Trajectory test-train overlap in next-location prediction datasets

Massimiliano Luca1,2 · Luca Pappalardo3 · Bruno Lepri2 · Gianni Barlacchi4

Abstract

Next-location prediction, consisting of forecasting a user’s location given their historical trajectories, has important implications in several fields, such as urban planning, geo-marketing, and disease spreading. Several predictors have been proposed in the last few years to address it, including last-generation ones based on deep learning. This paper tests the generalization capability of these predictors on public mobility datasets, stratifying the datasets by whether the trajectories in the test set also appear fully or partially in the training set. We consistently discover a severe problem of trajectory overlapping in all analyzed datasets, highlighting that predictors memorize trajectories while having limited generalization capacities. We thus propose a methodology to rerank the outputs of the next-location predictors based on spatial mobility patterns. With these techniques, we significantly improve the predictors’ generalization capability, with a relative improvement in the accuracy up to 96.15% on the trajectories that cannot be memorized (i.e., low overlap with the training set).

Keywords Human mobility · Next-location prediction · Deep learning · Generalization · Test-train overlap

Editors: Fabio Vitale, Tania Cerquitelli, Marcello Restelli, Charalampos Tsourakakis.

Gianni Barlacchi: Work done prior joining Amazon.

✉ Massimiliano Luca
mluca@fbk.eu

Luca Pappalardo
luca.pappalardo@isti.cnr.it

Bruno Lepri
lepri@fbk.eu

Gianni Barlacchi
gianni.barlacchi@gmail.com

1 Free University of Bolzano, Piazza Domenicani, 3, 39100 Bolzano, Italy
2 Bruno Kessler Foundation, Via Sommarive, 19, 38123 Trento, Italy
3 ISTI-CNR, Via Moruzzi, 1, 56127 Pisa, Italy
4 Amazon Alexa AI, Berlin, Germany
1 Introduction

Next-location prediction is the task of forecasting which location an individual will visit, given their historical trajectories (Luca et al., 2021). It is crucial in many applications such as travel recommendation and optimization (Shi et al., 2019; Khaidem et al., 2020), early warning of potential public emergencies (Barlacchi et al., 2017; Canzian & Musolesi, 2015; Pappalardo et al., 2016; Voukelatou et al., 2020; Pappalardo et al., 2023), estimation of urban emissions (Böhm et al., 2022; Cornacchia et al., 2022; Arora et al., 2021), location-aware advertisements and geo-marketing, and recommendation of friends in social network platforms (Zhu et al., 2015; Burby & Martin, 2012; Wu et al., 2018; Zheng et al., 2018; Zhao, 2020). Predicting an individual’s location is challenging as it requires capturing mobility patterns (Barbosa et al., 2018; Luca et al., 2021) and combining heterogeneous data sources to model the factors influencing displacements (e.g., weather, transportation mode, preferences for specific points of interest).

The striking development of Deep Learning (DL) and the availability of large-scale mobility data has offered an unprecedented opportunity to design powerful next-location predictors (NLs) and has driven test-set performance on mobility data to new heights (Luca et al., 2021). However, little work has been done on how challenging these benchmarks are, what NLs learn, and their generalization capabilities. Although some studies investigate the predictability of human whereabouts and its relationship with the trajectories’ features (Song et al., 2010; Amichi et al., 2020), we know comparatively little about how the individuals’ trajectories are distributed in mobility benchmarks, making it hard to understand and contextualize the observed results. Recent studies in natural language processing (Lewis et al., 2020; Sen & Saffari, 2020) and computer vision (Zhou et al., 2021) show that DL models excel on specific test sets but are not solving the underlying task. In this paper, we investigate whether it is the case for NLs too.

We extensively study the test sets of several public next-location benchmark datasets (Luca et al., 2021) and evaluate a set of state-of-the-art DL-based NLs on their generalization capability. We identify three levels of generalization that an NL should exhibit: (1) known mobility, requiring no generalization beyond recognizing trajectories seen during the training phase; (2) fragmentary mobility, requiring generalization to novel compositions of previously observed trajectories; and (3) novel mobility, requiring generalization to a sequence of movements not present in the training set. It is unclear how well state-of-the-art NLs perform in these three scenarios.

To address this compelling issue, we stratify mobility data by whether the trajectories in the test set also appear (fully or partially) in the training set. We quantify the overlap between trajectories with several measures accounting for different ways of computing the percentage of locations in the test trajectories that are also in the training trajectories.

We find that, in five benchmark datasets, there is a severe problem of trajectory overlapping between the test and training sets when composing them randomly: ~43% to 72% of test trajectories overlap at least with 50% of the points with trajectories in the training set, and 7% to 14% of the test sub-trajectories entirely overlap with the training sub-trajectories. In other words, based on the standard way training and test sets are split, a significant portion of the trajectories in the test sets have already been seen during training.

Based on these observations, we propose to evaluate NLs on stratified test sets based on the overlap between trajectories in the training set. We find significant variability in model performance, varying the percentage of overlap. Indeed, we find an accuracy ≤ 5% when predicting unseen trajectories (novel mobility) and ≥ 90% when predicting trajectories
with high overlaps (known mobility). We also find that DL-based NLs perform even worse than baseline models (e.g., Mobility Markov Chain or MMC (Gambs et al., 2012)) when tested on novel mobility. Our results are consistent across the datasets analyzed and the NLs selected, demonstrating that current train/test splits are flawed, and more robust methods are needed to evaluate the generalization capabilities of NLs. We also show how to improve next-location prediction accuracy, especially for the novel mobility scenario, injecting mobility laws into NLs through a learning-to-rank task. In a nutshell, this paper provides the following novel contributions:

- We show that standard train/test splits of trajectory datasets generate a high trajectory overlap, proposing three metrics to quantify it;
- We evaluate NLs on stratified test sets and show that DL-based NLs do not generalize well on novel mobility, being outperformed by other simpler baselines (e.g., Mobility Markov Chains);
- We show how to improve the accuracy of DL-based NLs, especially for the novel mobility behavior, by performing a rerank of the models’ scores based on spatial mobility patterns;
- Based on our findings, we provide a list of recommendations to improve datasets’ creation and models’ evaluation for next-location prediction.

2 Related work

2.1 Model generalization

Measuring the generalization capabilities of deep neural networks has recently captured the attention of researchers in artificial intelligence. Lewis et al. (2020) find that, in popular Question Answering (QA) datasets, 30% of test-set questions have a near-duplicate in the training sets and that all models perform worse on questions that cannot be memorized from training sets. Sen and Saffari (2020) show that QA models do not generalize well on unseen question-context pairs. However, they still perform well on popular QA benchmarks because of their high overlap between train and test data. Liu et al. (2012) go beyond the data and study the key factors that impact generalization in QA. Cascading errors from retrieval, question pattern frequency, and entity frequency play an essential role in generalization.

2.2 Predictability of human mobility

Several studies measure the limits of predictability of human mobility (Barbosa et al., 2018; Luca et al., 2021). Song et al. (2010) analyze mobility traces of anonymized mobile phone users to find that 93% of the movements are potentially predictable. Zhang et al. (2022) show that, when considering the mobility context (e.g., visiting time, kind of place visited), the upper bound of potential predictability in human mobility increases. Other studies show that this upper bound depends on the data scale and the processing techniques adopted (Smolak et al., 2021; Kulkarni et al., 2019; Hofman et al., 2017). In Amichi et al. (2020); do Couto Teixeira et al. (2021), there are shreds of evidence that the so-called explorers (e.g., individuals without a routinary behavior) (Pappalardo et al., 2015) are less predictable than the others. All the works discussed suggest that models may memorize
specific trajectories (e.g., routinary mobility) while not generalizing well on novel mobility (i.e., mobility not observed during the training phase).

2.3 Next-location Predictors

There are two main ways to design next-location predictors: by leveraging statistical-based or pattern-based approaches and deep learning techniques. We refer to the former as traditional approaches.

Traditional approaches: In a seminal work, Calabrese et al. (2010) propose a probabilistic model that leverages people’s trajectories and geographical features such as Points Of Interest (POIs) and distance of trips. Gambs et al. (2010) introduce a Mobility Markov Chain (MMC) in which states represent POIs and transitions between states correspond to the probability of moving between two POIs (Gambs et al., 2010, 2012). Comito (2017) use individuals’ historical trajectories to train a decision tree for next-location prediction. A similar approach based on decision trees and frequent pattern mining is introduced in Comito (2020). Trasarti et al. (2017) exploit the concept of mobility profiles, a concise representation of individual mobility, to predict the next location using a pattern-based algorithm.

Although these traditional approaches achieve good performance with small data, they have some intrinsic limitations. For instance, they require considerable effort in feature engineering, a complex process requiring domain knowledge, and may neglect valuable features.

Deep learning approaches: Most NLs are based on (gated) Recurrent Neural Networks (RNNs) (Luca et al., 2021). RNNs (Rumelhart et al., 1986) can efficiently deal with sequential data such as time series, in which values are ordered by time, or sentences in natural language, in which the order of the words is crucial to shaping its meaning. In Spatial Temporal Recurrent Neural Networks (ST-RNN) (Liu et al., 2016), RNNs are augmented with time- and space-specific transition matrices. Through linear interpolation, each RNN layer learns an upper and lower bound for the temporal and spatial matrices, which are then used to infer an individual’s next visited location. Long Short-Term Memory Projection (LSTPM) (Sun et al., 2020) use sequential models to capture long- and short-term patterns in mobility data. The authors rely on a non-local network (Wang et al., 2018) for modeling long-term preferences and on geo-dilated RNNs inspired to capture short-term preferences (Chang et al., 2017). More sophisticated models like DeepMove (Feng et al., 2018) use attention layers to capture the periodicity in mobility data. First, past and current trajectories are sent to a multi-modal embedding module to construct a dense representation of spatio-temporal and individual-specific information. Next, an attention mechanism extracts mobility patterns from historical trajectories, while a Gated Recurrent Unit (GRU) handles current trajectories. Finally, the multi-modal embedding, GRU, and attention layer outputs are concatenated to predict the future location. Recently, Spatio-Temporal Attention Network (STAN) (Luo et al., 2021) proposes to explicitly capture spatio-temporal information to leverage spatial dependencies. In particular, the authors use a multi-modal embedding layer to model historical trajectories and the GPS locations in the current trajectories. The embeddings are then forwarded to a spatio-temporal attention mechanism that selects a set of potential next locations. Many other works deal with spatio-temporal data using (gated) RNNs and attention mechanisms. Some of them also deal with the semantic meaning associated with locations. Examples of such models are Semantics-Enriched Recurrent Model (SERM) (Yao et al., 2017), Hierarchical Spatial-Temporal Long-Short Term Memory
(HST-LSTM) (Kong & Wu, 2018), VANext (Gao et al., 2019), and Flashback (Yang et al., 2020). Other Deep Learning solutions to next-location prediction have been discussed in a recent survey paper (Luca et al., 2021).

3 Problem definition

Next-location prediction is commonly defined as the problem of predicting the next location an individual will visit given their historical movements, typically represented as spatio-temporal trajectories (Luca et al., 2021).

Definition 1 (Trajectory) A spatio-temporal point \( p = (t, l) \) is a tuple where \( t \) indicates a timestamp and \( l \) a geographic location. A trajectory \( P = p_1, p_2, \ldots, p_n \) is a time-ordered sequence of \( n \) spatio-temporal points visited by an individual, who may have several trajectories, \( P_1, \ldots, P_k \), where all the locations in \( P_i \) are visited before locations in \( P_{i+1} \).

Given this definition, we formalize next-location prediction as follows:

Problem 1 (Next-location prediction) Given the current trajectory of an individual \( P_k = p_1, p_2, \ldots, p_n \) with at least two points and their historical trajectories \( \mathcal{H} = P_1, \ldots, P_{k-1} \), next-location prediction is the problem of forecasting the next point \( p_{n+1} \in P_k \).

In other terms, a next-location predictor (NL) is a function \( \mathcal{M}(P_k, \mathcal{H}) \rightarrow p_{n+1} \), which takes the current trajectory \( P_k \), the set of \( u \)'s historical trajectories \( \mathcal{H} \), and returns a spatio-temporal point \( p_{n+1} \) in \( P_k \).

4 Trajectory overlap

An NL should be able to predict an individual’s location in three scenarios: (1) the NL has seen the individual’s entire current trajectory during the training phase; (2) it has seen the current trajectory only partially, or it has seen a very similar trajectory of the same individual; (3) the current trajectory was absent from the training set. The latter scenario is essential, as machine learning models’ ability to generalize is their capacity to make predictions on data never seen during the training phase (Kawaguchi et al., 2017).

However, in next-location prediction, there may be a significant overlap between trajectories in the test set and those in the training set. For example, some test and training trajectories may belong to the same individual. Since human mobility is routinary, an individual’s trajectories are similar to each other (Barbosa et al., 2018; Schläpfer et al., 2021), leading to scenarios (1) and (2) above. Given this discussion, we investigate the extent to which the overlap between trajectories in the test and training sets influences the model’s ability to generalize. We explore three ways to examine overlap: Jaccard Similarity (JS), Longest Common Subsequence (LCST), and Overlap From the End (OFE). Note that we measure the overlap only on the sequences of visited locations and regardless of the time of the visits. Each metric aims to measure different types of overlap, which may allow us to understand better what the models are memorizing. For example, JS does not consider the order of the visits between two trajectories. It simply measures the number of common locations. On the other hand, LCST and OFE also consider the order in which the locations appear in the trajectories.
Similarity (JS) measures the percentage of locations in the test trajectories that are also in the training trajectories, regardless of the order in which locations appear. Test trajectories with a high JS have many locations in common with training trajectories. In contrast, test trajectories with low JS should be less predictable as they are mainly composed of locations that are not in the training trajectories. Formally, we define the JS between a trajectory \( R \in D_{\text{test}} \) and \( P \in D_{\text{train}} \) as:

\[
JS(P, R) = \frac{|P \cup R| - |P \cap R|}{|P \cup R|}
\]

We quantify the overlap between \( R \) and the training set as the maximum JS over all the trajectories in the training set:

\[
\max_{P \in D_{\text{train}}} JS(P, R).
\]

JS \( \in [0, 1] \), where 1 indicates a full overlap (all locations in \( R \) are at least in a trajectory in \( D_{\text{train}} \)) and 0 indicates no overlap (none of the locations in \( R \) are in the training set). With JS, we measure the overlap between two trajectories without considering the temporal dimension (i.e., the order and the sequence of the visits) and the number of times a location is visited. This allows us to evaluate whether NLs can generalize over trajectories containing the same locations with different visitation frequencies and at different times. The behavior of NLs on different levels of overlap measured with JC may shed light on the role of the temporal and spatial order of the visits on the models’ generalization power.

The Longest Common SubTrajectory (LCST) is the longest subtrajectory in common between two trajectories. Formally, given two trajectories \( P = p_1, p_2, \ldots, p_n \) and \( R = r_1, r_2, \ldots, r_m \), a prefix \( P_i \) of \( P \) are the list of the first \( i \) locations in \( P \) (i.e., \( P_i = p_1, \ldots, p_i \)). Prefixes are defined in the same way for \( R \).

\[
\text{LCST}(P_i, R_j) = \begin{cases} 
0, & \text{if } i = 0 \text{ or } j = 0 \\
\text{LCST}(P_{i-1}, R_{j-1}) + 1, & \text{if } i, j > 0 \text{ and } p_i = r_j \\
\max (\text{LCST}(P_{i-1}, R_j), f(P_{i-1}, R_{j-1})) & \text{if } i, j > 0 \text{ and } p_i \neq r_j
\end{cases}
\]

We quantify the overlap between \( R \) and the training set as the maximum LCST over all the trajectories in the training set:

\[
\max_{P \in D_{\text{train}}} \text{LCST}(P, R).
\]

Differently from JC, LCST measures the overlap between two trajectories by considering the order and the frequency of the visits. Since the visits are time-ordered, even if the temporal dimension is not explicitly captured, LCST considers the temporal dimension to some extent. With LCST, we measure the role that periodicity (e.g., visits to the same locations in the same order) has on generalization. The Overlap From End (OFE) enforces that the common subtrajectory is at the end of the two trajectories. Given two trajectories \( P = p_1, p_2, \ldots, p_n \) and \( R = r_1, r_2, \ldots, r_m \), a suffix \( P^-_i \) of \( P \) is the list of the last \( i \) locations in \( P \) (i.e., \( P^-_i = p_{n-i}, \ldots, p_{n} \)). Suffixes are defined in the same way for \( R \).

\[
\text{OFE}(P, R) = \arg \max_{i \in 0, \ldots, \min(n,m)} P^-_i = R^-_i
\]
We quantify the overlap between $R$ and the training set as the maximum OFE over all the trajectories in the training set:

$$\max_{P \in D_{\text{train}}} \text{OFE}(P, R).$$

With LCST, it is unclear how far in time the overlapping sequence of visits took place. With OFE, we measure only the overlap of the most recent visits and thus the overlap of the most recent mobility and its effect on generalization.

We use the maximum values instead of the average because the distribution of visits to locations follows a power law (Schläpfer et al., 2021). Additionally, many places in mobility datasets (especially check-in ones) are only visited once and most visits to previously visited locations are made by the same user. This results in a high number of trajectories with a 0% overlap as the overlap calculation is not based on the user. Using the average instead of the maximum value would unfairly lower the overlap score for these trajectories.

### 5 Experimental setup

#### 5.1 Datasets

We use five public datasets widely adopted in the literature to evaluate NLs (Luca et al., 2021) (see Table 1). Three of them (Gowalla, Foursquare New York, and Foursquare Tokyo) are collected through social networking platforms, in which mobility traces are generated by the users’ georeferenced posts (check-ins). Consequently, these mobility traces are sparse both in time and space.

The other two datasets (Taxi Porto and Taxi San Francisco) describe GPS traces from taxis, thus being dense in space and time. In detail, Gowalla was a location-based social network platform that, like Foursquare, allowed users to check-in at so-called spots (venues) via a website or an app. The dataset (Cho et al., 2011) has almost six million check-ins collected over a year and a half, from February 2009 to October 2010. Each check-in contains the user identifier, location identifier, latitude and longitude pair, and timestamp. The dataset also contains information on the users’ friendship network, which has around 200,000 nodes and one million edges. Foursquare is another location-based social network platform that allows users to check in to places.

| Dataset            | Users | Locations | Trajectories |
|--------------------|-------|-----------|--------------|
| Gowalla (Cho et al., 2011) | 5300  | 125,771   | 72,593       |
| Foursquare NYC (Yang et al., 2014) | 4390  | 13,960    | 12,519       |
| Foursquare Tokyo (Yang et al., 2014) | 935   | 21,394    | 34,662       |
| Taxi Porto (Moreira-Matias et al., 2013) | 500   | 8524      | 94,214       |
| Taxi SF (Piorkowski et al., 2009) | 500   | 9321      | 103,120      |
Data can be collected through the available APIs. A widely used dataset based on Foursquare is described in Yang et al. (2014). The information contained is the same as Gowalla, with additional information about the venue category.

Piorkowski et al. (2009) collected taxi trajectories in San Francisco in May 2008. Each point in a trajectory includes the taxi’s identity, latitude, longitude, timestamp, and occupancy. Points are sampled every 10 s on average. Moreira-Matias et al. (2013) (ECML/PKDD Challenge) collected taxi trajectories in Porto, Portugal. For each trajectory, we have the taxi’s identifier, the latitude, longitude, and timestamp showing when the trip began. For each trajectory, data are sampled every 15 s. The dataset also includes auxiliary information for each trip, such as the trip’s typology (e.g., sent from the central, demanded to the operator, demanded to the driver), the stand from which the taxi left, and a phone number identification for the passenger.

To extract trajectories from these datasets, we follow the same approach as in Feng et al. (2018): we filter out the users with less than ten records and cut the sequence of records into several trajectories for each user based on the time interval between two neighbor records. As in Feng et al. (2018), we choose 72 h as the default interval threshold based on the practice. Finally, we remove the users with less than five trajectories.

5.2 Models

We consider a set of models that can deal with the spatial (i.e., sequences of identifiers of visited locations) and temporal (i.e., sequences of timestamps of visited locations) aspects of trajectories. Information like social dimension and time spent traveling between locations are not considered.

5.2.1 Baselines

We consider the following traditional approaches:

- **MMC** (Gambs et al., 2012) consists of a first-order Markov chain with the states representing locations and the transitions between states being the probability of going from the current location to the others.
- **NEXT** (Comito, 2020) is based on training M5 decision trees and frequent pattern mining to predict the next possible location.
- **MyWay** (Trasarti et al., 2017) introduces the concept of mobility profile, a way to represent meaningful information in historical traces. Mobility profiles are then analyzed using pattern-based algorithms to predict the next location.

Traditional approaches achieve good performances in next-location prediction but they require a significant effort in feature engineering. These methods also assume that patterns in the training data can be used to accurately predict future locations. However, these pattern-based approaches may not be effective in predicting new or novel mobility, as they are unable to discover new patterns in these cases. Deep learning-based approaches may provide a solution to this issue.
5.2.2 DL-based models

We consider the following state-of-the-art DL-based NLs:

- **RNN** (Rumelhart et al., 1986), the building block of the majority of NLs. RNNs are commonly adopted to model sequential data such as time series and natural language, in which the order of the items is crucial to shaping its meaning. RNNs are also widely used as building blocks of NLs to capture spatial and temporal patterns in the trajectories.

- **ST-RNN** (Liu et al., 2016) enhances RNNs with time- and space-specific transition matrices. Each RNN layer learns an upper and lower bound for the temporal and spatial matrices via linear interpolation. These matrices are then used to predict where a person will go next.

- **Deep Move** (Feng et al., 2018) uses attention mechanisms to capture spatio-temporal periodicity in the historical trajectories. Also, the model uses GRUs (gated RNNs) to capture patterns in the current trajectory, relying on a multi-modal embedding to capture individual preferences and project trajectories in a low-dimensional space before passing them to the attention mechanisms and GRUs.

- **LSTPM** (Sun et al., 2020) combines long- and short-term sequential models: long-term patterns are modeled using a non-local network (Wang et al., 2018), short term preferences are captured using a geographic-augmented version of the concept of dilated RNNs (Chang et al., 2017).

- **STAN** (Luo et al., 2021) explicitly captures spatio-temporal information using a multi-modal embedding to represent the trajectories and a spatio-temporal attention

![Fig. 1](image-url)  
**Fig. 1** Fraction of the test trajectories with an overlap of 0–20%, 20–40%, 40–60%, 60–80%, and 80–100% with the training trajectories, for all datasets and evaluation metrics (JS, LCST, and OFE)
mechanism to capture patterns in the data. The role of the attention mechanisms, supported by a balanced sampler, is to rank potential next locations.

In all the DL-based NLs mentioned, trajectories are represented as sequences of timestamps and identifiers. RNNs look for simple and complex patterns in the sequences, while attention mechanisms help the models determine which locations are the most likely next locations, given the context (e.g., the current trajectory). Like traditional approaches, DL solutions assume that training and test data come from the same distribution. However, this assumption may not hold true when predicting novel mobility patterns, leading to a problem with generalization.

5.3 Training

We split the trajectories into a training set, a validation set, and a test set for each dataset. All sets include trajectories from several users. We sort the trajectories temporally for each user and put the first 70% in the training set, the following 10% in the validation set, and the remaining 20% in the test set.

All models are implemented with PyTorch and are made available through the library LibCity (Wang et al. 2021). We follow the same configuration as Feng et al. (2018) and use Adam (Kingma and Ba 2014) as optimizer. Detailed information about the hyperparameters used is in “Appendix D”. We ran the experiments on a machine with 126GB of memory and two Nvidia RTX 2080Ti.

6 Testing generalization capability

We evaluate the performance of all models using the $k$-accuracy (ACC@$k$), the most common evaluation metric in the literature (Luca et al., 2021). NLs output a list of all possible locations an individual will visit ranked from the most to the least likely. ACC@$k$ indicates how often the true location is among the $k$ top predicted locations. We evaluate all models using ACC@5 and compare the DL models with traditional approaches.

For all datasets and overlap metrics (JS, LCST, and OFE), we compute the number of trajectories in the test set with an overlap with the training set between 0–20%, 20–40%, 40–60%, 60–80%, and 80–100%. We divide the trajectories into five bins to categorize the overlap and evaluate the generalization of NLs. This helps present the results in a clear manner. However, it is important to note that the results are still valid even without dividing the trajectories into bins, as shown in Appendices F and G. Figure 1 shows the results for all the datasets analyzed.

The amount of trajectories with a high overlap (80–100%) varies depending on the overlap metric and dataset. Taxi datasets have more high-overlap trajectories compared to check-in datasets, indicating the overlap problem is worse in GPS traces. JS and LCST have similar overlaps, but OFE has significantly more low-overlap trajectories due to its strict constraints (e.g., overlap evaluation only starts from the end of the trajectory).

Figure 1 highlights that a significant overlap exists between the test and the training set, introducing a bias when evaluating NLs using a random train-test split. Hence, we investigate to what extent this overlap affects model performance. In “Appendix F”, we provide
a more fine-grained distribution of the overlaps with trajectories that are not divided into bins.

The performance of all NLs and overlap metrics is shown in Fig. 2. Higher overlap leads to significant improvement in the performance of all NLs. Traditional models, such as MMC, have ACC@5 comparable to DL models, particularly in check-in datasets. MyWay and NexT perform better in taxi datasets than MMC. Detailed performance results can be found in “Appendix A”. For instance, with Foursquare New York and OFE, the performance of NLs is nearly 100% for test set trajectories with 80–100% overlap with the training set. Results for Porto taxi show a similar trend, with less impressive results.

Figure 2 suggests that model performance is highly impacted by the overlap between the test and train trajectories, indicating that NLs tend to memorize rather than generalize. The NLs perform well when the test trajectories have high overlap with the training set but poorly when there is low overlap. These results raise the question of how to improve the accuracy of NLs for low-overlap scenarios. In Fig. 2 we report the average accuracy for each bin of trajectories, and in “Appendix G” we report the accuracy for each trajectory without classification in bins. “Appendix G” shows that our results are general and independent of the particular choice of bins (e.g., the results still hold for more fine-grained bins).
7 Learning to rank locations using mobility laws

A possible reason why NLs perform poorly on trajectories is that RNNs focus on memorizing regularities in long sequences, thus limiting NLs’ generalization capabilities. Wrong location predictions may happen, for example, when the NL’s probabilities assigned to each potential location (i.e., the locations’ scores) are relatively uniformly distributed. Indeed, in “Appendix H” we compute the normalized entropy over the output of the softmax layer for trajectories with 0–20 overlaps. Entropy, by definition, can be used to measure how much a distribution is uniformly distributed. In particular, as closer the values in “Appendix H” are to 1, then the probabilities of choosing a particular next location are uniformly distributed. For instance, if the softmax is supposed to output a probability for \( N \) classes, having a score of 1 means that all the classes have an equal probability of \( 1/N \) to be chosen as the next location. The tables in “Appendix H” suggest that NLs may have to deal with uncertain and uniformly distributed outputs of the softmax layer. Our intuition is that while a location in the test set may not (or rarely) appear in the training set, we may exploit well-known mobility laws to capture an individual’s behavior and predict the location they will

![Figure 3](image-url)  

**Fig. 3** Results (in terms of ACC@5) for all the datasets for the three overlap metrics (JS, LCST, OFE) for trajectories with a 0–20% overlap. In green, a baseline using only the mobility laws (reranker). In black, the accuracy of the models without the reranker and, in red, the accuracy with the reranker on top of the models. We provide the results for all datasets and the other overlaps in “Appendix B”
potentially visit and reduce the entropy score through re-ranking techniques that account for mobility laws. Note that we do not introduce a novel DL-based NL but a framework that can be placed on top of any NL. Our reranker leverages well-known mobility laws to support NLs in making the correct prediction when dealing with novel mobility.

We select three prominent human mobility laws (Barbosa et al., 2018; Luca et al., 2021):

- the distance law (Barbosa et al., 2018): people prefer traveling short distances. Given an individual’s trajectories \( P = p_1, p_2, \ldots, p_n \), we compute the haversine distance between all the consecutive locations \( p_i, p_{i+1} \) and consider the average of the distances as a feature \( \text{dist}_u \);
- the visitation law (Schläpfer et al., 2021): the visits to a location decrease as the inverse square of the product of their visiting frequency and travel distance. We denote as \( f \) the number of visits to a location (by any individual) and compute the number of people

Table 2  ACC@5 of all the models on all the datasets

| NL + RR | JS | LCST | OFE |
|---------|----|------|------|
| Foursquare NYC | RNN .233 (+ 9.38%) | .051 (+ 96.15%) | .158 (+ 19.69%) | .241 (+ 26.17%) |
| ST-RNN .261 (+ 5.24%) | .059 (+ 84.37%) | .186 (+ 15.52%) | .299 (+ 31.14%) |
| Deep Move .277 (+ 6.94%) | .084 (+ 64.71%) | .213 (+ 19.66%) | .268 (+ 16.01%) |
| LSTPM .272 (+ 8.36%) | .072 (+ 56.52%) | .184 (+ 10.84%) | .271 (+ 18.34%) |
| STAN .281 (+ 6.43%) | .101 (+ 32.89%) | .214 (+ 15.05%) | .283 (+ 18.90%) |

| Foursquare TKY | RNN .196 (+ 6.56%) | .028 (+ 40.02%) | .123 (+ 21.78%) | .171 (+ 15.54%) |
| ST-RNN .213 (+ 7.58%) | .057 (+ 67.65%) | .133 (+ 15.64%) | .194 (+ 24.36%) |
| Deep Move .223 (+ 5.69%) | .060 (+ 46.34%) | .142 (+ 19.33%) | .201 (+ 19.64%) |
| LSTPM .233 (+ 6.88%) | .074 (+ 57.45%) | .151 (+ 21.77%) | .236 (+ 34.86%) |
| STAN .246 (+ 9.33%) | .093 (+ 82.35%) | .153 (+ 18.60%) | .239 (+ 32.04%) |

| Gowalla | RNN .142 (+ 29.09%) | .157 (+ 68.82%) | .016 (+ 45.45%) | .144 (+ 50.03%) |
| ST-RNN .149 (+ 8.76%) | .143 (+ 28.83%) | .033 (+ 17.86%) | .127 (+ 19.81%) |
| Deep Move .165 (+ 4.43%) | .164 (+ 41.38%) | .041 (+ 13.89) | .151 (+ 32.46%) |
| LSTPM .171 (+ 12.50%) | .182 (+ 61.06%) | .043 (+ 34.38%) | .151 (+ 36.04%) |
| STAN .206 (+ 6.77%) | .178 (+ 43.55%) | .059 (+ 15.69%) | .146 (+ 22.69%) |

| Taxi Porto | RNN .421 (+ 2.68%) | .069 (+ 4.54%) | .296 (+ 10.2%) | .398 (+ 1.79%) |
| ST-RNN .427 (+ 2.64%) | .077 (+ 5.47%) | .313 (+ 3.98%) | .418 (+ 5.55%) |
| DeepMove .466 (+ 5.42%) | .104 (+ 6.12%) | .341 (+ 3.96%) | .434 (+ 6.11%) |
| LSTPM .457 (+ 6.52%) | .095 (+ 6.74%) | .336 (+ 6.32%) | .419 (+ 5.01%) |
| STAN .483 (+ 6.62%) | .111 (+ 7.96%) | .351 (+ 6.68%) | .440 (+ 7.05%) |

| Taxi SF | RNN .288 (+ 2.49%) | .193 (+ 4.89%) | .208 (+ 1.46%) | .276 (+ 5.34%) |
| ST-RNN .297 (+ 4.95%) | .200 (+ 5.82%) | .225 (+ 6.64%) | .298 (+ 8.76%) |
| Deep Move .313 (+ 5.74%) | .202 (+ 4.12%) | .227 (+ 4.13%) | .297 (+ 6.45%) |
| LSTPM .301 (+ 5.24%) | .199 (+ 3.65%) | .238 (+ 9.68%) | .293 (+ 5.02%) |
| STAN .330 (+ 5.11%) | .208 (+ 3.48%) | .233 (+ 4.48%) | .309 (+ 5.79%) |

We find a significant relative improvement, especially on the trajectories with a 0–20 overlap. Regarding check-in datasets, we have the greatest relative improvement on the stratification based on JS (in bold). In taxi datasets, we have similar improvements on JS and OFE, and while in Taxi Porto we have the best improvements on JS, on Taxi San Francisco we reach the best improvements on OFE. In general, the improvements in check-in datasets are higher with respect to taxi datasets. There is not a specific model on which we have the best improvements.
visiting it within a distance $r$. An individual’s probability to visit location $p_{i+1}$ is given by a power-law of the form $p_{i+1}(r, f) = \mu_i/(rf)^\gamma$, with $\gamma = 1.6$, a parameter fitted with the least squares method. We use the five most probable locations $top_n, n \in 1 \ldots 5$ as an input to the reranker.

- the returner and explorer dichotomy (Pappalardo et al., 2015): individuals naturally tend to split into two profiles based on their degree of spatial exploration. We compute the average radius of gyration $r_g(u)$ and the 2-radius of gyration $r_g^{(2)}(u)$ for each individual and compute the ratio $\frac{r_g(u)}{r_g^{(2)}(u)}$ using the scikit-mobility library (Pappalardo et al., 2022). The individual’s profile is then translated into a binary feature: zero if the individual is a returner and one if the individual is an explorer. We denote this feature as $re_u$.

Our approach consists of predicting the next location using an NL, then combining it into a single scoring model, i.e., a fully connected neural network, both the NL score for the location and the mobility laws. We train the network using the binary cross-entropy loss $L = -\sum_{i \in \{0, 1\}} y_i \log p(y_i)$, where $y_i$ is the label (i.e., 0 or 1) and $p(y_i)$ is the predicted probability.

The training dataset consists of vectors of the form $[NL_i(P), dist_u, top_1, \ldots, top_5, re_u]$. We denote with $NL_i(P)$ the score of the NL for a given location $i$ starting from a trajectory $P$. The label is one if the location $i$ is the individual’s next location and zero otherwise. This means that in a dataset with $n$ locations, we have a positive sample (e.g., the correct location) and $n - 1$ negative samples for each trajectory. As the number of incorrect samples is much higher than the correct ones, for each correct location, we randomly sampled $k = 20$ wrong locations (e.g., those different from the actual location) as a good trade-off between model performance and dataset size.

Table 2 and Fig. 3 show how the accuracy changes on the test trajectories with 0–20% overlap on all the datasets and models considered. For the sake of completeness, we also test the possibility of using only the mobility laws as NL (green bars in Fig. 3). Not surprisingly, the performance of the reranker alone is low as it does not consider the history of the users: the reranker cannot capture any routinely spatial and temporal aspects from the data.

When used with NLS, our reranking improves accuracy for all datasets and overlap metrics. The most considerable improvements are related to the trajectories with an overlap of 0–20%. We define the relative improvement (RI) as the difference between the reranked prediction $\hat{y}$ and the original prediction $y$, divided by the original prediction:

$$RI = \frac{\hat{y} - y}{y}$$

On Foursquare New York, the improvement varies from + 3.25% (ST-RNN) to + 9.38 (LSTPM). Similarly, on Foursquare Tokyo, the improvement varies from + 5.69% (DeepMove) to a + 9.33% of improvement (STAN). On Gowalla, we have the lowest relative improvement on DeepMove (+ 4.43%) and the highest on RNN (+ 29.09%). On Taxi Porto, the relative improvement on the average case (i.e., without stratifying the test set) varies from a + 2.68% (RNN) to + 5.84% (STAN). On Taxi San Francisco, the relative improvement varies from + 2.49% (RNN) to + 5.74% (DeepMove). Regarding the 0–20% stratification, the most considerable relative improvement is associated with JS, followed by LCST and OFE. On Foursquare New York, the relative improvement with JS is up to + 96.15%, LCST being + 20.39%, and OFE being + 33.05%. Similarly, on Foursquare
Tokyo, we have top relative improvements of +82.35%, +21.78%, and +24.36% with JS, LCTS, and OFE, respectively. Finally, Gowalla’s top relative improvements for JS, LCTS, and OFE are +68.82%, +45.45%, and +50.03%, respectively. In general, taxi datasets are associated with the lowest relative improvement: on Taxi Porto, with JS, it is up to +7.96%, with LCST +6.68%, and with OFE +7.05%. On Taxi San Francisco, the relative improvements for JS, LCST, and OFE are +5.82%, +9.68%, and +8.76%. The highest relative improvement is associated with the 0–20% overlap scenario.

Regarding the other overlap intervals, we reported the relative improvements in “Appendix C”. Concerning JS and the check-in datasets, with an overlap of 20–40%, we have relative improvements from 12.57 to 42.47%. With LCST, the performances improve between 0.23 and 16.26%; with OFE, the improvement is between 12.25 and 36.41%. For the overlap 60–80%, the performance improves between 0.93 and 7.03% with JS, between 0.17 and 6.99% with LCST, and between 0.0009 and 0.77% with OFE. When the overlap measured with OFE is between 40 and 100%, the models have an improvement larger than 0.009% only on Gowalla. We generally have similar behavior on taxi datasets but considerably lower relative improvements. In other words, our rerank strategy brings the most considerable improvement in accuracy, especially where NLs are the least accurate. In “Appendix E”, we present some results concerning the stability of the training and the statistical significance of the results obtained. In particular, we run the experiments for all the datasets and all the models five times and we present the average ACC@5 and the relative standard deviations.

8 Discussion and recommendations

This work finds that the models’ performances are deeply affected by the level of overlap in the test trajectories. Based on the amount of trajectory overlap, we identify three scenarios:

- **Known mobility**: the NL sees the entire trajectory in the training phase (overlap between 80% and 100%). Predictive performance is much higher than the performance on a non-stratified test set (close to 100%) as the test trajectories are almost identical to the training trajectories.
- **Fragmentary mobility**: the NL sees a significant portion of the trajectory (overlap between 20% and 80%). The majority of trajectories in the test set lie in this scenario. There is a drop in the model performance compared to the previous scenario, decreasing to ~80%.
- **Novel mobility**: the NL sees a tiny or no portion of the trajectory (overlap below 20%). A significant number of trajectories lie in this scenario. However, since NLs cannot rely on the trajectories already seen in the training phase, these are the most difficult trajectories to predict. Indeed, the performance of NLs on test sets with low overlap is considerably lower than on a non-stratified test set.

While predicting known mobility is a simple task, inferring mobility patterns for fragmentary mobility and novel mobility presents challenges (e.g., dealing with under-represented locations or not represented at all in the training set). From a modeling perspective, this may suggest that current models are excellent in memorizing already seen trajectories but cannot generalize well. Some works suggest that reranking techniques or few-shot learning algorithms may help solve this problem (Wang et al., 2020). Also, results indicated that
NLs might not be evaluated adequately. In this sense, here we provide a set of recommendations for the evaluation of NLs:

1. MMCs achieve performance similar to NLs. Therefore, we claim that MMCs and other Markov chain approaches should always be used as a baseline.

2. Although NLs achieve good overall performance, they are biased due to trajectory overlap. Besides the NLs’ average performance, researchers should report the performance for the known mobility and the novel mobility scenarios. It is indeed crucial to understand whether the improved performance of the proposed NL is actually due to its increasing generalization capability or because it is memorizing better the trajectories in the training set;

3. NLs achieve the worst performance on the 0–20% overlap bin. We can improve the performance on this bin, hence increasing NLs’ generalization capability with the support of well-known spatial mobility laws, which are loosely captured by state-of-the-art NLs given their reliance on RNNs.

Designing models able to generalize is crucial in many contexts, from what-if analysis to urban planning and sustainability (Blanc, 2015; Kroll et al., 2019). For instance, we may use such models to analyze the mobility of individuals never seen in a region (e.g., tourists), to assess how attractive a new Point of Interest in a city would be, or to investigate the impact of mobility on pollution and inclusion for those cities for which there is a scarcity of data. In this paper, we show that injecting mobility laws to rerank the outputs of NLs increases accuracy for trajectories with low overlaps, representing a first step towards improving the generalization power of NLs.

9 Conclusions

In this work, we investigate the generalization capabilities of next-location prediction datasets. We find that model performance is considerably affected by trajectory test-train overlap, suggesting that NLs memorize training trajectories rather than generalizing. We mitigate this issue by injecting mobility laws into state-of-the-art NLs, achieving positive relative improvement on test sets with low overlap with the training ones. We aim to consider other mobility laws and use more sophisticated models to rerank the results in future work. It would also be helpful to use explainable AI techniques to understand better the role of mobility laws and the relations between the DL modules in NLs.

Appendix A: NL’s performances

See Table 3.
|        | JC                      | LCST                    | OFE                      |
|--------|-------------------------|-------------------------|--------------------------|
| Gowalla |                         |                         |                          |
| MMC    | 0.129 0.108 0.119 0.323 0.526 0.776 | 0.013 0.182 0.398 0.644 0.837 | 0.099 0.211 0.546 0.783 0.874 |
| NEXT   | 0.101 0.077 0.089 0.310 0.487 0.749 | 0.009 0.154 0.373 0.613 0.804 | 0.084 0.192 0.518 0.763 0.852 |
| MyWay  | 0.098 0.074 0.089 0.306 0.484 0.748 | 0.008 0.158 0.371 0.612 0.801 | 0.079 0.191 0.518 0.762 0.855 |
| RNN    | 0.110 0.093 0.109 0.321 0.498 0.764 | 0.011 0.173 0.385 0.631 0.824 | 0.096 0.209 0.532 0.777 0.871 |
| ST-RNN | 0.137 0.111 0.128 0.355 0.513 0.782 | 0.028 0.195 0.406 0.658 0.849 | 0.106 0.217 0.558 0.793 0.888 |
| DeepMove | 0.158 0.116 0.142 0.373 0.536 0.798 | 0.036 0.211 0.434 0.681 0.869 | 0.114 0.226 0.573 0.822 0.899 |
| LSTPM  | 0.152 0.113 0.141 0.369 0.528 0.793 | 0.032 0.203 0.427 0.669 0.861 | 0.111 0.219 0.569 0.818 0.897 |
| STAN   | 0.192 0.124 0.183 0.398 0.561 0.813 | 0.051 0.221 0.449 0.704 0.888 | 0.119 0.231 0.579 0.829 0.911 |
| Foursquare NYC |                         |                         |                          |
| MMC    | 0.245 0.045 0.214 0.518 0.730 0.889 | 0.102 0.336 0.553 0.588 0.987 | 0.230 0.529 0.778 0.885 0.988 |
| NEXT   | 0.201 0.016 0.167 0.455 0.623 0.723 | 0.120 0.253 0.486 0.518 0.851 | 0.179 0.59 0.696 0.909 0.971 |
| MyWay  | 0.195 0.015 0.161 0.455 0.623 0.721 | 0.112 0.252 0.489 0.512 0.850 | 0.177 0.59 0.698 0.907 0.977 |
| RNN    | 0.213 0.026 0.186 0.472 0.643 0.740 | 0.132 0.271 0.504 0.528 0.869 | 0.191 0.610 0.712 0.926 0.988 |
| ST-RNN | 0.248 0.032 0.236 0.538 0.709 0.859 | 0.161 0.322 0.544 0.589 0.920 | 0.228 0.679 0.779 0.928 0.989 |
| DeepMove | 0.259 0.051 0.241 0.539 0.716 0.873 | 0.178 0.330 0.569 0.631 0.926 | 0.231 0.683 0.786 0.935 0.996 |
| LSTPM  | 0.251 0.046 0.239 0.533 0.717 0.865 | 0.166 0.327 0.551 0.599 0.922 | 0.229 0.680 0.782 0.929 0.991 |
| STAN   | 0.264 0.076 0.269 0.544 0.732 0.875 | 0.186 0.331 0.574 0.663 0.925 | 0.238 0.686 0.792 0.936 0.992 |
| Foursquare TKY |                         |                         |                          |
| MMC    | 0.216 0.043 0.172 0.450 0.691 0.856 | 0.096 0.243 0.428 0.634 0.718 | 0.199 0.360 0.691 0.716 0.991 |
| NEXT   | 0.173 0.017 0.143 0.377 0.599 0.774 | 0.087 0.195 0.371 0.578 0.675 | 0.132 0.454 0.686 0.724 0.882 |
| MyWay  | 0.171 0.018 0.142 0.376 0.596 0.772 | 0.084 0.191 0.371 0.578 0.673 | 0.138 0.453 0.683 0.725 0.879 |
| RNN    | 0.183 0.020 0.153 0.390 0.609 0.783 | 0.101 0.205 0.381 0.593 0.692 | 0.148 0.464 0.703 0.737 0.892 |
| ST-RNN | 0.198 0.034 0.158 0.397 0.637 0.794 | 0.115 0.224 0.394 0.608 0.699 | 0.156 0.468 0.711 0.739 0.906 |
| LSTPM  | 0.211 0.041 0.164 0.431 0.688 0.848 | 0.119 0.233 0.419 0.621 0.715 | 0.168 0.471 0.718 0.743 0.931 |
| DeepMove | 0.218 0.047 0.168 0.452 0.694 0.854 | 0.124 0.239 0.424 0.638 0.719 | 0.175 0.472 0.723 0.744 0.935 |
| STAN   | 0.225 0.051 0.174 0.458 0.696 0.857 | 0.129 0.244 0.426 0.639 0.722 | 0.181 0.475 0.728 0.749 0.938 |
|                | JC 0–20 | 20–40 | 40–60 | 60–80 | 80–100 | LCST 0–20 | 20–40 | 40–60 | 60–80 | 80–100 | OFE 0–20 | 20–40 | 40–60 | 60–80 | 80–100 |
|----------------|---------|-------|-------|-------|--------|-----------|-------|-------|-------|-------|--------|-------|-------|-------|-------|
| Taxi Porto     |         |       |       |       |        |           |       |       |       |       |        |       |       |       |       |
| MMC            | 0.309   | 0.113 | 0.258 | 0.316 | 0.385  | 0.482     | 0.274 | 0.302 | 0.343 | 0.397 | 0.411  | 0.286 | 0.333 | 0.349 | 0.427 | 0.436 |
| NEXT           | 0.399   | 0.056 | 0.257 | 0.388 | 0.503  | 0.628     | 0.281 | 0.37  | 0.447 | 0.531 | 0.636  | 0.381 | 0.446 | 0.504 | 0.607 | 0.631 |
| MyWay          | 0.396   | 0.047 | 0.253 | 0.379 | 0.504  | 0.627     | 0.280 | 0.368 | 0.446 | 0.526 | 0.633  | 0.378 | 0.439 | 0.502 | 0.601 | 0.629 |
| RNN            | 0.410   | 0.066 | 0.269 | 0.398 | 0.516  | 0.638     | 0.293 | 0.381 | 0.461 | 0.542 | 0.649  | 0.391 | 0.458 | 0.516 | 0.617 | 0.643 |
| ST-RNN         | 0.416   | 0.073 | 0.272 | 0.404 | 0.519  | 0.639     | 0.301 | 0.395 | 0.467 | 0.544 | 0.653  | 0.396 | 0.463 | 0.518 | 0.623 | 0.649 |
| DeepMove       | 0.442   | 0.098 | 0.296 | 0.419 | 0.528  | 0.651     | 0.328 | 0.417 | 0.486 | 0.561 | 0.672  | 0.409 | 0.467 | 0.532 | 0.637 | 0.684 |
| LSTPM          | 0.429   | 0.089 | 0.281 | 0.413 | 0.521  | 0.647     | 0.316 | 0.411 | 0.473 | 0.558 | 0.668  | 0.399 | 0.466 | 0.524 | 0.629 | 0.661 |
| STAN           | 0.453   | 0.103 | 0.305 | 0.421 | 0.532  | 0.649     | 0.329 | 0.423 | 0.492 | 0.558 | 0.679  | 0.411 | 0.474 | 0.537 | 0.631 | 0.692 |
| Taxi SF        |         |       |       |       |        |           |       |       |       |       |        |       |       |       |       |
| MMC            | 0.242   | 0.169 | 0.261 | 0.385 | 0.542  | 0.583     | 0.194 | 0.248 | 0.308 | 0.451 | 0.505  | 0.231 | 0.256 | 0.389 | 0.427 | 0.692 |
| NEXT           | 0.269   | 0.171 | 0.266 | 0.392 | 0.551  | 0.589     | 0.196 | 0.251 | 0.310 | 0.464 | 0.529  | 0.251 | 0.292 | 0.400 | 0.491 | 0.806 |
| MyWay          | 0.267   | 0.167 | 0.265 | 0.392 | 0.550  | 0.586     | 0.192 | 0.250 | 0.309 | 0.463 | 0.526  | 0.248 | 0.289 | 0.400 | 0.491 | 0.804 |
| RNN            | 0.281   | 0.184 | 0.279 | 0.403 | 0.565  | 0.601     | 0.205 | 0.263 | 0.321 | 0.475 | 0.539  | 0.262 | 0.303 | 0.414 | 0.504 | 0.816 |
| ST-RNN         | 0.283   | 0.189 | 0.284 | 0.411 | 0.568  | 0.607     | 0.211 | 0.274 | 0.329 | 0.486 | 0.549  | 0.274 | 0.315 | 0.429 | 0.521 | 0.834 |
| LSTPM          | 0.296   | 0.194 | 0.286 | 0.429 | 0.574  | 0.609     | 0.218 | 0.278 | 0.334 | 0.492 | 0.558  | 0.279 | 0.319 | 0.438 | 0.529 | 0.847 |
| DeepMove       | 0.286   | 0.192 | 0.287 | 0.414 | 0.571  | 0.609     | 0.217 | 0.277 | 0.332 | 0.489 | 0.551  | 0.279 | 0.318 | 0.435 | 0.527 | 0.841 |
| STAN           | 0.314   | 0.201 | 0.295 | 0.433 | 0.581  | 0.614     | 0.223 | 0.281 | 0.349 | 0.498 | 0.573  | 0.293 | 0.326 | 0.443 | 0.532 | 0.850 |
## Appendix B: NL’s performances after re-ranking

See Table 4.

|                       | JC         |                  | LCST       |                  | OFE        |                  |
|-----------------------|------------|-----------------|------------|-----------------|------------|-----------------|
|                       | 0–20       | 20–40           | 40–60      | 60–80           | 80–100     | 0–20            | 20–40           | 40–60           | 60–80           | 80–100        |
| Gowalla               |            |                  |            |                 |            |                 |                  |                 |                 |              |
| RNN                   | 0.140      | 0.119            | 0.148      | 0.347           | 0.533      | 0.777           | 0.016            | 0.192           | 0.403          | 0.647         | 0.837         | 0.144         | 0.257          | 0.546          | 0.783          | 0.874          |
| ST-RNN                | 0.149      | 0.126            | 0.148      | 0.356           | 0.526      | 0.785           | 0.033            | 0.245           | 0.472          | 0.704         | 0.889         | 0.127         | 0.296          | 0.569          | 0.796          | 0.894          |
| DeepMove              | 0.165      | 0.184            | 0.176      | 0.391           | 0.541      | 0.803           | 0.053            | 0.224           | 0.435          | 0.681         | 0.870         | 0.151         | 0.285          | 0.578          | 0.823          | 0.901          |
| LSTPM                 | 0.171      | 0.182            | 0.178      | 0.379           | 0.544      | 0.799           | 0.043            | 0.225           | 0.452          | 0.684         | 0.869         | 0.151         | 0.276          | 0.573          | 0.822          | 0.899          |
| STAN                  | 0.205      | 0.188            | 0.206      | 0.405           | 0.615      | 0.814           | 0.059            | 0.229           | 0.458          | 0.709         | 0.889         | 0.146         | 0.267          | 0.578          | 0.832          | 0.912          |
| Foursquare NYC        |            |                  |            |                 |            |                 |                  |                  |                 |              |
| RNN                   | 0.233      | 0.051            | 0.265      | 0.479           | 0.651      | 0.741           | 0.158            | 0.276           | 0.598          | 0.531         | 0.869         | 0.241         | 0.742          | 0.712          | 0.926          | 0.988          |
| ST-RNN                | 0.261      | 0.059            | 0.287      | 0.563           | 0.722      | 0.861           | 0.186            | 0.328           | 0.552          | 0.590         | 0.920         | 0.299         | 0.787          | 0.779          | 0.928          | 0.989          |
| DeepMove              | 0.277      | 0.084            | 0.300      | 0.571           | 0.724      | 0.875           | 0.213            | 0.358           | 0.593          | 0.633         | 0.927         | 0.268         | 0.766          | 0.786          | 0.935          | 0.996          |
| LSTPM                 | 0.272      | 0.072            | 0.290      | 0.562           | 0.728      | 0.873           | 0.184            | 0.356           | 0.576          | 0.601         | 0.923         | 0.271         | 0.763          | 0.782          | 0.929          | 0.991          |
| STAN                  | 0.281      | 0.101            | 0.327      | 0.570           | 0.740      | 0.877           | 0.214            | 0.359           | 0.592          | 0.638         | 0.925         | 0.283         | 0.778          | 0.792          | 0.936          | 0.992          |
| Foursquare TKY        |            |                  |            |                 |            |                 |                  |                  |                 |              |
| RNN                   | 0.195      | 0.028            | 0.186      | 0.408           | 0.618      | 0.785           | 0.123            | 0.317           | 0.394          | 0.596         | 0.692         | 0.171         | 0.520          | 0.703          | 0.737          | 0.892          |
| ST-RNN                | 0.213      | 0.057            | 0.213      | 0.411           | 0.647      | 0.800           | 0.133            | 0.235           | 0.408          | 0.614         | 0.699         | 0.194         | 0.525          | 0.711          | 0.739          | 0.906          |
| DeepMove              | 0.223      | 0.060            | 0.208      | 0.452           | 0.698      | 0.856           | 0.142            | 0.246           | 0.437          | 0.622         | 0.715         | 0.200         | 0.540          | 0.718          | 0.743          | 0.931          |
| LSTPM                 | 0.230      | 0.074            | 0.209      | 0.488           | 0.704      | 0.858           | 0.151            | 0.338           | 0.444          | 0.643         | 0.719         | 0.236         | 0.564          | 0.723          | 0.744          | 0.935          |
| STAN                  | 0.236      | 0.093            | 0.209      | 0.501           | 0.701      | 0.865           | 0.153            | 0.329           | 0.447          | 0.642         | 0.726         | 0.239         | 0.553          | 0.728          | 0.749          | 0.938          |
| Taxi Porto            |            |                  |            |                 |            |                 |                  |                  |                 |              |
| RNN                   | 0.421      | 0.069            | 0.275      | 0.436           | 0.525      | 0.643           | 0.306            | 0.387           | 0.468          | 0.545         | 0.649         | 0.398         | 0.458          | 0.516          | 0.617          | 0.643          |
| ST-RNN                | 0.427      | 0.077            | 0.279      | 0.415           | 0.529      | 0.641           | 0.317            | 0.401           | 0.473          | 0.548         | 0.654         | 0.418         | 0.463          | 0.518          | 0.623          | 0.649          |
| LSTPM                 | 0.466      | 0.104            | 0.301      | 0.429           | 0.536      | 0.652           | 0.348            | 0.426           | 0.498          | 0.563         | 0.673         | 0.434         | 0.467          | 0.532          | 0.637          | 0.684          |
| DeepMove              | 0.457      | 0.095            | 0.287      | 0.423           | 0.529      | 0.650           | 0.337            | 0.422           | 0.482          | 0.559         | 0.668         | 0.419         | 0.466          | 0.524          | 0.629          | 0.661          |
| STAN                  | 0.483      | 0.111            | 0.316      | 0.436           | 0.542      | 0.652           | 0.355            | 0.431           | 0.502          | 0.563         | 0.680         | 0.440         | 0.474          | 0.537          | 0.631          | 0.692          |
|                | JC   |       |       |       |       |       |       |       |       |       |       |       |       |       |
|----------------|------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
|                | 0–20 | 20–40 | 40–60 | 60–80 | 80–100| 0–20  | 20–40 | 40–60 | 60–80 | 80–100| 0–20  | 20–40 | 40–60 | 60–80 | 80–100|
| Taxi SF        |      |       |       |       |       |       |       |       |       |       |       |       |       |       |       |
| RNN            | 0.288| 0.193 | 0.285 | 0.422 | 0.571 | 0.604 | 0.208 | 0.271 | 0.327 | 0.477 | 0.539 | 0.604 | 0.208 | 0.271 | 0.327 |
| ST-RNN         | 0.297| 0.200 | 0.294 | 0.433 | 0.575 | 0.608 | 0.225 | 0.278 | 0.333 | 0.488 | 0.549 | 0.608 | 0.225 | 0.278 | 0.333 |
| DeepMove       | 0.313| 0.202 | 0.292 | 0.449 | 0.581 | 0.610 | 0.230 | 0.282 | 0.338 | 0.493 | 0.558 | 0.610 | 0.230 | 0.282 | 0.338 |
| LSTPM          | 0.301| 0.199 | 0.297 | 0.433 | 0.578 | 0.613 | 0.238 | 0.282 | 0.338 | 0.493 | 0.552 | 0.613 | 0.238 | 0.282 | 0.338 |
| STAN           | 0.330| 0.208 | 0.304 | 0.461 | 0.592 | 0.617 | 0.233 | 0.288 | 0.355 | 0.503 | 0.574 | 0.617 | 0.233 | 0.288 | 0.355 |

Table 4 (continued)
### Appendix C: Relative improvements with reranker

See Table 5.

|                     | JC            |                     |                     |                     |                     |
|---------------------|---------------|---------------------|---------------------|---------------------|---------------------|
|                     | 0–20  | 20–40  | 40–60  | 60–80  | 80–100  | 0–20  | 20–40  | 40–60  | 60–80  | 80–100  | 0–20  | 20–40  | 40–60  | 60–80  | 80–100  |
| Gowalla RNN (%)     | 29.09  | 27.96  | 35.78  | 8.10   | 7.03    | 1.70   | 45.45  | 10.98  | 4.68   | 2.54   | 1.58   | 50.00  | 22.97  | 2.63   | 0.77   | 0.34   |
| ST-RNN (%)          | 8.76   | 13.51  | 15.63  | 0.28   | 2.53    | 0.38   | 17.86  | 25.64  | 16.26  | 6.99   | 4.71   | 19.81  | 36.41  | 1.97   | 0.38   | 0.90   |
| DeepMove (%)        | 4.43   | 58.62  | 23.94  | 4.83   | 0.93    | 0.63   | 47.22  | 6.16   | 0.23   | 0.00   | 0.12   | 32.46  | 26.11  | 0.87   | 0.12   | 0.22   |
| LSTPM (%)           | 12.50  | 61.06  | 26.24  | 2.71   | 3.03    | 0.76   | 34.38  | 10.84  | 5.85   | 2.24   | 0.93   | 36.04  | 26.03  | 0.70   | 0.49   | 0.22   |
| STAN (%)            | 6.77   | 51.61  | 12.57  | 1.76   | 9.63    | 0.12   | 15.69  | 3.62   | 2.45   | 0.71   | 0.11   | 22.69  | 15.58  | 0.36   | 0.11   |
| Foursquare NY (%)   | 9.39   | 96.15  | 42.47  | 1.48   | 12.47   | 0.14   | 19.70  | 1.85   | 18.65  | 0.57   | 0.00   | 26.18  | 21.64  | 0.00   | 0.00   | 0.00   |
| ST-RNN (%)          | 5.24   | 84.38  | 21.61  | 4.65   | 1.83    | 0.23   | 15.53  | 1.86   | 1.47   | 0.17   | 0.00   | 31.14  | 15.91  | 0.00   | 0.00   | 0.00   |
| DeepMove (%)        | 6.95   | 64.71  | 24.48  | 5.94   | 1.12    | 0.23   | 19.66  | 8.48   | 4.22   | 0.32   | 0.11   | 16.02  | 12.15  | 0.00   | 0.00   | 0.00   |
| LSTPM (%)           | 8.37   | 56.52  | 21.34  | 5.44   | 1.53    | 0.92   | 10.84  | 8.87   | 4.54   | 0.33   | 0.11   | 18.34  | 12.21  | 0.00   | 0.00   | 0.00   |
| STAN (%)            | 6.44   | 32.89  | 21.56  | 4.78   | 1.09    | 0.23   | 15.05  | 8.46   | 3.14   | 0.31   | 0.00   | 18.91  | 13.41  | 0.00   | 0.00   | 0.00   |
| Foursquare TKY (%)  | 6.56   | 40.00  | 21.57  | 4.62   | 1.48    | 0.26   | 21.78  | 54.63  | 3.41   | 0.51   | 0.00   | 15.54  | 12.07  | 0.00   | 0.00   | 0.00   |
| ST-RNN (%)          | 7.58   | 67.65  | 34.81  | 3.53   | 1.57    | 0.76   | 15.65  | 4.91   | 3.55   | 0.99   | 0.00   | 24.36  | 12.18  | 0.00   | 0.00   | 0.00   |
| LSTPM (%)           | 5.69   | 46.34  | 26.83  | 4.87   | 1.45    | 0.94   | 19.33  | 5.58   | 4.30   | 0.16   | 0.00   | 19.05  | 14.65  | 0.00   | 0.00   | 0.00   |
| DeepMove (%)        | 5.50   | 57.45  | 24.40  | 7.96   | 1.44    | 0.47   | 21.77  | 41.42  | 4.72   | 0.78   | 0.00   | 34.86  | 19.49  | 0.00   | 0.00   | 0.00   |
| STAN (%)            | 4.89   | 82.35  | 59.77  | 9.39   | 1.58    | 0.93   | 18.60  | 34.84  | 4.93   | 0.47   | 0.55   | 32.04  | 16.42  | 0.00   | 0.00   | 0.00   |
| Taxi Porto RNN (%)  | 2.68   | 4.55   | 2.23   | 9.55   | 1.74    | 0.78   | 4.44   | 1.57   | 1.52   | 0.55   | 0.00   | 1.79   | 0.00   | 0.00   | 0.00   | 0.00   |
| ST-RNN (%)          | 2.64   | 5.48   | 2.57   | 2.72   | 1.93    | 0.31   | 5.32   | 1.52   | 1.28   | 0.74   | 0.15   | 5.56   | 0.00   | 0.00   | 0.00   | 0.00   |
| DeepMove (%)        | 5.43   | 6.12   | 1.69   | 2.39   | 1.52    | 0.15   | 6.10   | 2.16   | 2.47   | 0.36   | 0.15   | 6.11   | 0.00   | 0.00   | 0.00   | 0.00   |
| LSTPM (%)           | 6.53   | 6.74   | 2.14   | 2.42   | 1.54    | 0.46   | 6.65   | 2.68   | 1.90   | 0.18   | 0.00   | 5.01   | 0.00   | 0.00   | 0.00   | 0.00   |
| STAN (%)            | 6.62   | 7.77   | 3.61   | 3.56   | 1.88    | 0.46   | 7.90   | 1.89   | 2.03   | 0.90   | 0.15   | 7.06   | 0.00   | 0.00   | 0.00   | 0.00   |
|                | JC   |               |                | LCST |               |                | OFE  |               |                |
|----------------|------|---------------|----------------|------|---------------|----------------|------|---------------|----------------|
|                | 0–20 | 20–40 | 40–60 | 60–80 | 80–100 | 0–20 | 20–40 | 40–60 | 60–80 | 80–100 | 0–20 | 20–40 | 40–60 | 60–80 | 80–100 |
| Taxi SF        |      |          |          |       |         |      |          |          |       |         |      |          |          |       |         |
| RNN (%)        | 2.49 | 4.89 | 2.15 | 4.71 | 1.06 | 0.50 | 1.46 | 3.04 | 1.87 | 0.42 | 0.00 | 5.34 | 0.00 | 0.00 | 0.00 | 0.00 |
| ST-RNN (%)     | 4.95 | 5.82 | 3.52 | 5.35 | 1.23 | 0.16 | 6.64 | 1.46 | 1.22 | 0.41 | 0.00 | 8.76 | 0.00 | 0.00 | 0.00 | 0.00 |
| LSTPM (%)      | 5.74 | 4.12 | 2.10 | 4.66 | 1.22 | 0.16 | 5.50 | 1.44 | 1.20 | 0.20 | 0.00 | 6.45 | 0.00 | 0.00 | 0.00 | 0.00 |
| DeepMove (%)   | 5.24 | 3.65 | 3.48 | 4.59 | 1.23 | 0.66 | 9.68 | 1.81 | 1.81 | 0.82 | 0.18 | 5.02 | 0.00 | 0.00 | 0.00 | 0.00 |
| STAN (%)       | 5.10 | 3.48 | 3.05 | 6.47 | 1.89 | 0.49 | 4.48 | 2.49 | 1.72 | 1.00 | 0.17 | 5.80 | 0.00 | 0.00 | 0.00 | 0.00 |
Appendix D: Models hyperparameters

See Table 6.

| Table 6 | Hyperparameters of the models |
|---------|-------------------------------|
|         | Gowalla | Foursquare NYC | Foursquare Tokyo | Taxi Porto | Taxi San Francisco |
| RNN     | lr: 0.005 epochs: 300 batch: 64 | lr: 0.005 epochs: 75 batch: 32 | lr: 0.005 epochs: 75 batch: 32 | lr: 0.005 epochs: 200 batch: 32 | lr: 0.005 epochs: 200 batch: 32 |
| ST-RNN  | lr: 0.001 epochs: 300 batch: 64 | lr: 0.001 epochs: 75 batch: 32 | lr: 0.001 epochs: 75 batch: 32 | lr: 0.001 epochs: 200 batch: 32 | lr: 0.001 epochs: 200 batch: 32 |
| DeepMove| lr: 0.001 epochs: 300 batch: 64 | lr: 0.001 epochs: 75 batch: 32 | lr: 0.001 epochs: 75 batch: 32 | lr: 0.001 epochs: 200 batch: 32 | lr: 0.001 epochs: 200 batch: 32 |
| LSTPM   | lr: 0.0001 epochs: 300 batch: 64 | lr: 0.0001 epochs: 75 batch: 32 | lr: 0.0001 epochs: 75 batch: 32 | lr: 0.0001 epochs: 200 batch: 32 | lr: 0.0001 epochs: 200 batch: 32 |
| STAN    | lr: 0.0001 epochs: 300 batch: 64 | lr: 0.0001 epochs: 75 batch: 32 | lr: 0.0001 epochs: 75 batch: 32 | lr: 0.0001 epochs: 200 batch: 32 | lr: 0.0001 epochs: 200 batch: 32 |
### Appendix E: Stability and statistical significance

See Table 7.

Table 7  Average relative improvements and relative standard deviation for all the models after the re-ranking on all the datasets

| Model     | JC   | LCST | OFE   |
|-----------|------|------|-------|
|           | 0–20 | 20–40| 40–60 | 60–80| 80–100| 0–20 | 20–40| 40–60 | 60–80| 80–100| 0–20 | 20–40| 40–60 | 60–80| 80–100|
| Gowalla RNN | 29.08% (±2.13) | 29.12% (±4.26) | 33.86% (±1.74) | 7.93% (±1.12) | 1.76% (±0.31) | 45.98% (±2.85) | 11.15% (±2.33) | 4.68% (±0.94) | 1.57% (±0.04) | 7.82% (±1.12) | 1.76% (±0.31) | 49.27% (±2.85) | 22.82% (±0.42) | 2.56% (±0.21) | 0.79% (±0.09) | 33.86% (±3.52) | 7.82% (±1.12) | 1.76% (±0.31) | 49.27% (±3.94) | 22.82% (±0.42) | 2.56% (±0.21) | 0.79% (±0.09) | 33.86% (±3.52) |
| ST-RNN     | 8.79% (±1.25) | 15.21% (±3.18) | 16.08% (±2.23) | 0.78% (±0.39) | 0.37% (±0.02) | 18.24% (±2.99) | 26.11% (±1.73) | 7.42% (±0.77) | 4.69% (±0.09) | 7.82% (±1.12) | 1.76% (±0.31) | 19.04% (±2.83) | 35.99% (±0.94) | 2.09% (±0.26) | 0.88% (±0.07) | 7.93% (±3.52) | 7.82% (±1.12) | 1.76% (±0.31) | 45.98% (±3.94) | 22.82% (±0.42) | 2.56% (±0.21) | 0.79% (±0.09) | 33.86% (±3.52) |
| Deep-Move  | 4.43% (±1.09) | 57.12% (±4.73) | 22.76% (±2.88) | 4.58% (±0.68) | 0.64% (±0.02) | 45.33% (±3.76) | 6.64% (±0.92) | 0.21% (±0.23) | 0.12% (±0.00) | 45.33% (±3.76) | 6.64% (±0.92) | 0.21% (±0.23) | 32.18% (±2.98) | 26.31% (±0.72) | 0.86% (±0.09) | 0.14% (±0.02) | 29.08% (±2.13) | 29.12% (±4.26) | 33.86% (±1.74) | 7.93% (±1.12) | 1.76% (±0.31) | 45.98% (±2.85) |
| LSTPM      | 12.10% (±0.84) | 59.88% (±3.96) | 28.01% (±3.39) | 2.73% (±0.21) | 0.76% (±0.01) | 35.86% (±3.12) | 10.26% (±1.88) | 0.86% (±0.39) | 0.93% (±0.01) | 35.86% (±3.12) | 10.26% (±1.88) | 0.86% (±0.39) | 35.34% (±3.37) | 25.88% (±1.06) | 0.82% (±0.19) | 0.49% (±0.01) | 6.98% (±1.12) | 54.22% (±4.06) | 13.17% (±2.83) | 1.76% (±0.45) | 0.12% (±0.00) | 15.92% (±2.61) |
| STAN       | 6.98% (±1.12) | 54.22% (±4.06) | 13.17% (±2.83) | 1.76% (±0.45) | 0.12% (±0.00) | 15.92% (±2.61) | 3.69% (±0.42) | 0.99% (±0.11) | 0.11% (±0.01) | 15.92% (±2.61) | 3.69% (±0.42) | 0.99% (±0.11) | 23.44% (±1.86) | 15.63% (±0.88) | 0.00% (±0.00) | 0.34% (±0.08) | 6.98% (±1.12) | 54.22% (±4.06) | 13.17% (±2.83) | 1.76% (±0.45) | 0.12% (±0.00) | 15.92% (±2.61) |
|                  | JC 0–20 | 20–40 | 40–60 | 60–80 | 80–100 | LCST 0–20 | 20–40 | 40–60 | 60–80 | 80–100 | OFE 0–20 | 20–40 | 40–60 | 60–80 | 80–100 |
|------------------|---------|-------|-------|-------|--------|-----------|-------|-------|-------|-------|----------|-------|-------|-------|-------|
| **Four-square**  |         |       |       |       |        |           |       |       |       |       |          |       |       |       |       |
| NYC RNN          | 9.06% (±0.82) | 94.73% (±5.17) | 41.52% (±4.31) | 1.62% (±0.53) | 1.22% (±0.12) | 0.13% (±0.04) | 19.99% (±1.22) | 1.89% (±18.31) | 0.54% (±1.13) | 0.01% (±0.09) | 25.46% (±2.57) | 21.39% (±0.96) | 0.00% (±0.00) | 0.00% (±0.00) |
| Deep-Move        | 5.33% (±0.65) | 83.91% (±3.83) | 23.84% (±4.55) | 5.03% (±0.99) | 1.88% (±0.21) | 0.21% (±0.03) | 14.84% (±1.27) | 1.86% (±14.1) | 0.19% (±0.04) | 0.00% (±0.00) | 30.68% (±2.57) | 16.14% (±1.03) | 0.00% (±0.00) | 0.00% (±0.00) |
| ST-RNN           | 7.07% (±0.89) | 65.35% (±2.39) | 24.08% (±3.17) | 5.75% (±0.84) | 1.14% (±0.15) | 0.24% (±0.06) | 19.17% (±1.06) | 7.97% (±4.37) | 0.35% (±0.08) | 0.10% (±0.02) | 16.45% (±1.88) | 11.88% (±0.86) | 0.01% (±0.01) | 0.00% (±0.00) |
| LSTPM            | 8.16% (±0.44) | 56.94% (±1.39) | 22.34% (±2.58) | 5.49% (±0.87) | 1.59% (±0.17) | 0.95% (±0.04) | 10.55% (±0.95) | 8.73% (±4.39) | 0.35% (±0.04) | 0.12% (±0.01) | 18.09% (±2.02) | 12.94% (±1.21) | 0.01% (±0.01) | 0.00% (±0.00) |
| STAN             | 6.29% (±0.57) | 33.01% (±2.52) | 21.93% (±3.41) | 4.41% (±0.79) | 1.01% (±0.12) | 0.25% (±0.03) | 15.39% (±0.96) | 8.48% (±3.51) | 0.33% (±0.06) | 0.00% (±0.00) | 19.02% (±1.84) | 13.02% (±1.14) | 0.00% (±0.00) | 0.00% (±0.00) |
|                | JC    |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
|----------------|------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
|                | 0-20 | 20-40 | 40-60 | 60-80 | 80-100 | 0-20 | 20-40 | 40-60 | 60-80 | 80-100 | 0-20 | 20-40 | 40-60 | 60-80 | 80-100 | 0-20 | 20-40 | 40-60 | 60-80 | 80-100 |
| **Four-square** |      |     |     |     |     |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
| TKY RNN        | 6.48% | 40.39% | 20.38% | 4.39% | 0.98% | 0.29% | 21.04% | 54.22% | 3.28% | 0.53% | 0.00% | 0.00% | 15.22% | 12.01% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% |
| ST-RNN         | 7.63% | 68.24% | 32.16% | 3.60% | 1.49% | 0.74% | 16.27% | 42.77% | 3.64% | 1.01% | 0.00% | 0.00% | 24.89% | 12.22% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% |
| LSTPM          | 5.73% | 45.88% | 27.73% | 4.43% | 1.48% | 0.91% | 18.74% | 7.59% | 4.71% | 0.15% | 0.00% | 0.00% | 19.95% | 14.68% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% |
| Deep-Move      | 5.48% | 58.67% | 24.38% | 8.38% | 1.65% | 0.48% | 22.68% | 40.84% | 4.86% | 0.78% | 0.00% | 0.00% | 34.71% | 19.73% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% |
| STAN           | 4.92% | 79.26% | 55.42% | 8.85% | 1.11% | 0.93% | 19.31% | 29.31% | 5.13% | 0.48% | 0.59% | 0.07% | 33.21% | 16.05% | 0.03% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% |
| Model  | RNN   | Taxi Porto | ST-RNN | DeepMove | LSTPM | STAN |
|--------|-------|------------|--------|----------|-------|------|
| JC     |       |            |        |          |       |      |
| 0–20  | 2.65% (±0.28) | 2.59% (±0.21) | 5.48% (±0.34) | 6.60% (±0.29) | 6.81% (±0.31) |
| 20–40 | 4.93% (±0.76)  | 5.61% (±0.69)  | 6.41% (±0.74)  | 6.12% (±0.99)  | 7.37% (±0.86)  |
| 40–60 | 2.39% (±0.43)  | 2.51% (±0.27)  | 1.73% (±0.29)  | 2.06% (±0.32)  | 3.73% (±0.31)  |
| 60–80 | 9.79% (±1.72)  | 2.36% (±0.82)  | 2.84% (±0.79)  | 2.86% (±0.81)  | 2.93% (±0.92)  |
| 80–100| 1.77% (±0.33)  | 1.88% (±0.36)  | 1.63% (±0.34)  | 1.85% (±0.42)  | 1.67% (±0.41)  |

| LCST   |       |            |        |          |       |      |
| 0–20  | 4.76% (±1.38)  | 5.48% (±1.59)  | 6.28% (±2.20)  | 6.93% (±2.73)  | 8.27% (±1.81)  |
| 20–40 | 1.38% (±0.69)  | 1.59% (±1.00)  | 2.20% (±0.72)  | 2.73% (±0.83)  | 1.81% (±0.74)  |
| 40–60 | (±0.42)        | (±0.37)        | (±0.36)        | (±0.39)        | (±0.19)        |
| 60–80 | 0.85% (±0.22)  | 0.34% (±0.07)  | 0.16% (±0.16)  | 0.17% (±0.02)  | 0.17% (±0.01)  |
| 80–100| (±0.06)        | (±0.03)        | (±0.03)        | (±0.02)        | (±0.01)        |

| OFE    |       |            |        |          |       |      |
| 0–20  | 2.86% (±3.84) | 5.56% (±2.11) | 5.92% (±1.84) | 5.09% (±1.12) | 7.76% (±1.37) |
| 20–40 | 0.02% (±0.01) | 0.03% (±0.01) | 0.05% (±0.02) | 0.02% (±0.01) | 0.02% (±0.01) |
| 40–60 | 0.00% (±0.00) | 0.00% (±0.00) | 0.00% (±0.00) | 0.00% (±0.00) | 0.00% (±0.00) |
| 60–80 | 0.00% (±0.00) | 0.00% (±0.00) | 0.00% (±0.00) | 0.00% (±0.00) | 0.00% (±0.00) |
| 80–100| 0.00% (±0.00) | 0.00% (±0.00) | 0.00% (±0.00) | 0.00% (±0.00) | 0.00% (±0.00) |
|       | JC          |            |            |            | LCST        |            |            |            | OFE         |            |            |            |
|-------|-------------|------------|------------|------------|-------------|------------|------------|------------|-------------|------------|------------|------------|
|       |             | 0–20       | 20–40      | 40–60      | 60–80       | 80–100     | 0–20       | 20–40      | 40–60       | 60–80       | 80–100     | 0–20       | 20–40      | 40–60      | 60–80       | 80–100     |
| Taxi SF | 2.66% (±0.49) | 4.73% (±0.59) | 2.18% (±0.28) | 5.17% (±0.90) | 1.13% (±0.29) | 0.52% (±0.11) | 1.61% (±3.17%) | 1.90% (±0.42%) | 0.00% (±0.02) | 5.83% (±0.92) | 0.00% (±0.00) | 0.00% (±0.00) | 0.00% (±0.00) |
| ST-RNN | 4.73% (±0.56) | 5.55% (±0.62) | 3.43% (±0.37) | 5.79% (±0.85) | 1.17% (±0.33) | 0.20% (±0.06) | 6.90% (±1.32%) | 1.18% (±0.41%) | 0.02% (±0.01) | 9.21% (±1.12) | 0.06% (±0.00) | 0.00% (±0.00) | 0.00% (±0.00) |
| LSTPM  | 5.96% (±0.59) | 4.41% (±0.83) | 2.22% (±0.33) | 4.28% (±0.77) | 1.48% (±0.32) | 0.19% (±0.04) | 5.63% (±1.40%) | 1.09% (±0.19%) | 0.00% (±0.02) | 6.98% (±1.45) | 0.00% (±0.00) | 0.00% (±0.00) | 0.00% (±0.00) |
| Deep-Move | 5.40% (±0.52) | 3.58% (±0.78) | 3.49% (±0.34) | 4.03% (±0.79) | 1.40% (±0.35) | 0.65% (±0.09) | 8.94% (±1.86%) | 1.88% (±0.80%) | 0.18% (±0.01) | 5.26% (±0.92) | 0.02% (±0.00) | 0.00% (±0.00) | 0.00% (±0.00) |
| STAN  | 5.33% (±0.88) | 3.42% (±0.56) | 3.08% (±0.26) | 6.99% (±0.97) | 1.81% (±0.36) | 0.44% (±0.04) | 4.51% (±2.26%) | 1.80% (±1.09%) | 0.17% (±0.02) | 5.76% (±0.49) | 0.00% (±0.00) | 0.00% (±0.00) | 0.00% (±0.00) |

We ran the experiments 5 times and we report the behavior of the models.
Appendix F: Overlap for each trajectory without bin classification

See Fig. 4.

Fig. 4 In this figure, we report the density function of the overlap of all the trajectories in the test set. While in the main paper we classify the trajectories into five bins, here we report the distribution of the overlap scores without a division in bins. Similarly, in "Appendix G", we report the accuracies of each trajectory without a bin division. This figure and "Appendix G" are necessary to show that the results presented in the main paper do not depend on the way we chose the five bins of overlap (namely, 0–20, 20–40, 40–60, 60–80, 80–100)
Appendix G: ACC@5 for each trajectory without bin classification

In this Appendix, we report all the accuracies for all the test trajectories in each dataset without the five-bin aggregation proposed in the main paper. Thus, for each trajectory, we measure the overlap with the training set and we measure its ACC@5. In the following plots, each point represents a trajectory. On the x-axis, we have the overlap score and on the y-axis, we have the relative ACC@5. The orange line in the plots is the trend line (Figs. 5, 6, 7, 8, 9).

Fig. 5 Performances of models on Gowalla
Fig. 6 Performances of models on Foursquare NYC
Fig. 7 Performances of models on Foursquare TKY
Fig. 8 Performances of models on Taxi Porto
Appendix H: Average normalized entropies of softmax layer

In this section, we report the average normalized entropies computed of the distribution provided by the softmax layer of the NLs as output when we input trajectories with a 0–20% overlap. As closer, the score is to 1, as uniformly distributed the underlying distribution is.

| Metric: JS | FSQ. NYC | FSQ. TKY | Gowalla | Taxi Porto | Taxi SF |
|------------|----------|----------|---------|------------|---------|
| RNN        | 0.86     | 0.81     | 0.93    | 0.95       | 0.93    |
| ST-RNN     | 0.91     | 0.89     | 0.90    | 0.88       | 0.91    |
| DeepMove   | 0.88     | 0.89     | 0.88    | 0.79       | 0.81    |
| STAN       | 0.94     | 0.93     | 0.94    | 0.88       | 0.91    |
| LSTPM      | 0.89     | 0.86     | 0.88    | 0.89       | 0.89    |

| Metric: LCST | FSQ. NYC | FSQ. TKY | Gowalla | Taxi Porto | Taxi SF |
|--------------|----------|----------|---------|------------|---------|
| RNN          | 0.81     | 0.79     | 0.82    | 0.80       | 0.84    |
| ST-RNN       | 0.86     | 0.90     | 0.88    | 0.81       | 0.85    |
| DeepMove     | 0.92     | 0.89     | 0.81    | 0.78       | 0.79    |
| Metric: LCST | FSQ. NYC | FSQ. TKY | Gowalla | Taxi Porto | Taxi SF |
|-------------|---------|---------|---------|------------|--------|
| STAN        | 0.84    | 0.84    | 0.82    | 0.85       | 0.81   |
| LSTPM       | 0.86    | 0.81    | 0.82    | 0.83       | 0.82   |
| Metric: OFE | FSQ. NYC | FSQ. TKY | Gowalla | Taxi Porto | Taxi SF |
| RNN         | 0.91    | 0.94    | 0.88    | 0.89       | 0.83   |
| ST-RNN      | 0.84    | 0.78    | 0.79    | 0.77       | 0.77   |
| DeepMove    | 0.81    | 0.83    | 0.83    | 0.82       | 0.86   |
| STAN        | 0.80    | 0.78    | 0.81    | 0.86       | 0.87   |
| LSTPM       | 0.91    | 0.90    | 0.91    | 0.84       | 0.83   |

**Author contributions** ML designed the methodology to compute the overlap and the rerank methodology. GB directed the study. All the authors contributed to interpreting the results and writing the paper. GB developed this work prior joining Amazon.

**Funding** Luca Pappalardo has been partially supported by EU project SoBigData++ grant agreement 871042 and NextGenerationEU - National Recovery and Resilience Plan (Piano Nazionale di Ripresa e Resilienza, PNRR), project “SoBigData.it - Strengthening the Italian RI for Social Mining and Big Data Analytics”, prot. IR0000013, avviso n. 3264 on 28/12/2021. Luca Pappalardo and Bruno Lepri acknowledge the support of the PNRR project FAIR - Future AI Research (PE00000013), under the NRRP MUR program funded by the NextGenerationEU.

**Availability of data and materials** All the data are publicly available and can be downloaded using the links at https://github.com/scikit-mobility/DeepLearning4HumanMobility.

**Code availability** The code used to compute the overlap can be found at https://github.com/MassimilianoLuca/overlap-processing the code of the models can be found at https://github.com/LibCity/Bigcity-LibCity.

**Declarations**

**Conflict of interest** The authors have no competing interests to declare that are relevant to the content of this article.

**Ethics approval** Not applicable.

**Consent to participate** Not applicable.

**Consent for publication** Not applicable.

**References**

Amichi, L., Viana, A. C., Crovella, M., & Loureiro, A. A. (2020). Understanding individuals’ proclivity for novelty seeking. In Proceedings of the 28th international conference on advances in geographic information systems (pp. 314–324).

Arora, N., Cabannes, T., Ganapathy, S.V., Li, Y., Mcafee, P., Nunkesser, M., Osorio, C., Tomkins, A., & Tsogtsuren, I. (2021). Quantifying the sustainability impact of google maps: A case study of salt lake city. arXiv:2111.03426

Barbosa, H., Barthelemy, M., Ghoshal, G., James, C. R., Lenormand, M., Louail, T., Menezes, R., Ramasco, J. J., Simini, F., & Tomasini, M. (2018). Human mobility: Models and applications. Physics Reports, 734, 1–74.
Barlacchi, G., Perentis, C., Mehrotra, A., Musolesi, M., & Lepri, B. (2017). Are you getting sick? Predicting influenza-like symptoms using human mobility behaviors. *EPJ Data Science, 27*, 1–15.

Blanc, L. (2015). David: Towards integration at last? The sustainable development goals as a network of targets. *Sustainable Development, 23*(3), 176–187. https://doi.org/10.1002/sd.1582

Böhm, M., Nanni, M., & Pappalardo, L. (2022). Gross polluters and vehicle emissions reduction. *Nature Sustainability, 5*(8), 699–707.

Burbey, I., & Martin, T. L. (2012). A survey on predicting personal mobility. *International Journal of Pervasive Computing and Communications, 8*, 5–22.

Calabrese, F., Di Lorenzo, G., & Ratti, C. (2010). Human mobility prediction based on individual and collective geographical preferences. In *13th International IEEE conference on intelligent transportation systems* (pp. 312–317).

Canzian, L., & Musolesi, M. (2015). Trajectories of depression: Unobtrusive monitoring of depressive states by means of smartphone mobility traces analysis. In *Proceedings of the 2015 ACM international joint conference on pervasive and ubiquitous computing* (pp. 1293–1304).

Chang, S., Zhang, Y., Han, W., Yu, M., Guo, X., Tan, W., Cui, X., Witbrock, M., Hasegawa-Johnson, M. A., & Huang, T. S. (2017). Dilated recurrent neural networks. *Advances in Neural Information Processing Systems, 30.*

Cho, E., Myers, S. A., Leskovec, J. (2011). Friendship and mobility: User movement in location-based social networks. In *Proceedings of the 17th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 1082–1090).

Comito, C. (2017). Where are you going? next place prediction from twitter. In *2017 IEEE international conference on data science and advanced analytics (DSAA)* (pp. 696–705). IEEE.

Comito, C. (2020). Next: A framework for next-place prediction on location based social networks. *Knowledge-Based Systems, 204*, 106205.

Cornacchia, G., Böhm, M., Mauro, G., Nanni, M., Pedreschi, D., & Pappalardo, L. (2022). How routing strategies impact urban emissions. In *Proceedings of the 30th international conference on advances in geographic information systems. SIGSPATIAL ‘22*. Association for Computing Machinery. https://doi.org/10.1145/3557915.3560977

do Couto Teixeira, D., Almeida, J. M., & Viana, A. C. (2021). On estimating the predictability of human mobility: The role of routine. *EPJ Data Science, 10*(1), 49.

Feng, J., Li, Y., Zhang, C., Sun, F., Meng, F., Guo, A., & Jin, D. (2018). Deepmove: Predicting human mobility with attentional recurrent networks. In *Proceedings of the 2018 world wide web conference* (pp. 1459–1468).

Gambs, S., Killijian, M.-O., & del Prado Cortez, M. N. (2010). Show me how you move and i will tell you who you are. In *Proceedings of the 3rd ACM SIGSPATIAL international workshop on security and privacy in GIS and LBS* (pp. 34–41).

Gambs, S., Killijian, M.-O., & del Prado Cortez, M. N. (2012). Next place prediction using mobility Markov chains. In *Proceedings of the first workshop on measurement, privacy, and mobility* (pp. 1–6).

Gao, Q., Zhou, F., Trajcevski, G., Zhang, K., Zhong, T., & Zhang, F. (2019). Predicting human mobility via variational attention. In *The world wide web conference* (pp. 2750–2756).

Hofman, J. M., Sharma, A., & Watts, D. J. (2017). Prediction and explanation in social systems. *Science, 355*(6324), 486–488.

Kawaguchi, K., Kaelbling, L. P., & Bengio, Y. (2017). Generalization in deep learning. arXiv preprint arXiv:1710.05468

Khaidem, L., Luca, M., Yang, F., Anand, A., Lepri, B., & Dong, W. (2020). Optimizing transportation dynamics at a city-scale using a reinforcement learning framework. *IEEE Access, 8*, 171528–171541.

Kong, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980

Kong, D., & Wu, F. (2018). Hst-lstm: A hierarchical spatial-temporal long-short term memory network for location prediction. In *IJCAI* (pp. 2341–2347).

Kroll, C., Warchold, A., & Pradhan, P. (2019). Sustainable development goals (SDGS): Are we successful in turning trade-offs into synergies? *Palgrave Communications, 5*(1), 1–11. https://doi.org/10.1057/s41599-019-0335-5

Kulkarni, V., Mahalunkar, A., Garbinato, B., & Kelleher, J. D. (2019). Examining the limits of predictability of human mobility. *Entropy, 21*(4), 432.

Lewis, P., Stenetorp, P., & Riedel, S. (2020). Question and answer test-train overlap in open-domain question answering datasets. arXiv preprint arXiv:2008.02637.

Liu, L., Lewis, P., Riedel, S., & Stenetorp, P. (2012). Challenges in generalization in open domain question answering.
Liu, Q., Wu, S., Wang, L., & Tan, T. (2016). Predicting the next location: A recurrent model with spatial and temporal contexts. In Thirtieth AAAI conference on artificial intelligence.

Luca, M., Barlacchi, G., Leperi, B., & Pappalardo, L. (2021). A survey on deep learning for human mobility. ACM Computing Surveys, 55(1), 1–44. https://doi.org/10.1145/3485125

Luo, Y., Liu, Q., & Liu, Z. (2021). Stan: Spatio-temporal attention network for next location recommendation. In Proceedings of the web conference 2021 (pp. 2177–2185).

Moreira-Matias, L., Gama, J., Ferreira, M., Mendes-Moreira, J., & Damas, L. (2013). Predicting taxi-passerger demand using streaming data. IEEE Transactions on Intelligent Transportation Systems, 14(3), 1393–1402.

Pappalardo, L., Simini, F., Barlacchi, G., & Pellungrini, R. (2022). Scikit-mobility: A Python library for the analysis, generation, and risk assessment of mobility data. Journal of Statistical Software, 103(4), 1–38. https://doi.org/10.18637/jss.v103.i04

Pappalardo, L., Cornacchia, G., Navarro, V., Bravo, L., & Ferres, L. (2023). A dataset to assess mobility changes in Chile following local quarantines. Scientific Data, 10(1), 6.

Pappalardo, L., Simini, F., Rinzivillo, S., Pedreschi, D., Giannotti, F., & Barabási, A.-L. (2015). Returners and explorers dichotomy in human mobility. Nature Communications, 6(1), 1–8.

Pappalardo, L., Vanhoof, M., Gabrielli, L., Smoreda, Z., Pedreschi, D., & Giannotti, F. (2016). An analytical framework to nowcast well-being using mobile phone data. International Journal of Data Science and Analytics, 2, 75–92.

Piorkowski, M., Sarafijanovic-Djukic, N., & Grossglauser, M. (2009). CRAWDAD dataset epfl/mobility (v. 2009-02-24). Downloaded from https://doi.org/10.15783/C7J010

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning internal representations by error propagation (pp. 318–362). MIT Press.

Schläpfer, M., Dong, L., O’Keeffe, K., Santi, P., Szell, M., Salat, H., Ankesaria, S., Vazifeh, M., Ratti, C., & West, G. B. (2021). The universal visitation law of human mobility. Nature, 593(7860), 522–527.

Sen, P., & Saffari, A. (2020). What do models learn from question answering datasets? arXiv preprint arXiv:2004.03490

Shi, Y., Feng, H., Geng, X., Tang, X., & Wang, Y. (2019). A survey of hybrid deep learning methods for traffic flow prediction. In Proceedings of the 2019 3rd international conference on advances in image processing (pp. 133–138).

Smolak, K., Siła-Nowicka, K., Delvenne, J.-C., Wierzbiński, M., & Rohm, W. (2021). The impact of human mobility data scales and processing on movement predictability. Scientific Reports, 11(1), 1–10.

Song, C., Qu, Z., Blumm, N., & Barabási, A.-L. (2010). Limits of predictability in human mobility. Science, 327, 1018–1021.

Sun, K., Qian, T., Chen, T., Liang, Y., Nguyen, Q. V. H., & Yin, H. (2020). Where to go next: Modeling long-and short-term user preferences for point-of-interest recommendation. In Proceedings of the AAAI conference on artificial intelligence (Vol. 34, pp. 214–221).

Trasarti, R., Guidotti, R., Monreale, A., & Giannotti, F. (2017). Myway: Location prediction via mobility profiling. Information Systems, 64, 350–367.

Voukelatou, V., Gabrielli, L., Miliou, I., Cresci, S., Sharma, R., Tesconi, M., & Pappalardo, L. (2020). Measuring objective and subjective well-being: Dimensions and data sources. International Journal of Data Science and Analytics, 11, 279–309.

Wang, X., Girshick, R., Gupta, A., & He, K. (2018). Non-local neural networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 7794–7803).

Wang, J., Jiang, J., Jiang, W., Li, C., & Zhao, W. X. (2021). Libcity: An open library for traffic prediction. In Proceedings of the 29th international conference on advances in geographic information systems. SIGSPATIAL ’21, pp. 145–148). Association for Computing Machinery. https://doi.org/10.1145/3474717.3483923

Yang, Y., Yao, Q., Kwok, J. T., & Ni, L. M. (2020). Generalizing from a few examples: A survey on few-shot learning. ACM Computing Surveys (CSUR), 53(3), 1–34.

Wu, R., Luo, G., Shao, J., Tian, L., & Peng, C. (2018). Location prediction on trajectory data: A review. Big Data Mining and Analytics, 1, 108–127.

Yang, D., Fankhauser, B., Rosso, P., & Cudre-Mauroux, P. (2020). Location prediction over sparse user mobility traces using RNNs: Flashback in hidden states! In Proceedings of the twenty-ninth international joint conference on artificial intelligence (IJCAI-20, pp. 2184–2190).

Yang, D., Zhang, D., Zheng, V. W., & Yu, Z. (2014). Modeling user activity preference by leveraging user spatial temporal characteristics in LBSNS. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 45(1), 129–142.
Yao, D., Zhang, C., Huang, J., & Bi, J. (2017). SerM: A recurrent model for next location prediction in semantic trajectories. In Proceedings of the 2017 ACM on conference on information and knowledge management (pp. 2411–2414).

Zhang, C., Zhao, K., & Chen, M. (2022). Beyond the limits of predictability in human mobility prediction: Context-transition predictability. IEEE Transactions on Knowledge and Data Engineering, 35, 4514–4526.

Zhao, L. (2020). Event prediction in big data era: A systematic survey. arXiv preprint arXiv:2007.09815

Zheng, X., Han, J., & Sun, A. (2018). A survey of location prediction on twitter. IEEE Transactions on Knowledge and Data Engineering, 30(9), 1652–1671.

Zhou, K., Liu, Z., Qiao, Y., Xiang, T., & Loy, C. C. (2021). Domain generalization: A survey. arXiv preprint arXiv:2103.02503

Zhu, W.-Y., Peng, W.-C., Chen, L.-J., Zheng, K., & Zhou, X. (2015). Modeling user mobility for location promotion in location-based social networks. In Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining (pp. 1573–1582).

**Publisher’s Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.