for better methods to defend against our attacks.

Contributions:

- We introduce a novel privacy attack suite[2] that incorporates unique characteristics of mobility data (e.g., spatial-temporal information) into the attack design. In particular, we target a previously under-defended attack surface: neural-network-based POI recommendation. To the best of our knowledge, our work is the first to comprehensively evaluate the privacy risks in POI recommendation models using inference attacks from both location and trajectory levels.
- We conduct extensive experiments on state-of-the-art POI recommendation models and datasets to demonstrate that POI recommendation models are vulnerable to data extraction and membership inference attacks in our attack suite.
- We provide an in-depth analysis to understand what unique factors in mobility data make them vulnerable to privacy attacks. We also explore the reason regarding how our attack design works and test existing defenses against our attacks. Our analysis identifies the challenges and future directions for developing privacy-preserving POI recommendation models.

2 BACKGROUND

2.1 Point-of-Interest Recommendation

POI recommendation has recently gained much attention due to its importance in many business applications [27], such as user experience personalization and resource optimization. Initially, researchers focused on feature engineering and algorithms such as Markov chain [10, 71], matrix factorization algorithms [11, 36], and Bayesian personalized ranking [25, 72] for POI recommendation. However, more recent studies have shifted their attention towards employing neural networks like RNN [37, 67], LSTM [31, 58], and self-attention models [35, 39]. Neural networks can better learn from spatial-temporal correlation in mobility data (e.g., check-ins) to predict users' future locations and thus outperform other POI recommendation algorithms by a large margin. Meanwhile, this could introduce potential privacy leakage. Thus, we aim to design an attack suite to measure the privacy risks of neural-network-based POI recommendations systematically.

We first provide the basics of POI recommendations and notations used throughout this paper. Let $\mathcal{U}$ be the user space, $\mathcal{L}$ be the location space, and $\mathcal{T}$ be the timestamp space. A POI recommendation model takes the observed trajectory of a user as input and predicts the next POI that will be visited, which is formulated as $f_\theta : \mathcal{U} \times \mathcal{L}^n \times \mathcal{T}^n \rightarrow \mathbb{R}^{|\mathcal{L}|}$. Here, the length of the input trajectory is n. We denote a user by its user ID $u \in \mathcal{U}$ for simplicity. For an input trajectory with n check-ins, we denote its *trajectory sequence* as $x_T^{0:n-1} = \{(l_0, t_0), \ldots, (l_{n-1}, t_{n-1})\}$, where $l_i \in \mathcal{L}$ and $t_i \in \mathcal{T}$ indicate the POI location and corresponding time interval of i-th check-in. Also, the *location sequence* of this trajectory is denoted as $x_L^{0:n-1} = \{l_0, \ldots, l_{n-1}\}$. The POI recommendation model predicts the next location l_n (also denoted as y by convention) by outputting the logits of all the POIs. Then, the user can select the POI with the highest logit as its prediction $\hat{y}$, where $\hat{y} = \arg\max f_\theta(u, x_T^{0:n-1})$. Given the training set D_{tr} sampled from

[2]Our code is publicly available at: https://github.com/KunlinChoi/POIPrivacy

Table 1: A summary of the threat model.

Attack	Adversary Objective	Adversary Knowledge
LocExtract	Extract the most frequently visited location l of a target user u	–
TrajExtract	Extract the location sequence of a target user u with length n: $x_L = \{l_0, \ldots, l_{n-1}\}$	Starting location l_0
LocMIA	Infer the membership of a user-location pair (u,l)	Shadow dataset D_s
TrajMIA	Infer the membership of a trajectory sequence $x_T = \{(l_0, t_0), \ldots, (l_n, t_n)\}$	Shadow dataset D_s

an underlying distribution $\mathcal{D}$, the model weights are optimized to minimize the prediction loss on the overall training data, i.e., $\min_\theta \frac{1}{|D_{\text{tr}}|} \sum_{(u,x_T^{0:n-1},y)\in D_{\text{tr}}} \ell(f_\theta(u, x_T^{0:n-1}), y)$, where ℓ is the cross-entropy loss, i.e., $\ell(f_\theta(u, x_T^{0:n-1}), y) = -\log(f_\theta(u, x_T^{0:n-1}))_y$. The goal of the training process is to maximize the performance of the model on the unseen test dataset $D_{te} \in \mathcal{D}$, which is drawn from the same distribution as the training data. During inference, this prediction $\hat{y}$ is then compared to the next real location label l_n to compute the prediction accuracy. The performance evaluation of POI recommendation models typically employs metrics such as top-k accuracy (e.g., $k = 1, 5, 10$).

2.2 Threat Models

Adversary Objectives. To understand the potential privacy leakage of training data in POI recommendation models, we design the following four attacks from the two most common privacy attack families: membership inference attack [53] and data extraction attacks [9], based on the characteristics of the mobility data for POI recommendation, namely *common location extraction* (LocExtract), *training trajectory extraction* (TrajExtract), *location-level membership inference attack* (LocMIA), and *trajectory-level membership inference attack* (TrajMIA). These four attacks aim to extract or infer different sensitive information about a user in the POI recommendation model training data.

LocExtract focuses on extracting a user's most frequently visited location; TrajExtract extracts a user's location sequence with a certain length given a starting location; LocMIA infers whether a user has been to a location and used for training; TrajMIA infers where a trajectory sequence has been used for training. The summary of the threat model is outlined in Table 1.

Adversary Knowledge. For all attacks, we assume the attacker has access to the query interface of the victim model. Specifically, the attacker can query the victim model with the target user and obtain the corresponding output logits. This assumption is realistic in two scenarios: (1) A malicious third-party entity is granted access to the POI model query API hosted by the model owner (e.g., location service providers like Foursquare or Yelp) for specific businesses such as personalized advertisement. This scenario is well-recognized by [42, 55, 66]. (2) The retention period of the training data expires. Still, the model owner keeps the model and an adversary (e.g., a malicious insider of location service providers) can extract or infer the sensitive information using our attack suite, even if the training data have been deleted. In this scenario, the model owner may violate privacy regulations such as GDPR [15].

Depending on different attack objectives, the adversary also possesses different auxiliary knowledge. In particular, for TrajExtract, we assume the attacker can query the victim model with a starting location l_0 that the target user visited. This assumption is reasonable because an attacker can use real-world observation [56, 60], LocExtract, and LocMIA as cornerstones. As for LocMIA and TrajMIA, we assume the attacker has access to a shadow dataset following the standard settings of membership inference attacks [6, 52].

3 ATTACK SUITE

Our attack suite is used to evaluate privacy vulnerabilities of POI recommendation models at both location and trajectory levels. The subsequent sections detail the technical approaches and design of attacks, taking into account the unique aspects of mobility data.

3.1 Data Extraction Attacks

Our data extraction attacks are rooted in the idea that victim models display varying levels of memorization in different subsets of training data. By manipulating the spatial-temporal information in the queries, the attacker can extract users' locations or trajectories that these victim models predominantly memorize.

LocExtract. Common location extraction attack (LocExtract) aims to extract a user's most frequently visited location in the victim model training, i.e.,

$$\textsc{LocExtract}(f_\theta, u) \rightarrow \hat{l}_{top1}, \ldots, \hat{l}_{topk}.$$

The attack takes the victim model f_θ and the target user u as the inputs and generates k predictions $\hat{l}_{top1}, \ldots, \hat{l}_{topk}$ to extract the most frequently visited location of user u. The attack is motivated by our key observation: querying POI recommendation models with a random location reveals that these models tend to "over-learn" a user's most frequently visited locations, making these locations more likely to appear in the model output. For example, we randomly choose 10 users and query the victim model using 100 randomly selected locations. Of these queries, 32.5% yield the most frequent location for the target user. Yet, these most common locations are present in only 18.7% of these users' datasets.

In LocExtract, we first generate a set of different random inputs for a specific user and use them to make iterative queries to the victim model. Each query returns the prediction logits with a length of $|\mathcal{L}|$ outputted by the victim model. The larger the logit value, the more confident the model is in predicting the corresponding location as the next POI. Therefore, by iterating queries to the model given a target user and aggregating the logit values of all queries, the most visited location is more likely to have a large logit value after aggregation. In particular, we use a soft voting mechanism, i.e., averaging the logits of all the queries. With the resulting mean logits, we output the top-k locations with k largest logit values as the attack results. Algorithm 1 outlines LocExtract. Though the attack is straightforward, it is effective and can be a stepping stone for TrajExtract in our attack suite.

TrajExtract. Our training trajectory extraction attack (TrajExtract) aims to extract the location sequence $x_L^{0:n-1} = \{l_0, \ldots, l_{n-1}\}$ in a training trajectory of user u with a length of n from the victim model f_θ. Formally,

$$\textsc{TrajExtract}(f_\theta, u, l_0, n) \rightarrow \hat{x}_{L_0}^{0:n-1}, \ldots, \hat{x}_{L_\beta}^{0:n-1},$$

Algorithm 1 Common Location Extraction Attack

Input: Victim model: f_θ, target user: u, query budget: q, query timestamp: t, output size: k

Output: Top-k predictions: $[\hat{l}_{top1},\ldots,\hat{l}_{topk}]$

logits $\leftarrow \{\}$
for q times **do**
  $l \leftarrow$ RandomSample($\mathcal{L}$) ▷ *Randomly generate a location from the location space*
  logits $\cup f_\theta(u, \{(l,t)\})$
end for
logits$_{\text{agg}}$ = Aggregate(logits) ▷ *Aggregate confidence for all locations*
return $\hat{l}_{top1},\ldots,\hat{l}_{topk} \leftarrow \text{Argmax}_k(\text{logits}_{\text{agg}})$

where $\hat{x}^{0:n-1}_{L_0},\ldots,\hat{x}^{0:n-1}_{L_\beta}$ indicate the top-β extracted location sequences by the attack.

The key idea of the training trajectory extraction attack is to identify the location sequence with the lowest log perplexity, as models tend to demonstrate lower log perplexity when they see trained data. We denote log perplexity as:

$$\text{PPL}_{f_\theta}(u, x_T^{0:n-1}) = -\log \Pr_{f_\theta}(u, x_T^{0:n-1}) = -\sum_{i=0}^{n-1} \log \Pr_{f_\theta}(u, x_T^{0:i-1}),$$

where $\Pr_{f_\theta}(\cdot)$ is the likelihood of observing $x_T^{0:n-1}$ with user u under the victim model f_θ. In order to get the lowest log perplexity of location sequences with a length of n, we have to enumerate all possible location sequences. However, in the context of POI recommendation, there are $O(|\mathcal{L}|^{n-1})$ possible location sequences for a given user. $|\mathcal{L}|$ equals the number of unique POIs within the mobility dataset and can include thousands of options. Thus, the cost of calculating the log perplexity of all location sequences can be very high. To this end, we use beam search to extract the location sequences with both time and space complexity $O(|\mathcal{L}|\times n \times \beta)$, where β is the beam size. In particular, to extract a trajectory of length n, we iteratively query the victim model using a set of candidate trajectories with a size of β and update the candidate trajectories until the extraction finishes. As highlighted in the prior work [16], when using beam search to determine the final outcome of a sequential neural network, there is a risk of generating non-diverse outputs and resembling the training data sequence. However, in our scenario, this property can be leveraged as an advantage in TrajExtract, as our primary objective revolves around extracting the training location sequence with higher confidence. As a final remark, both LocExtract and TrajExtract need a query timestamp to query the victim model, and we will show the effects of the timestamp in our experiments. Algorithm 2 gives the detailed steps of TrajExtract.

3.2 Membership Inference Attacks

Membership inference attack (MIA) aims to determine whether a target data sample is used in the model training. We extend the notion to infer whether certain sensitive information (e.g., user-location pair (u,l) and trajectory sequence (u, x_T)) of the user's data is involved in the training of the victim model f_θ. Since POI

Algorithm 2 Training Trajectory Extraction Attack

Input: Victim model: f_θ, target user: u, starting location: l_0, target extraction length: n, query timestamp: t, beam width: β

Output: Top-β possible extraction results: $\hat{x}^{0:n}_{L_0},\ldots,\hat{x}^{0:n}_{L_\beta}$

for $b \leftarrow 0$ to $\beta - 1$ **do**
  $\hat{x}^{0:0}_{T_b} \leftarrow (u, (l_0, t))$ ▷ *Initialize the beam with l_0 and t*
end for
for $i \leftarrow 1$ to $n-1$ **do**
  for $\hat{x}^{0:i-1}_T$ in $\{\hat{x}^{0:i-1}_{T_0},\ldots,\hat{x}^{0:i-1}_{T_\beta}\}$ **do**
    $\{\hat{x}^{0:i}_{T_0},\ldots,\hat{x}^{0:i}_{T_\beta}\} \leftarrow \text{UpdateBeam}_\beta(f_\theta(u, \hat{x}^{0:i-1}_T))$ ▷ *Update the beam by keeping β trajectory with the smallest PPL from the query output and current beam*
  end for
end for
$\hat{x}^{0:n-1}_{L_0},\ldots,\hat{x}^{0:n-1}_{L_\beta} \leftarrow \text{GetLoc}(\hat{x}^{0:n-1}_{T_0},\ldots,\hat{x}^{0:n-1}_{T_\beta})$ ▷ *Take the location sequence from $\hat{x}^{0:n-1}_T$ as result $\hat{x}^{0:n-1}_L$*
return $\hat{x}^{0:n-1}_{L_0},\ldots,\hat{x}^{0:n-1}_{L_\beta}$

recommendation models use multi-modal sequential data as inputs and adversaries lack sufficient information to construct a complete input, we propose attack designs to manipulate the spatial-temporal information in queries to enhance effectiveness of attacks. The membership inference attack can be formulated as follow:

$$\text{MIA}(f_\theta, X_{target}, D_s) \rightarrow \{\text{member}, \text{nonmember}\},$$

where X_{target} represents the target sensitive information ($X_{target} = (u,l)$ in LocMIA and $X_{target} = (u, x_T)$ in TrajMIA), and D_s is the shadow dataset owned by the adversary.

To effectively infer the membership of a given X_{target}, we adapt the state-of-the-art membership inference attack – likelihood ratio attack (LiRA) [6] to the context of POI recommendation. The key insight of LiRA is that the model parameters trained with X_{target} differ from those trained without it, and the effect of the model parameter on a data sample can be well approximated using a loss value. By conducting a hypothesis test on the distributions of the loss values, we can identify if the victim model is trained with the X_{target} or not. LiRA consists of four steps: (1) train multiple shadow models, (2) query the shadow models trained with X_{target} and without X_{target} to obtain two distributions, (3) query the victim model X_{target} to obtain the output logits, and (4) conduct a Λ hypothesis test to infer the membership of the X_{target} based on the two distributions and the query results.

LocMIA. In this attack, the adversary aims to determine whether a given user u has visited a location l in the training data. However, it is not feasible to directly apply LiRA to LocMIA as the victim model takes the trajectory sequences as inputs, but the adversary only has a target location without the needed sequential context. In particular, LocMIA needs the auxiliary inputs to calculate the membership confidence score since this process cannot be completed only using $X_{target} = (u,l)$. This attack is a stark contrast to MIA for image/text classification tasks where the X_{target} itself is sufficient to compute the membership confidence score.

To this end, we design a spatial-temporal model query algorithm (Algorithm 3) to tailor LiRA to LocMIA and optimize membership

Algorithm 3 SpaTemQuery: Spatial-Temporal Model Query Algorithm for LocMIA

Input: Target model: f_{target}, number of query timestamps: n_t, number of query locations: n_l, target example: X_{target}
Output: Membership confidence score: $conf$

$u, l \leftarrow X_{target}$
$conf_{all} \leftarrow \{\}$
for $i \leftarrow 0$ to $n_t - 1$ **do**
  $conf_t \leftarrow \{\}$
  for $j \leftarrow 0$ to $n_l - 1$ **do**
    $t_i \leftarrow i/n_t$
    $l_j \leftarrow \text{RandomSample}(\mathcal{L})$
    $conf_t \leftarrow conf_t \cup f_{target}(u, (l_j, t_i))$ ▷ *Query the model with random location and a synthetic timestamp*
  end for
  $conf_{all} \leftarrow conf_{all} \cup \text{mean}(conf_t)$ ▷ *Calculate average confidence from all queries for this timestamp*
end for
return $conf \leftarrow \max(conf_{all})$ ▷ *Take the confidence scores with largest confidence at position l as output*

Algorithm 4 Membership Inference Attack

Below, we demonstrate our location-level MIA and trajectory-level MIA algorithms. The lines marked in red are specific to LocMIA, while the lines marked in purple are specific to TrajMIA. Both attacks share the remaining lines.

Input: Victim model: f_θ, shadow data: D_s, number of shadow models: N, inference target: X_{target}, number of query timestamps: n_t, number of query locations: n_l
Output: The likelihood ratio to determine if we should reject the hypothesis that X_{target} is a member of f_θ: Λ

$conf_{in}, conf_{out} \leftarrow \{\},\{\}$
$X_S \leftarrow \text{RandomSample}(\{X_S : X_{target} \in X_S\})$ ▷ *Sample a location sequence and includes X_{target}*
$X_S \leftarrow X_{target}$
for $i \leftarrow 0$ to N **do**
  $D_{in} \leftarrow \text{RandomSample}(D_s) \cup X_S$
  $D_{out} \leftarrow \text{RandomSample}(D_s) \setminus X_S$
  $f_{in}, f_{out} \leftarrow \text{Train}(D_{in}), \text{Train}(D_{out})$ ▷ *Train f_{in} and f_{out}*
  $conf_{in} \leftarrow conf_{in} \cup \phi(\text{SpaTemQuery}(f_{in}, n_t, n_l, X_{target}))$
  $conf_{out} \leftarrow conf_{out} \cup \phi(\text{SpaTemQuery}(f_{out}, n_t, n_l, X_{target}))$
  $conf_{in} \leftarrow conf_{in} \cup \phi\left(\text{mean}(\{f_{in}(X_S)^{0:0}, \ldots, f_{in}(X_S)^{0:n-1}\})\right)$
  $conf_{out} \leftarrow conf_{out} \cup \phi\left(\text{mean}(\{f_{out}(X_S)^{0:0}, \ldots, f_{out}(X_S)^{0:n-1}\})\right)$
end for
$\mu_{\text{in}}, \mu_{\text{out}} \leftarrow \text{mean}(conf_{\text{in}}), \text{mean}(conf_{\text{out}})$
$\sigma^2_{\text{in}}, \sigma^2_{\text{out}} \leftarrow \text{var}(conf_{\text{in}}), \text{var}(conf_{\text{out}})$
$conf_{\text{obs}} \leftarrow \phi(\text{SpaTemQuery}(f_\theta, n_t, n_l, X_{target}))$
$conf_{\text{obs}} \leftarrow \phi\left(\text{mean}(\{f_\theta(X_S)^{0:0}, \ldots, f_\theta(X_S)^{0:n-1}\})\right)$
return $\Lambda = \frac{p(conf_{\text{obs}} \mid \mathcal{N}(\mu_{\text{in}}, \sigma^2_{\text{in}}))}{p(conf_{\text{obs}} \mid \mathcal{N}(\mu_{\text{out}}, \sigma^2_{\text{out}}))}$ ▷ *Hypothesis test*

confidence score calculation. The idea behind the algorithm is that if a particular user has been to a certain POI location, the model might "unintentionally" memorize its neighboring POI locations and the corresponding timestamp in the training data. Motivated by this, each time we query the models (e.g., the victim and shadow models), we generate n_l random locations and n_t fixed-interval timestamps. To obtain stable and precise membership confidence scores, we first average the corresponding confidence scores at the target location by querying with n_l locations at the same timestamp. While the adversary does not possess the ground truth timestamp linked with the target POI for queries, the adversary aims to mimic a query close to the real training data. To achieve this, we repeat the same procedure of querying different locations for n_t timestamps and take the maximum confidence scores among the n_t averaged confidence scores as the final membership inference score for the target example. Algorithm 4 gives the outline of LiRA in terms of LocMIA, and the lines marked with red are specific to LocMIA.

TrajMIA. The attack aims to determine whether a trajectory is used in the training data of the victim model. Unlike LocMIA, $X_{target} = (u, x_T)$ suffices to calculate the membership confidence score in LiRA, and we do not need any auxiliary inputs. To fully leverage information of the target example querying the victim model and improve the attack performance, we also utilize the $n-2$ intermediate outputs and the final output from the sequence x_T with a length of n to compute the membership confidence score, i.e., we take the average of all $n-1$ outputs. This change improves the attack performance as the intermediate outputs provide additional membership information for each point in the target trajectory. The purple lines in Algorithm 4 highlight steps specific to TrajMIA.

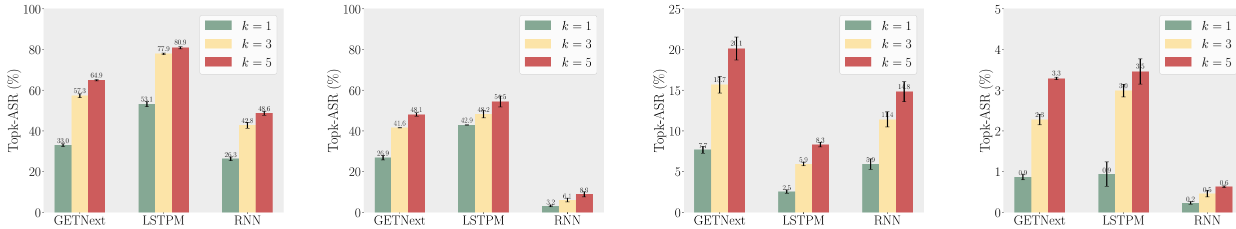
3.3 Practical Implications of the Attack Suite

Our attack suite is designed as an integrated framework focusing on the basic units of mobility data – locations and trajectories. It contains two prevalent types of privacy attacks: data extraction and membership inference attacks. Each attack in our attack suite targets a specific type of mobility data and could serve as a privacy auditing tool [28]. They can also be used to infer additional sensitive information in mobility data:

- LocExtract extracts a user's most common location. Combined with the semantics of the POI, we may infer the user's address such as work address, which is closely related to user identity;
- TrajExtract can be further used to infer user trajectories and identify trip purposes by analyzing the POIs visited during a journey [40];
- LocMIA can determine the membership of multiple POIs, thereby facilitating the inference of a user's activity range and social connections in Cho et al. [12], Ren et al. [48];
- TrajMIA infers if a user's trajectory is in the training dataset, which can serve as an auditing tool to examine the privacy leakage by assuming a worst-case adversary.

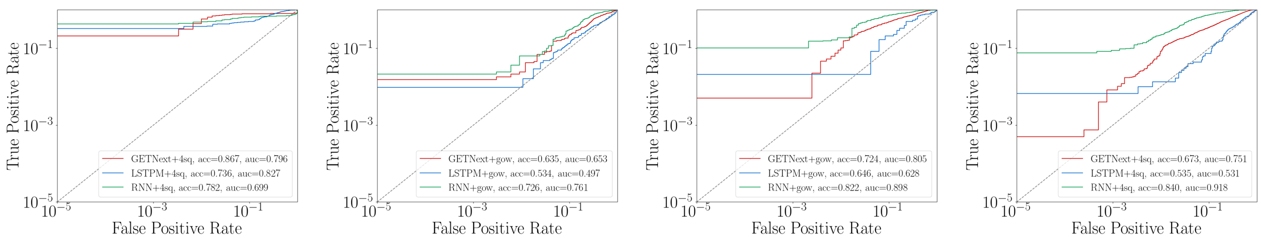
4 EXPERIMENTS

We empirically evaluate the proposed attack suite to answer the following research questions: (1) What's the performance of the proposed attacks in extracting or inferring the sensitive information from POI recommendation models (Sec. 4.2.1)? (2) What unique factors (e.g., user, location, trajectory) in mobility data correlate



(a) LocExtract (4sq) (b) LocExtract (Gowalla) (c) TrajExtract (4sq) (d) TrajExtract (Gowalla)

Figure 2: Main results of data extraction attacks (LocExtract and TrajExtract) on three victim models and two datasets.



(a) LocMIA (4sq) (b) LocMIA (Gowalla) (c) TrajMIA (4sq) (d) TrajMIA (Gowalla)

Figure 3: Main results of membership inference attacks (LocMIA and TrajMIA) on three victim models and two mobility datasets. The diagonal line indicates the random guess baseline.

Table 2: The performance of victim models.

Dataset	Model	Top-1 ACC	Top-10 ACC
4sq	GETNext	0.34	0.71
	LSTPM	0.25	0.67
	RNN	0.24	0.68
Gowalla	GETNext	0.16	0.48
	LSTPM	0.15	0.39
	RNN	0.10	0.26

with the attack performance (Sec. 4.2.2)? (3) How do different attack designs (e.g., spatial-temporal querying for membership inference attacks) improve the attack performance (Sec. 4.2.3)?

4.1 Experimental Setup

We briefly describe the datasets, models, and evaluation metrics used in our experiments. Due to the space limit, we defer the details of datasets (e.g., statistics of each dataset), data pre-processing pipeline, default training and attack parameters to Appendix A.

Datasets. *Following the literature* [31, 69], we comprehensively evaluate four privacy attacks on two POI recommendation benchmarks: FourSquare (4sq) [68] and Gowalla [12]. We use the check-ins collected in NYC for both sources.

Models. We experiment with three representative POI recommendation models, including GETNext[3] [69], LSTPM[4] [58], and RNN [63]. Note that GETNext and LSTPM are the state-of-the-art POI recommendation methods based on the transformer and hierarchical LSTM, respectively.

Evaluation Metrics. We use the top-k extraction attack success rate (ASR) to evaluate the effectiveness of data extraction attacks. For LocExtract, the top-k ASR is defined as $|U_{\text{extracted}}|/|\mathcal{U}|$, where $U_{\text{extracted}}$ is the set of users whose most visited locations are in the top-k predictions outputted by our attack; For TrajExtract the top-k ASR is |correct extractions|/|all (u, l_0) pairs|, where correct extractions are (u, l_0) pairs with top-k extracted results matching an exact location sequence in the training data.

For LocMIA and TrajMIA, we utilize the commonly employed metrics for evaluating membership inference attacks, namely the area under the curve (AUC), average-case "accuracy" (ACC), and true positive rate (TPR) versus false positive rate (FPR) in the low-false positive rate regime. Our primary focus is the TPR versus FPR metric in the low-false positive rate regime because evaluating membership inference attacks should prioritize the worst-case privacy setting rather than average-case, as emphasized in [6].

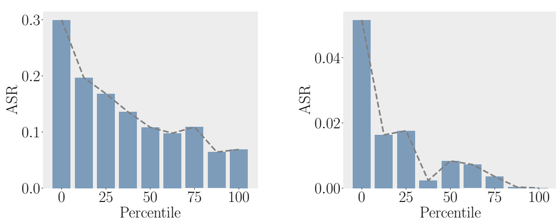
4.2 Experimental Results and Analysis

4.2.1 Attack performance (RQ1). Figures 2 and 3 visualize the attack performance of data extraction and membership inference attacks, respectively. In Figure 2, we observe that LocExtract and TrajExtract can effectively extract users' most common locations and trajectories across various model architectures and datasets as the attack performance is significantly better than the random guess baseline, i.e., $1/|\mathcal{L}|$ (0.04% for LocExtract) and $1/|\mathcal{L}|^{n-1}$(10^{-8}% for TrajExtract). Likewise, as shown in Figure 3, LocMIA and TrajMIA successfully determine the membership of a specific user-location pair or trajectory, significantly outperforming the random guess baseline (represented by the diagonal line in both figures).

The attack performance also demonstrates that trajectory-level attacks are significantly more challenging than location-level attacks, evident from the better performance of LocExtract and LocMIA compared to TrajExtract and TrajMIA for data extraction and membership inference. We suspect this is because POI recommendation models are primarily designed to predict a single location. In contrast, our trajectory-level attacks aim to extract or infer a trajectory encompassing multiple consecutive locations.

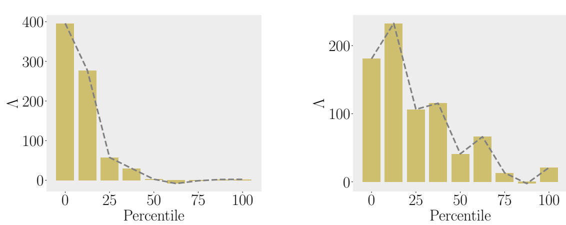
[3] https://github.com/songyangme/GETNext
[4] https://github.com/NLPWM-WHU/LSTPM

The attack performance also differs across different model architectures and datasets. We see a general trend of privacy-utility trade-off in POI recommendation models based on the model performance of the victim model in Table 2: with better victim model performance comes better attack performance. While this is a common trend, it might not hold in some cases. For example, the MIA performance against RNN is sometimes better than GETNext and LSTPM performances. This might be because GETNext and LSTPM improve upon RNN by better leveraging spatial-temporal information in the mobility datasets. However, the adversary cannot use the exact spatial-temporal information in shadow model training since the adversary cannot access that information. This result can be inspiring in that even though spatial-temporal information can effectively improve attack performance, victim models that better utilize spatial-temporal information are still more resilient to MIAs. Future studies should also consider this characteristic when designing attacks or privacy-preserving POI recommendation models with better privacy-utility trade-offs.



(a) # Unique POIs (4sq) (b) # Unique POIs (Gowalla)

Figure 4: How user-level aggregate statistics are related to TrajExtract. Users who have fewer unique POIs are more vulnerable to TrajExtract.



(a) # of Users Visited (b) # of Nearby Check-ins

Figure 5: How location-level aggregate statistics are related to LocMIA. Locations visited by fewer different users or have fewer surrounding check-ins are more vulnerable to LocMIA.

4.2.2 Factors in mobility data that make it vulnerable to the attacks (RQ2). Prior research demonstrates that data outliers are the most vulnerable examples to privacy attacks [6, 59] in image and text datasets. However, it is unclear whether the same conclusion holds in mobility data and what makes mobility data as data outliers. To this end, we investigate which factors of the mobility datasets influence the attack's efficacy. In particular, we collect aggregate statistics of mobility data from three perspectives: user, location, and trajectory. We analyze which factors in these three categories make mobility data vulnerable to our attacks. We defer the details of selecting the aggregate statistics and the list of selected aggregate statistics in our study in Appendix A. Our findings are as follows:

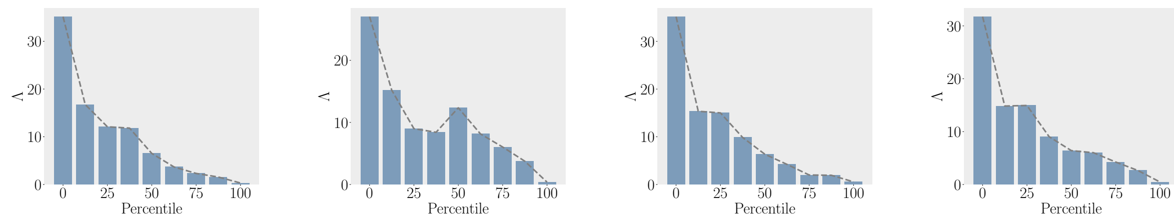
- For LocExtract, we do not identify any meaningful pattern correlated with its attack performance. We speculate that a user's most common location is not directly related to the aggregate statistics we study.
- For TrajExtract, our findings indicate that *users who have visited fewer unique POIs* are more vulnerable to this attack, as referenced in Figure 4. This can be explained by the fact that when users have fewer POIs, the model is less uncertain in predicting the next location due to the reduced number of possible choices that the model memorizes.
- For LocMIA, as shown in Figures 5(a) and 5(b), we find that *locations visited by fewer users or have fewer surrounding check-ins* are more susceptible to LocMIA. We believe this is because those locations shared with fewer users or surrounding check-ins make them training data outliers.
- For TrajMIA, *users with fewer total check-ins* (Figure 6(a)), *unique POIs* (Figure 6(b)), and *fewer or shorter trajectories* (Figures 6(c) and 6(d)) are more susceptible. In Figures 7(a) and 7(b), we also see that *trajectories intercepting less with others or with more check-ins* are more vulnerable to TrajMIA. We believe these user-level and trajectory-level aggregate statistics make the target examples data outliers.

In summary, we conclude that the effect of data outliers also exists in privacy attacks against POI recommendations. In the context of POI recommendation, the mobility data outliers could be characterized from the perspectives of user, location, and trajectory. Different attacks in our attack suite might be vulnerable to particular types of data outliers, which are more unique and are vulnerable against our attacks compared to other data.

4.2.3 The impacts of various attack designs (RQ3). We explore different attack designs that may affect the performance of the attack suite on the 4sq dataset. We summarize the key findings as follows: **The optimal query timestamp leads to better attack performance.** The significance of timestamps to our attack aligns with the fact that POI recommendation relies on temporal information for making accurate predictions. From Figure 8, we find that the extraction attacks (i.e.,LocExtract and TrajExtract) ASRs first increase and then decrease as the query timestamp increases, which peaks at 0.5 (the middle of the day). The reason is that most check-ins occur during the daytime rather than at night.

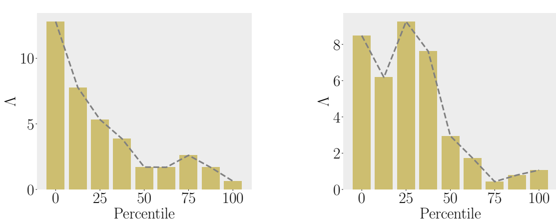
For LocExtract, from Figure 9(b), *we observed that utilizing more queries with different timestamps* in our spatial-temporal model query algorithm improves the results of inferring the membership of a target user-location pair (u, l). Since the adversary lacks information about real input sequences before the target location l, utilizing more queries helps to traverse the search space to approximate the correct timestamp and promotes the attack performance. **A small number of queries is sufficient for our attacks.** Figure 10 in the Appendix demonstrates that our data extraction attacks (LocExtract and TrajExtract) remain effective even with a limited number of queries. Specifically, a few queries (i.e., $q = 50$) and a small beam width (i.e., $\beta = 10$) allow the attacker to achieve high ASRs for LocExtract and TrajExtract, respectively. In other words, our attacks are practical in real-world scenarios.

Additionally, Figure 9(a) indicates that LocMIA remains effective even with a limited number of queries for different location choices



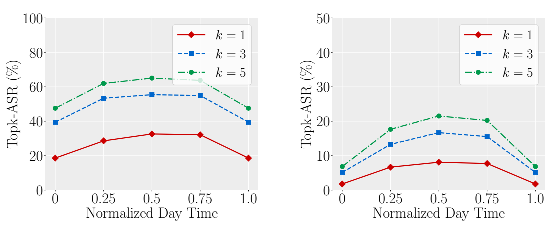
(a) # of User Check-ins (b) # of User POIs (c) # of User Trajectories (d) Avg User Traj Length

Figure 6: How user-level aggregate statistics are related to TrajMIA. *x-axis:* Percentile categorizes users/locations/trajectories into different groups according to their feature values. *y-axis:* Λ indicates the (averaged) likelihood ratio of training trajectories/locations being the member over non-member from the hypothesis test for each group, with a higher value indicating the larger vulnerability. The users with fewer total check-ins, fewer unique POIs, and fewer or shorter trajectories are more vulnerable to TrajMIA.



(a) # of Intercepting Trajs (b) # of POIs in Traj

Figure 7: How trajectory-level aggregate statistics are related to TrajMIA. The trajectories with fewer intercepting trajectories or fewer POIs are more vulnerable to TrajMIA.



(a) LocExtract (b) TrajExtract

Figure 8: The optimal query timestamp can improve the performance of LocExtract and TrajExtract.

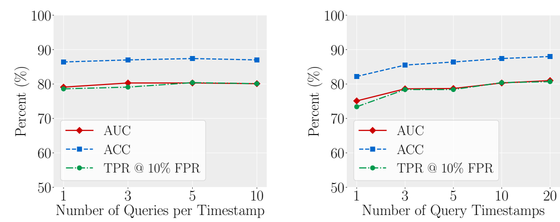
even with a small n_l (i.e., number of query locations) per timestamp, which yields better practicability of our attack.

A larger number of shadow models leads to better performance of membership inference attacks. Figure 11 in the Appendix shows that a larger number of shadow models greatly improves the performance of LocMIA and TrajMIA since more shadow models provide more samples to approximate the real loss distributions of in-samples and out-samples.

5 DEFENSE

We evaluate existing defenses against privacy attacks on machine learning models. Due to the limited space, we illustrate the defense mechanisms and the key findings in the main paper and defer experimental details to Appendix B.

Defense Techniques. In particular, we evaluate two streams of defense mechanisms on proposed attacks, including (1) standard



(a) Impact of n_l on LocMIA (b) Impact of n_t on LocMIA

Figure 9: Both the number of query timestamps n_t and the number of query locations n_l affect the performance of LocMIA. We use $n_t = 10$ and $n_l = 10$ in (a) and (b), respectively. LocMIA is effective with a limited number of queries.

techniques to reduce overfitting (e.g., early stopping, l_2 regularization) and (2) differential privacy-based defenses (e.g., DP-SGD [1]) for provable risk mitigation. The standard techniques reduce the victim model's memorization to some degree, but they are insufficient due to the lack of statistical guarantees. To fill this gap, differential privacy [14] is also used to defend against our attacks, which can theoretically limit the impact of a single data point on the model's performance.

Specifically, we first experiment with DP-SGD [1], the most representative DP-based defense, to train differentially-private POI recommendation models. The key idea of DP-SGD is to add Gaussian noises $\mathcal{N}(0, \sigma^2 C^2 I)$ to the clipped gradients g of the model during its training process. C is a clipping threshold that bounds the sensitivity of g by ensuring $\|g\| \leq C$. To achieve (ϵ, δ)-DP, we have $\sigma = \sqrt{2 \ln \frac{1.25}{\delta}} / \epsilon$. Despite that DP-SGD provides promising defense performance on language tasks [17], we find that it can substantially sacrifice the model's utility on the POI recommendation task. Specifically, the top-10 accuracy is only 4.97% when the mechanism satisfies $(5, 0.001)$-DP, while the original top-10 accuracy without DP is 71%. The reason for this performance decrease is that POI recommendation aims to make accurate user-level predictions within a large output space (i.e., $> 4,000$ possible POIs). For different users, even the same location sequence may lead to a different result, which means that the model needs to capture user-specific behavior patterns from a relatively small user dataset. As a result, the training is quite sensitive to the noises introduced by DP-SGD, making it not applicable to POI recommendations.

Table 3: The exposure of sensitive information in each attack.

Attack	Sensitive Information
LocExtract	Most common location of each user
TrajExtract	Each location sequence/sub-sequence (x_L)
LocMIA	Each user-location pair (u, l)
TrajMIA	Each trajectory sequence/sub-sequence (x_T)

However, we argue that DP-SGD provides undifferentiated protection for all the mobility data, while for POI recommendation, protecting more tailored sensitive information is more important. For example, a defender may only care about whether a list of check-ins about home addresses is protected or not. To this end, we introduce the notion of selective DP [51] to relax DP and improve the model's utility-privacy trade-offs. Specifically, we apply the state-of-the-art selective DP method JFT [50] to protect different levels of sensitive information for each attack. The key idea of JFT is to adopt a two-phase training process: in the phase-I training, JFT redacts the sensitive information in the training dataset and optimizes the model with a standard optimizer; in the phase-II training, JFT applies DP-SGD to finetune the model on the original dataset in a privacy-preserving manner. Due to the phase-I training, we observe that the model's utility is significantly promoted. In addition to JFT, we also apply Geo-Indistinguishability (Geo-Ind) [2] to protect common locations in LocExtract. We note that Geo-Ind is only applicable to LocExtract (but not LocMIA) because it requires modifying the training data and is incompatible with the notion of membership inference.

Takeaway Messages from the Defense. We evaluate different defense mechanisms in terms of their performance in preventing each attack from stealing the corresponding sensitive information. Table 3 summarizes the exposure of sensitive information in each attack. Besides, Appendix B illustrates our evaluation metrics, experimental setup, and the results. Recall that the check-ins within a mobility dataset are not equally important. Therefore, in our experiments, we comprehensively evaluate the defense mechanisms from two perspectives. Specifically, we measure their performance in *(1)* protecting all the sensitive information and *(2)* protecting a targeted subset of sensitive information from being attacked. Figures 12 and 13 in the Appendix show that existing defenses provide a certain degree of guarantee in mitigating privacy risks of ML-based POI recommendations, especially for the targeted subset of sensitive information. However, there is no such unified defense that can successfully defend against all the proposed attacks within a small utility drop. In other words, our exploration highlights the need for more advanced defenses.

6 RELATED WORK

Mobility Data Privacy. Mobility data contain rich information that can reveal individual privacy such as user identity. Previous work utilizes side-channel attacks to extract sensitive information about mobility data from LBS, including social relationships [44, 57], aggregated trajectories [45, 46, 70], trajectory history [22, 33], network packets [29, 61] and location embeddings [13]. Despite the focus of previous work, deep neural networks (DNN) built on large volumes of mobility data have recently become state-of-the-art backbones for LBS, opening a new surface for privacy attacks. To the best of our knowledge, our work is the first of its kind to investigate the vulnerabilities of DNN models in leaking sensitive information about mobility data using inference attacks. Moreover, previous defenses [30, 43, 47, 62, 65] primarily focus on data collection, aggregation, and publishing, which can not protect DNN models built on POI data.

Privacy Attacks. Various types of privacy attacks, such as membership inference attacks [6, 23, 49, 52], training data extraction attacks [7, 8], and model inversion attacks [19] have been proposed to infer sensitive information from model training data. Our attack suite contains membership inference and data extraction attacks. Existing data extraction and membership inference attacks [6, 8] are insufficient for POI recommendation models due to the spatio-temporal nature of the data. Our work takes the first step to extracting sensitive location and trajectory patterns from POI recommendation models and solving unique challenges to infer the membership of both user-location pairs and user trajectories. As a final remark, our attacks differ from previous MIAs in mobility data [45, 70], which focus on the privacy risks of data aggregation.

7 CONCLUSION

In this work, we take the first step to evaluate the privacy risks of the POI recommendation models. In particular, we introduce an attack suite containing data extraction attacks and membership inference attacks to extract and infer sensitive information about location and trajectory in mobility data. We conduct extensive experiments to demonstrate the effectiveness of our attacks. Additionally, we analyze what types of mobility data are vulnerable to the proposed attacks. To mitigate our attacks, we further adapt two mainstream defense mechanisms to the task of POI recommendation. Our results show that there is no single solid defense that can simultaneously defend against all proposed attacks. Our findings underscore the urgent need for better privacy-preserving approaches for POI recommendation models. Interesting future directions include: (1) Generalize the attack suite to measure privacy risks of real-world location-based services (e.g., Yelp/Google Maps) in a more challenging setting (e.g., label-only setting); (2) Develop more advanced defense mechanisms against our attacks.

Ethics Statement. This work introduces a novel attack suite on POI recommendation models trained on public anonymized mobility datasets with no personally identifiable information, aimed at bringing potential vulnerabilities in POI models to public attention. The success of our attacks offers insights into future privacy leakage measurement in learning-based models involving spatio-temporal data. Moreover, it underscores the need for improved defense solutions for POI models with better utility-privacy trade-offs. We hope our study fosters further research in protecting the privacy of mobility data.

ACKNOWLEDGMENTS

We sincerely thank the reviewers for their valuable feedback on the paper. This work is supported in part by the National Science Foundation (NSF) Awards 1951890, 1952096, 2003874, 2047822, 2317184, 2325369, 2411151, 2411152, 2411153, UCLA ITLP and Okawa foundation. Any opinions, findings, conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of sponsors.

REFERENCES

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep learning with differential privacy. ACM SIGSAC Conference on Computer and Communications Security.

[2] Miguel E Andrés, Nicolás E Bordenabe, Konstantinos Chatzikokolakis, and Catuscia Palamidessi. 2013. Geo-indistinguishability: Differential privacy for location-based systems. ACM SIGSAC Conference on Computer and Communications Security.

[3] Lei Bai, Lina Yao, Can Li, Xianzhi Wang, and Can Wang. 2020. Adaptive graph convolutional recurrent network for traffic forecasting. International Conference on Neural Information Processing Systems.

[4] Andrew J Blumberg and Peter Eckersley. 2009. On locational privacy, and how to avoid losing it forever. *Electronic frontier foundation* 10, 11 (2009), 1–7.

[5] Nicolás E Bordenabe, Konstantinos Chatzikokolakis, and Catuscia Palamidessi. 2014. Optimal geo-indistinguishable mechanisms for location privacy. ACM SIGSAC Conference on Computer and Communications Security.

[6] Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramer. 2022. Membership inference attacks from first principles. IEEE Symposium on Security and Privacy.

[7] Nicolas Carlini, Jamie Hayes, Milad Nasr, Matthew Jagielski, Vikash Sehwag, Florian Tramer, Borja Balle, Daphne Ippolito, and Eric Wallace. 2023. Extracting training data from diffusion models. 32nd USENIX Security Symposium (USENIX Security 23), 5253–5270.

[8] Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. 2019. The Secret Sharer: Evaluating and testing unintended memorization in neural networks. USENIX Security Symposium.

[9] Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. 2021. Extracting training data from large language models. 30th USENIX Security Symposium (USENIX Security 21), 2633–2650.

[10] Meng Chen, Yang Liu, and Xiaohui Yu. 2014. Nlpmm: A next location predictor with markov modeling. Advances in Knowledge Discovery and Data Mining: Pacific-Asia Conference.

[11] Chen Cheng, Haiqin Yang, Michael R Lyu, and Irwin King. 2013. Where you like to go next: Successive point-of-interest recommendation. International Joint Conference on Artificial Intelligence.

[12] Eunjoon Cho, Seth A Myers, and Jure Leskovec. 2011. Friendship and mobility: user movement in location-based social networks. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.

[13] Jiaxin Ding, Shichuan Xi, Kailong Wu, Pan Liu, Xinbing Wang, and Chenghu Zhou. 2022. Analyzing sensitive information leakage in trajectory embedding models. International Conference on Advances in Geographic Information Systems.

[14] Cynthia Dwork, Aaron Roth, et al. 2014. The algorithmic foundations of differential privacy. (2014).

[15] EU. 2018. General data protection regulation. https://en.wikipedia.org/wiki/General_Data_Protection_Regulation.

[16] Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. (2018).

[17] Natasha Fernandes, Mark Dras, and Annabelle McIver. 2019. Generalised differential privacy for text document processing. Springer International Publishing, Principles of Security and Trust: 8th International Conference, POST 2019, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2019, Prague, Czech Republic, April 6–11, 2019, Proceedings 8, 123–148.

[18] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. 2015. Model inversion attacks that exploit confidence information and basic countermeasures. ACM SIGSAC Conference on Computer and Communications Security.

[19] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. 2015. Model Inversion Attacks That Exploit Confidence Information and Basic Countermeasures. Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, New York, NY, USA, 1322–1333. https://doi.org/10.1145/2810103.2813677

[20] Sébastien Gambs, Marc-Olivier Killijian, and Miguel Núñez del Prado Cortez. 2014. De-anonymization attack on geolocated data. (2014).

[21] Bugra Gedik and Ling Liu. 2005. Location privacy in mobile systems: A personalized anonymization model. IEEE International Conference on Distributed Computing Systems.

[22] Philippe Golle and Kurt Partridge. 2009. On the anonymity of home/work location pairs. Springer, Pervasive Computing: 7th International Conference, Pervasive 2009, Nara, Japan, May 11-14, 2009. Proceedings 7, 390–397.

[23] Maziar Gomrokchi, Susan Amin, Hossein Aboutalebi, Alexander Wong, and Doina Precup. 2023. Membership Inference Attacks Against Temporally Correlated Data in Deep Reinforcement Learning. *IEEE Access* (2023).

[24] Wajih Ul Hassan, Saad Hussain, and Adam Bates. 2018. Analysis of privacy protections in fitness tracking social networks-or-you can run, but can you hide? USENIX Security Symposium.

[25] Jing He, Xin Li, and Lejian Liao. 2017. Category-aware next point-of-interest recommendation via listwise bayesian personalized ranking. International Joint Conference on Artificial Intelligence.

[26] Benjamin Henne, Christian Szongott, and Matthew Smith. 2013. SnapMe if you can: Privacy threats of other peoples' geo-tagged media and what we can do about it. ACM Conference on Security and Privacy in Wireless and Mobile Networks.

[27] Md. Ashraful Islam, Mir Mahathir Mohammad, Sarkar Snigdha Sarathi Das, and Mohammed Eunus Ali. 2020. A survey on deep learning based Point-Of-Interest (POI) recommendations. arXiv:cs.IR/2011.10187

[28] Matthew Jagielski, Jonathan Ullman, and Alina Oprea. 2020. Auditing differentially private machine learning: How private is private SGD? (2020).

[29] Tao Jiang, Helen J Wang, and Yih-Chun Hu. 2007. Preserving location privacy in wireless LANs. International Conference on Mobile Systems, Applications and Services.

[30] Yuzhou Jiang, Emre Yilmaz, and Erman Ayday. 2023. Robust Fingerprint of Privacy-Preserving Location Trajectories. *Proceedings on Privacy Enhancing Technologies* (2023).

[31] Dejiang Kong and Fei Wu. 2018. HST-LSTM: A hierarchical spatial-temporal long-short term memory network for location prediction. International Joint Conference on Artificial Intelligence.

[32] John Krumm. 2007. Inference attacks on location tracks. Springer, Pervasive Computing: 5th International Conference, PERVASIVE 2007, Toronto, Canada, May 13-16, 2007. Proceedings 5, 127–143.

[33] John Krumm. 2007. Inference attacks on location tracks. Springer, Pervasive Computing: 5th International Conference, PERVASIVE 2007, Toronto, Canada, May 13-16, 2007. Proceedings 5, 127–143.

[34] Shiyong Lan, Yitong Ma, Weikang Huang, Wenwu Wang, Hongyu Yang, and Pyang Li. 2022. Dstagnn: Dynamic spatial-temporal aware graph neural network for traffic flow forecasting. International Conference on Machine Learning.

[35] Defu Lian, Yongji Wu, Yong Ge, Xing Xie, and Enhong Chen. 2020. Geography-aware sequential location recommendation. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.

[36] Defu Lian, Cong Zhao, Xing Xie, Guangzhong Sun, Enhong Chen, and Yong Rui. 2014. GeoMF: Joint geographical modeling and matrix factorization for point-of-interest recommendation. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.

[37] Q. Liu, Shu Wu, Liang Wang, and Tieniu Tan. 2016. Predicting the next location: A recurrent model with spatial and temporal contexts. AAAI Conference on Artificial Intelligence.

[38] Yanchi Liu, Chuanren Liu, Xinjiang Lu, Mingfei Teng, Hengshu Zhu, and Hui Xiong. 2017. Point-of-Interest demand modeling with human mobility patterns. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.

[39] Yingtao Luo, Qiang Liu, and Zhaocheng Liu. 2021. Stan: Spatio-temporal attention network for next location recommendation. Web Conference.

[40] Chuishi Meng, Yu Cui, Qing He, Lu Su, and Jing Gao. 2017. Travel purpose inference with GPS trajectories, POIs, and geo-tagged social media data. IEEE International Conference on Big Data.

[41] Joseph Meyerowitz and Romit Roy Choudhury. 2009. Hiding stars with fireworks: location privacy through camouflage. Annual International Conference on Mobile Computing and Networking.

[42] Mike Boland. 2021. Foursquare's power play continues with relaunched places and new API. https://www.localogy.com/2021/03/foursquares-power-play-continues-with-relaunched-places-and-new-api/

[43] Àlex Miranda-Pascual, Patricia Guerra-Balboa, Javier Parra-Arnau, Jordi Forné, and Thorsten Strufe. 2023. SoK: Differentially private publication of trajectory data. *Proceedings on Privacy Enhancing Technologies* (2023).

[44] Jovan Powar and Alastair R Beresford. 2023. SoK: Managing risks of linkage attacks on data privacy. (2023).

[45] Apostolos Pyrgelis, Carmela Troncoso, and Emiliano De Cristofaro. 2017. Knock knock, who's there? Membership inference on aggregate location data. (2017).

[46] Apostolos Pyrgelis, Carmela Troncoso, and Emiliano De Cristofaro. 2020. Measuring membership privacy on aggregate location time-series. *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 4, 2 (2020), 1–28.

[47] Jinmeng Rao, Song Gao, Yuhao Kang, and Qunying Huang. 2020. LSTM-TrajGAN: A deep learning approach to trajectory privacy protection. arXiv preprint arXiv:2006.10521.

[48] Lingfei Ren, Ruimin Hu, Dengshi Li, Zheng Wang, Junhang Wu, Xixi Li, and Wenyi Hu. 2023. Who is your friend: inferring cross-regional friendship from mobility profiles. (2023).

[49] Ahmed Salem, Yang Zhang, Mathias Humbert, Pascal Berrang, Mario Fritz, and Michael Backes. 2018. Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models. (2018).

[50] Weiyan Shi, Si Chen, Chiyuan Zhang, Ruoxi Jia, and Zhou Yu. 2022. Just fine-tune twice: Selective differential privacy for large language models. (2022).

[51] Weiyan Shi, Aiqi Cui, Evan Li, Ruoxi Jia, and Zhou Yu. 2022. Selective Differential Privacy for Language Modeling. Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies.

[52] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership inference attacks against machine learning models. IEEE Symposium on

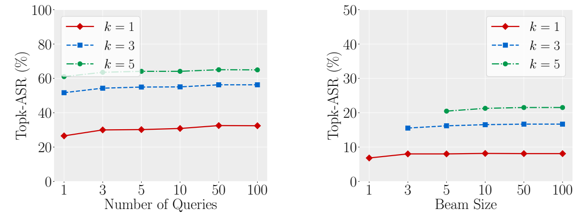
Security and Privacy.
[53] R. Shokri, M. Stronati, C. Song, and V. Shmatikov. 2017. Membership inference attacks against machine learning models. IEEE Symposium on Security and Privacy.
[54] Reza Shokri, George Theodorakopoulos, Panos Papadimitratos, Ehsan Kazemi, and Jean-Pierre Hubaux. 2013. Hiding in the mobile crowd: Location privacy through collaboration. (2013).
[55] Shubham Sharma. 2022. How Foursquare helps enterprises drive positive results with geospatial technology. https://venturebeat.com/data-infrastructure/how-foursquare-helps-enterprises/
[56] Mudhakar Srivatsa and Mike Hicks. 2012. Deanonymizing mobility traces: Using social network as a side-channel. ACM Conference on Computer and Communications Security.
[57] Mudhakar Srivatsa and Mike Hicks. 2012. Deanonymizing mobility traces: Using social network as a side-channel. ACM Conference on Computer and Communications Security.
[58] 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. AAAI Conference on Artificial Intelligence.
[59] Florian Tramèr, Reza Shokri, Ayrton San Joaquin, Hoang Le, Matthew Jagielski, Sanghyun Hong, and Nicholas Carlini. 2022. Truth serum: Poisoning machine learning models to reveal their secrets. ACM SIGSAC Conference on Computer and Communications Security.
[60] Carmen Ruiz Vicente, Dario Freni, Claudio Bettini, and Christian S Jensen. 2011. Location-related privacy in geo-social networks. (2011).
[61] Nevena Vratonjic, Kévin Huguenin, Vincent Bindschaedler, and Jean-Pierre Hubaux. 2014. A location-privacy threat stemming from the use of shared public IP addresses. (2014).
[62] Huandong Wang, Changzheng Gao, Yuchen Wu, Depeng Jin, Lina Yao, and Yong Li. 2023. PateGail: a privacy-preserving mobility trajectory generator with imitation learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*.
[63] Jingyuan Wang, Jiawei Jiang, Wenjun Jiang, Chao Li, and Wayne Xin Zhao. 2021. LibCity: An open library for traffic prediction. International Conference on Advances in Geographic Information Systems.
[64] Zehui Wang, Wolfram Höpken, and Dietmar Jannach. 2023. A survey on Point-of-Interest recommendations leveraging heterogeneous data. arXiv:cs.IR/2308.07426
[65] Zhibo Wang, Wenxin Liu, Xiaoyi Pang, Ju Ren, Zhe Liu, and Yongle Chen. 2020. Towards pattern-aware privacy-preserving real-time data collection. In *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*. IEEE.
[66] Haoran Xin, Xinjiang Lu, Tong Xu, Hao Liu, Jingjing Gu, Dejing Dou, and Hui Xiong. 2021. Out-of-town recommendation with travel intention modeling. arXiv:cs.IR/2101.12555
[67] Dingqi Yang, Benjamin Fankhauser, Paolo Rosso, and Philippe Cudre-Mauroux. 2020. Location prediction over sparse user mobility traces using rnns. International Joint Conference on Artificial Intelligence.
[68] Dingqi Yang, Daqing Zhang, Vincent W Zheng, and Zhiyong Yu. 2014. Modeling user activity preference by leveraging user spatial temporal characteristics in LBSNs. (2014).
[69] Song Yang, Jiamou Liu, and Kaiqi Zhao. 2022. GETNext: Trajectory flow map enhanced transformer for next POI recommendation. ACM SIGIR Conference on Research and Development in Information Retrieval.
[70] Guanglin Zhang, Anqi Zhang, and Ping Zhao. 2020. Locmia: Membership inference attacks against aggregated location data. (2020).
[71] Jia-Dong Zhang, Chi-Yin Chow, and Yanhua Li. 2014. Lore: Exploiting sequential influence for location recommendations. ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems.
[72] Shenglin Zhao, Tong Zhao, Haiqin Yang, Michael Lyu, and Irwin King. 2016. STELLAR: Spatial-temporal latent ranking for successive point-of-interest recommendation. AAAI Conference on Artificial Intelligence.

Table 4: Statistics of POI Recommendation Datasets.

	#POIs	#Check-ins	#Users	#Trajectories	Avg. Len.
4sq	4,556	63,648	1,070	17,700	3.63
Gowalla	2,559	32,633	1,419	7,256	4.46

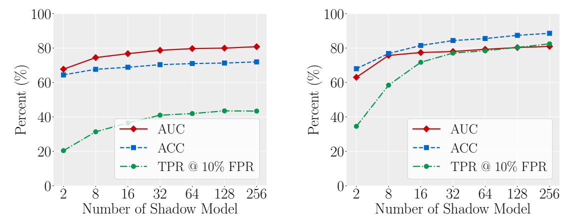
A EXPERIMENT DETAILS

Data Preprocessing. We preprocess each dataset following the literature [69]: (1) Filter out unpopular POIs and users that appear less than ten times.(2) Construct trajectories of different users in a daily manner (24-hours) (3) Normalize the timestamp (from 0:00 AM to 11:59 PM) in each check-in record into $[0, 1]$. After the aforementioned steps, the key statistics of the 4sq and Gowalla datasets are shown in Table 4. (4) Lastly, we split the datasets into the training, validation, and test sets using the ratio of 8:1:1. For victim model training, we use the official implementation of GETNext and LSTPM to train victim models. By default, we train each model with a batch size 32 for 200 epochs and use five random seeds in all experiments to report the average results.



(a) LocExtract (4sq) (b) TrajExtract (4sq)

Figure 10: Our LocExtract is effective with a small number of queries and TrajExtract is effective with a small beam size (i.e., both attacks are effective in a small query budget).

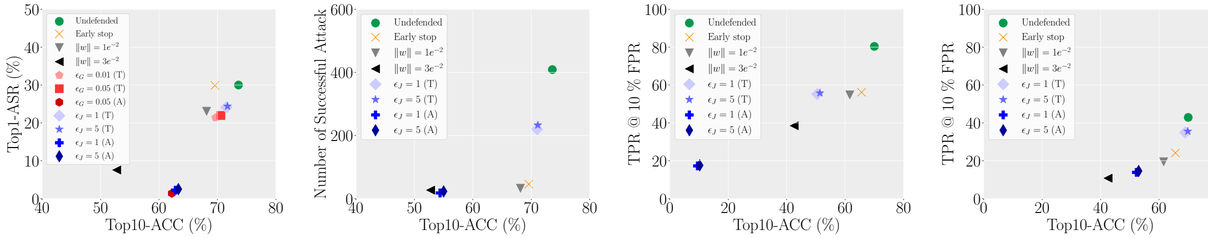


(a) LocMIA (4sq) (b) TrajMIA (4sq)

Figure 11: The attack performance of LocMIA and TrajMIA significantly improves with more shadow models.

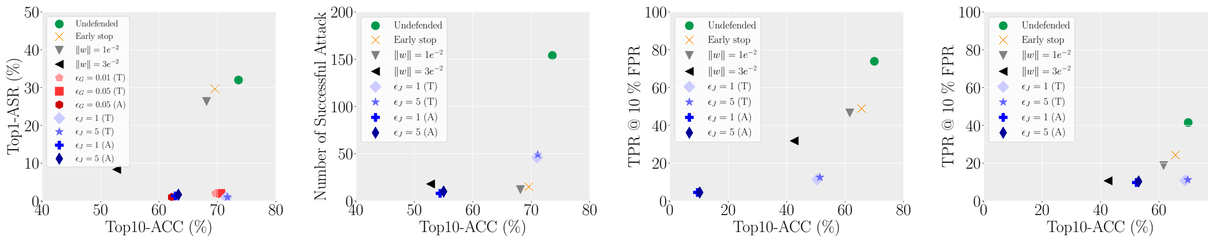
Attack Settings. (1) LocExtract Given a target user u, we extract the most visited location l_{top1} from the victim model f_θ with a query number $q = 50$. We set the query timestamp $t = 0.5$ (i.e., the middle of the day) by default, and discuss the effect of different query timestamps on the attack performance in the ablation study. (2) TrajExtract In this attack, we experiment with $n = 4$ by default, though the attacker can potentially extract location (sub-)sequences with arbitrary length. We set the beam size $\beta = 50$ and also have the default query timestamp $t = 0.5$. (3) LocMIA In our experiments, we randomly sample 80% of trajectories as the training dataset D_{tr} to build a victim model and treat the remaining 20% data as non-members. For each target user u and the POI location l pair, we generate $N = 64$ synthesis trajectories using TrajSynthesis with the query timestamp $t_s = 0.5$. With the synthesis trajectories, we can also have 64 in-models (f_{in}) and 64 out-models (f_{out}). We also set $n_t = 10$ and $n_l = 10$. (4) TrajMIA We extract the membership information of some trajectory sequences with arbitrary lengths from the victim model. We also build $N = 64$ in-models (f_{in}) and $N = 64$ out-models (f_{out}) for a target trajectory sequence. For evaluation, we conduct a hypothesis test on a balanced number of members and non-members for both LocMIA and TrajMIA.

How to Select Aggregate Statistics. This section outlines the basic principles and details for selecting representative aggregate statistics for analysis. For user-level aggregate statistics, we target the basic statistical information quantifying properties of locations and trajectories of a user: (u1) Total number of check-ins; (u2)



(a) LocExtract (b) TrajExtract (c) LocMIA (d) TrajMIA

Figure 12: Defense performance on protecting all corresponding sensitive information for each attack.



(a) LocExtract (b) TrajExtract (c) LocMIA (d) TrajMIA

Figure 13: Defense performance on protecting the targeted subset of sensitive information for each attack.

Number of unique visited POIs; (u3) Number of trajectories; (u4) Average trajectory length. For location-level statistics, we study their users, "neighboring" check-ins and trajectories, and the check-in time information. (l1) Number of users who have visited this POI; (l2) Number of check-ins surrounding ($\leq$ 1km) this POI; (l3) Number of trajectories sharing this POI; (l4) Average time in a day for the visits to the POI. Similar to location-level statistics, for trajectory-level aggregate statistics, we select: (t1) Number of users who have the same trajectories; (t2) Number of check-ins surrounding ($\leq$ 1km) all POI in the trajectory; (t3) Number of intercepting trajectories; (t4) Average check-in time of the trajectory.

B DEFENSE

Defense Metrics. As summarized in Table 3, our attacks target different sensitive information on the training dataset. To this end, we evaluate defense mechanisms in terms of their performance in preventing each attack from stealing the corresponding sensitive information. Specifically, we measure their defense performance on protecting *all the sensitive information* and *a targeted subset of sensitive information* for each attack. We define the targeted subset of sensitive information as what defenders want to protect in practice (e.g., some selected user-location pairs in LocMIA). We include this metric because not all the mobility data are sensitive or equally important. In fact, we should evaluate how defense mechanisms perform on the sensitive information that needs to be protected (e.g., user-location pairs that may leak personal identity).

To this end, we jointly measure the defense performance in protecting all the sensitive information and the targeted subset of sensitive information for each attack. Based on different attack objectives, we construct a different targeted subset of sensitive information for measurement by randomly sampling a portion of (e.g., 30%) the most common locations in LocExtract, location sequences in TrajExtract, user-location pairs in LocMIA and trajectory sequences in TrajMIA. It is noted that we randomly sample 30% of sensitive information in each attack to construct the targeted subset for the ease of experiments. In practice, the defender may have more personalized choices based on user-specific requirements, which we leave as future work.

Defense Setup. The GETNext model and 4sq dataset are used for experiments. For L_2 regularization, we use weight decay $\|w\| = 1e^{-2}$ and $3e^{-2}$. For early stopping, we stop training after 5 epochs. For JFT, we mask sensitive information that needs to be protected in phase-I. Then in phase-II, we use DP-SGD [1] with different ϵ_J (1 and 5) to finetune the model. The C and δ are set to 10 and $1e^{-3}$. For Geo-Ind against LocExtract, we apply different ϵ_G (0.01 and 0.05) to replace each sensitive POI with its nearby location such that the original POI is indistinguishable from any location within $r = 400$ meters. Since both JFT and Geo-Ind can be used to protect different amounts of sensitive information, we either protect nearly all the sensitive information or only the targeted subset of sensitive information for each attack, denoted by suffixes (A) and (T).

Results. Figure 12 and 13 showcase the results of existing defenses in mitigating the privacy risks of all the sensitive information and the targeted subset of sensitive information regarding each attack. The results show that existing defenses mitigate our privacy attacks to some extent. However, it is challenging to remove all the vulnerabilities within a reasonable utility drop. This is because existing POI recommendation models heavily rely on memorizing user-specific trajectory patterns to make predictions, which lack semantic information as guidance. As a result, defense mechanisms such as DP-SGD can easily compromise the utility of the protected model due to the noises added to the gradients. Moreover, defenses such as JFT are not general for all inference attacks since each attack steals different sensitive information. To this end, our evaluation calls for more advanced mechanisms to defend against our attacks.