Region Invariant Normalizing Flows for Mobility Transfer

Vinayak Gupta  
IIT Delhi  
vinayak.gupta@cse.iitd.ac.in

Srikanta Bedathur  
IIT Delhi  
srikanta@cse.iitd.ac.in

ABSTRACT
There exists a high variability in mobility data volumes across different regions, which deteriorates the performance of spatial recommender systems that rely on region-specific data. In this paper, we propose a novel transfer learning framework called ReformD, for continuous-time location prediction for regions with sparse checkin data. Specifically, we model user-specific checkin-sequences in a region using a marked temporal point process (MTPP) with normalizing flows to learn the inter-checkin time and geo-distributions. Later, we transfer the model parameters of spatial and temporal flows trained on a data-rich origin region for the next check-in and time prediction in a target region with scarce checkin data. We capture the evolving region-specific checkin dynamics for MTPP and spatial-temporal flows by maximizing the joint likelihood of next checkin with three channels (1) checkin-category prediction, (2) checkin-time prediction, and (3) travel distance prediction. Extensive experiments on different user mobility datasets across the U.S. and Japan show that our model significantly outperforms state-of-the-art methods for modeling continuous-time sequences. Moreover, we also show that ReformD can be easily adapted for product recommendations i.e., sequences without any spatial component.

CCS CONCEPTS
- Information systems → Location based services.

KEYWORDS
Normalizing Flows, POI Recommendation, Transfer Learning

1 INTRODUCTION
Recent research has shown that accurate advertisements on Points-of-Interest (POI) networks, such as Foursquare and Instagram, can achieve up to 25 times the return-on-investment [24]. Consequently, predicting the time-evolving mobility of users, i.e., where and when, is of utmost importance to power systems relying on spatial data.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2021 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-8446-9/21/11. . . $15.00
https://doi.org/10.1145/3459637.3482169

Current approaches [7, 18, 38] overlook the temporal aspect of a recommender system as it involves modeling continuous-time checkin sequences – which is non-trivial with standard neural architectures [10, 23, 40]. The problem is further aggravated by the variation in volumes of mobility data across regions due to the growing awareness for personal data privacy [3, 33]. Therefore, there exists an underlying region-based data-scarcity, which further exacerbates the training of large neural models.

In recent years, Marked Temporal Point Processes (MTPP) have outperformed other neural architectures for characterizing asynchronous events localized in continuous time and are even used in a wide range of applications, including healthcare [29], finance [1, 39], and social networks [15, 23, 40]. Recent works that deploy MTPP for predicting user mobility patterns are either: (i) limited to predicting the time of user-location interactions rather than actual locations [37], (ii) restricted to one dataset without a foreseeable way to easily utilize external information [19], or (iii) disregard the opportunity to reuse trained parameters from external datasets by jointly embedding the checkin and time distributions [5]. Thus, none of these approaches can be used for designing mobility prediction models for limited data regions.

In this paper, we present ReformD (Reusable Flows for Mobility Data), a novel transfer learning framework that learns spatial and temporal distribution of checkins using normalizing flows (NFs) on a checkin-rich origin region and transfers them for efficient prediction in a checkin-scarce target region. Specifically, we consider the series of checkins made by a user as her checkin sequence and model these sequences for all users from a region using a neural MTPP and learn the inter-checkin time interval and spatial-distance distributions as two independent NFs [22, 28]. To make the learned spatial and temporal NFs invariant of the underlying region, we restrict our model to learn the distribution of inter-checkin time intervals and spatial distance. These features are unaffected by the network characteristics that vary across regions – POI categories and user affinities towards these POIs. Therefore, these NFs can be easily extended for prediction in other mobility regions. The ability of NFs to provide faster sampling and closed-form training for continuous-time event sequences [30] make them a perfect medium to transfer mobility information. Moreover, for transferring across regions, we cluster the checkin sequences of each region, with each cluster containing checkin-sequences with similar spatial and temporal checkin patterns and only transfer the parameters across these clusters.

In summary, the key contributions we make in this paper via ReformD are three-fold: (i) We propose ReformD, a transfer-learning model for predicting mobility dynamics in checkin-scarce datasets by incorporating mobility parameters trained on a checkin-rich region. (ii) We present a novel NF-based transfer over the MTPP that not only enables a faster sampling of time and distance features of next checkin, but also achieves high performance even with limited
fine-tuning on the target region. (iii) Finally, we empirically show that ReFormD outperforms the state-of-the-art models by up to 20% and 23% for check-in-category and time prediction and can easily be extended to product recommendation datasets.

2 RELATED WORK

The key related works fall into the following categories:

Mobility Prediction: Recent sequential POI prediction models consider the check-in trajectory for each user as a sequence of events and utilize an RNN based learning [6, 21, 38] with some variants that incorporate the spatial features as well [12, 20]. Another approach [13] is a generic model for predicting user trajectories as well as next product recommendation. Recent approaches for check-in time prediction are limited to a single dataset [5, 19, 37]. They also model event-times as random variables rather than sequential flows and thus cannot be used for transfer across regions.

Temporal Point Process: In recent years TPPs have emerged as a powerful tool to model asynchronous events localized in continuous time [8, 16], which have a wide variety of applications, e.g. information diffusion, disease modeling, finance, etc. Driven by these motivations, in recent years, there has been a surge of works on TPPs [11, 15, 29]. Modeling the event sequences via a neural network led to further developments including neural Hawkes process [23] and several other neural models of TPPs [26, 34, 35], but cannot incorporate heterogeneous features as in a spatial networks. The approach most similar to our model is [30] that learns the inter-event time and space densities using a three-stage architecture:

\[
\Delta_{t,k} \sim \mathcal{LN}(\mu_t, \sigma^2_t), \quad \Delta_{d,k} \sim \mathcal{LN}(\mu_d, \sigma^2_d),
\]

The overall schematic of TPPs is given in Figure 1.

3 PROBLEM SETUP

We consider the mobility records for two regions with non-overlapping locations and users, origin and target as \(\mathcal{R}^{or} \) and \(\mathcal{R}^{tg} \) respectively. For any region, we represent a user trajectory as a sequence of check-ins represented by \( \mathcal{S}_k = [c_i = (t_i, d_i, c_i)] | i \in [k], t_i < t_{i+1}, d_i \in \mathbb{R}^+ \) the total distance traveled, and \( c_i \in C \) is a discrete category of the \( t \)-th check-in with \( C \) as the set of all categories, and \( \mathcal{S}_k \) denotes the first \( k \) check-ins. We represent the inter-check-in times and distances as, \( \Delta_{t,k} = t_k - t_{k-1} \) and \( \Delta_{d,k} = d_k - d_{k-1} \) respectively and model their distribution using NFs. Our goal is to capture these region invariant dynamics in origin region for mobility prediction in target region, i.e. given the check-in sequence for target region, \( \mathcal{S}_k^{tg} \) and the MTPP trained on origin, we aim to predict the time and category of the next check-in, \( c_{k+1}^{tg} \).

4 MODEL DESCRIPTION

We divide the working of ReFormD into two parts: (i) the neural MTTP to capture mobility dynamics specific to a region, and (ii) transfer of NFs trained on the origin region to the target region. The overall schematic of ReFormD is given in Figure 1.

4.1 Region-Specific MTTP

We model the check-in sequences using an MTTP that we build on a recurrent neural network (RNN). The RNN is used to obtain time-conditioned vector representation of sequences as in [10, 23, 26]. Later, via these embeddings we estimate the mark distribution and inter-event time and space densities using a three-stage architecture:

\[
\Delta_{t,k+1} \sim \mathcal{LN}(\mu_{t,k+1}, \sigma^2_{t,k+1}), \quad \Delta_{d,k+1} \sim \mathcal{LN}(\mu_{d,k+1}, \sigma^2_{d,k+1}),
\]

Figure 1: Architecture of ReFormD with flow-based transfer between origin region (red) and target region (blue).

Input stage: In this stage we represent the incoming checkin at index \( k \), \( c_k \) using a suitable vector embedding, \( \mathbf{v}_k \) as:

\[
\mathbf{v}_k = \mathbf{w}_c c_k + \mathbf{w}_t \Delta_{t,k} + \mathbf{w}_d \Delta_{d,k} + \mathbf{b}_v,
\]

where \( \mathbf{w}_*, \mathbf{b}_* \) are trainable parameters and \( \mathbf{v}_k \) denotes the vector embedding for checkin \( c_k \) respectively.

Update stage: In this stage, we update the hidden state representation of the RNN to include the current checkin \( c_k \) as:

\[
s_k = \tanh(g_s s_k - 1 + g_v \mathbf{v}_k + g_1 \Delta_{t,k} + g_d \Delta_{d,k} + \mathbf{b}_s),
\]

where \( g_s, b_s \) are trainable parameters and \( s_k \) denotes the RNN hidden state, i.e. a cumulative embedding for all previous checkins till the current time \( t_k \).

Output stage: Given the trajectory embedding \( s_k \), we predict the next checkin time and the checkin category. Unlike [10, 23] that learn the time distribution using the RNN hidden state, we model the density of arrival times using a LogNormal [30] flow denoted as \( \mathcal{p}_t(\Delta_{t,k+1} | s_k) \) conditioned on \( s_k \) as:

\[
\mathcal{p}_t(\Delta_{t,k+1} | s_k) = \text{LogNormal}(\mu_t(s_k), \sigma^2_t(s_k)),
\]

with \( \mu_t(s_k), \sigma^2_t(s_k) = [W_{\mu} s_k + \mu_t, W_{\sigma}^2 s_k + \sigma^2_t] \) denote the mean and variance of the time distribution. Such a formulation reduces model complexity, facilitates faster training and sampling in a closed-form [30].

To predict the time of the next checkin, we sample the probable time difference between the current and the next checkin as \( \Delta_{t,k+1} \sim \mathcal{LN}(\mu_t(s_k), \sigma^2_t(s_k)) \), where \( \mathcal{LN} \) denotes the learned log-normal parameters. The actual time of the next checkin is the sum of the sampled time difference and the current checkin time, \( t_{k+1} = t_k + \Delta_{t,k+1} \). Similar to the temporal flow, we also model the inter-check-in density of spatial distances using a log-normal denoted as \( \mathcal{p}_d(\Delta_{d,k+1} | s_k) \). We interpret this distribution as the spatial flow for a region.

The inter-location spatial distance plays a crucial role in determining the next POI [7, 21]. Unlike time, the distances between two checkin locations are unchanged throughout the data. Previous approaches [10, 23] ignore these spatial features and rely solely on the past checkin-categories. Moreover, in a sequential setting the distance that the user will travel for her next checkin is not known. Our MTPPs, being generative models, and spatial flows overcome this drawback as we can sample the probable travel distance for the next checkin from the spatial flow as \( \Delta_{d,k+1} \sim \mathcal{LN}(\mu_d(s_k), \sigma^2_d(s_k)) \). Then, for predicting the next checkin, we use the sampled distance \( \Delta_{d,k+1} \) and RNN hidden state \( s_k \) via an attention-weighted embedding [2].

\[
s_k^{tg} = s_k + \alpha \cdot \mathbf{w}_f \Delta_{d,k+1},
\]
where $\alpha$, $w_f$ denote the attention weight, a trainable parameter and $s_k^u$ denotes the updated hidden state. We then predict the next checkin category as:

$$
\mathbb{P}(c_{k+1} = c | s_k^u) = \frac{\exp (V_{s,c} s_k^u + b_s,c)}{\sum_{c' \in C} \exp (V_{s,c'} s_k^u + b_s,c')},
$$

where $V_{s,c}, b_{s,c}$ are trainable parameters and $\bullet$ denotes the entry corresponding to a category. $\mathbb{P}(c_{k+1} = c | s_k^u)$ denotes the probability of next checkin being of category $c$ with $c \in C$.

**Optimization:** Given the set of all sequences $S$ for a region $R$, we maximize the joint likelihood for the next checkin, the log-normal density distribution of spatial, and temporal normalizing flows.

$$
\mathcal{L} = \sum_{S} \sum_{k=1}^{S} \log (\mathbb{P}(c_{k+1} | s_k^u) p_t(\Delta t_{i,k+1} | s_k) p_d(\Delta t_d_{i,k+1} | s_k)).
$$

where $\mathcal{L}$ denotes the joint likelihood, which we represent as the sum of the likelihoods for all user sequences. We learn the parameters of Reformd using Adam [17] optimizer.

### 4.2 Flow-based Transfer

For transferring the mobility parameters across the regions, we follow the standard transfer learning procedure [27, 31] of training exclusively on the origin region and then fine-tuning for the target region. However, the affinity of a user towards a POI evolves with time [7, 18]. For example, a POI with frequent user-checkins during the summer season might not be an attractive option in winters. We consider the time of checkin and category as event dynamics using an aggregation of historical events.

Table 1: Statistics of datasets used in our experiments. The origin region columns are followed by target regions.

| Property | NY | MI | NE | VI | TY | AI | CH | SA |
|---------|----|----|----|----|----|----|----|----|
| #Users or #Sequences (S) | 25.4k | 6.7k | 4.1k | 6.5k | 32.1k | 10.9k | 7.3k | 11.4k |
| Mean Length (\mu_S) | 57.17 | 66.21 | 48.56 | 56.33 | 61.72 | 56.60 | 63.60 | 53.08 |

### 5 EVALUATION

In this section, we conduct an empirical evaluation of Reformd. Specifically, we address the following research questions.

**RQ1** Can Reformd outperform state-of-the-art baselines for time and checkin prediction?

**RQ2** What is the advantage of transferring via normalizing flows?

**RQ3** Can we extend Reformd for non-spatial datasets?

For evaluating mobility prediction, we consider six POI datasets from the U.S. and Japan. All our models are implemented in TensorFlow on an NVIDIA Tesla V100 GPU and are made public at https://github.com/data-iitd/reformd.

### 5.1 Experimental Settings

**Dataset Description:** We use POI data from Foursquare [36] in United States(US) and Japan(JP) and for each country we construct 4 datasets: one with large check-in data and three with limited data. The statistics of all datasets is given in Table 1 with each acronym denoting the following region: (i) NY: New York(US), (ii) MI: Michigan(US), (iii) NV: Nevada(US), (iv) VI: Virginia(US), (v) TY: Tokyo(JP), (vi) CH: Chiba(JP), (vii) SA: Saitama(JP) and (viii) AI: Aichi(JP). We consider NY and TY as the origin regions and MI, NV, VI and CH, SA, AI as the corresponding target regions.

For each region, we consider the time of checkin and category as event time and mark and normalize the times based on the minimum and maximum event times. We set the embedding and RNN hidden dimension to 64 and $M = 3$ for all our experiments. Other values for the model parameter had negligible differences.

**Evaluation Protocol:** We split each stream of say N checkins $S_N$ into training and test set, where the training set (test set) consists of first 80% (last 20%) checkin. We evaluate models using standard metrics [10] of (i) mean absolute error (MAE) of predicted and actual checkin times $\frac{1}{|S|} \sum_{c_i \in S} ||t_i - \tilde{t}_i||$ and (ii) mark (checkin category) prediction accuracy (MPA), i.e., $\frac{1}{|S|} \sum_{c_i \in S} #(c_i = \tilde{c}_i)$. Here $t_i$ and $\tilde{c}_i$ are the predicted time and category of the i-th checkin. Moreover, the clustering of sequences into different sets is done based solely on the training data and using these thresholds we assign clusters to sequences in the test data.

### 5.2 Baselines

We compare the prediction performance of Reformd with the following state-of-the-art methods:

**RMTTPP** [10]: A recurrent neural network that models time-differences to learn a representation of the past events.

**NHP** [23]: Models an MTPP using continuous-time LSTMs for capturing the temporal evolution of sequences.

**SAHP** [39]: A self-attention model to learn the temporal dynamics using an aggregation of historical events.
5.3 Prediction Performance (RQ1)

We report the prediction performance of different methods across our target datasets in Table 2 and make the following observations:

- Reformd consistently yields the best performance on all the datasets. In particular, it improves over the strongest baselines by 10% and 19% for category and time prediction respectively. These results indicate the importance of spatial and temporal flow-based transfer from external data for prediction in limited-data regions.
- RMTPP [10] is the second-best performer in terms of MAE of time prediction almost for all the datasets. For some datasets, THP [40] outperforms RMTPP for mark category prediction. However, Reformd significantly outperforms it across all metrics.

Qualitative Analysis: We also perform a qualitative analysis to demonstrate how Reformd is able to model the checkin time distribution. For this, we plot the actual inter-checkin time differences and the difference time predicted by Reformd in Figure 2 for Virginia and Aichi datasets. From the results we note that the predicted inter-arrival times closely match with the true inter-arrival times and Reformd is even able to capture large time differences (peaks). For brevity, we omit the results for other datasets.

Table 2: Performance of all the methods in terms of mark prediction accuracy (MPA) and mean absolute error (MAE) across all datasets. Results marked † are statistically significant (i.e. two-sided t-test with \( p \leq 0.1 \)) over the best baseline.

| \( \mathcal{R}^\text{orig} \rightarrow \mathcal{R}^\text{tgt} \) | Mark Prediction Accuracy | Mean Absolute Error |
|-----------------------------------------------|--------------------------|----------------------|
| \( \mathcal{R}^\text{orig} \rightarrow \mathcal{R}^\text{orig} \) | | |
| NHP [23] | 0.1745 | 0.1672 | 0.1348 | 0.2162 | 0.4673 | 0.3820 |
| RMTPP [10] | 0.1761 | 0.1684 | 0.1577 | 0.2293 | 0.4250 | 0.4036 |
| SAHP [39] | 0.1587 | 0.1329 | 0.1303 | 0.1968 | 0.3864 | 0.3943 |
| THP [40] | 0.1793 | 0.1545 | 0.1493 | 0.2361 | 0.4229 | 0.4077 |
| Reformd | 0.2159† | 0.1868† | 0.1631† | 0.2586† | 0.4474 | 0.4208† |
| \( \Delta (\%) \) | 20.41 | 19.92 | 9.24 | 5.27 | 3.72 | 2.52 |

We consider Digital Music(\( \mathcal{S} = 7k \)) as origin and Appliances(\( \mathcal{S} = 6k \)) as target. From the results in Table 3, we note that even in the absence of spatial flows, Reformd outperforms other baselines across all metrics.

Table 3: Prediction performance of all the methods for product recommendation in Amazon datasets. Results marked † are statistically significant as in Table 2.

| \( \mathcal{R}^\text{orig} \rightarrow \mathcal{R}^\text{orig} \) | Mark Prediction Accuracy | Mean Absolute Error |
|-----------------------------------------------|--------------------------|----------------------|
| \( \mathcal{R}^\text{orig} \rightarrow \mathcal{R}^\text{orig} \) | | |
| NHP [23] | 0.8773 | 0.5711 | 0.0903 | 0.1795 |
| RMTPP [10] | 0.8975 | 0.5530 | 0.0884 | 0.1758 |
| SAHP [39] | 0.8931 | 0.5517 | 0.1439 | 0.2244 |
| THP [40] | 0.9084 | 0.5879 | 0.1253 | 0.2035 |
| Reformd | 0.9129 | 0.6035 | 0.0756† | 0.1564† |
| \( \Delta (\%) \) | 0.49 | 2.65 | 14.47 | 11.03 |

5.4 Transfer Advantage (RQ2)

Reformd outperforms other baselines and also brings exhibits a key feature of transfer learning, i.e. quick parameter learning [27, 31]. We highlight this by plotting the time prediction error (MAE) corresponding to the epochs trained on the target region for Reformd and the best time prediction model, i.e. RMTPP. Figure 3 summarizes the results for Virginia and Aichi. We note that Reformd exhibits faster convergence than RMTPP for both datasets. More specifically, the flow-based transfer procedure of Reformd can outperform most baselines even with a fine-tuning of a few epochs. The results also highlight the stable learning procedure of Reformd.

5.5 Product Recommendation (RQ3)

We further evaluate the performance of Reformd in product recommendation, i.e. without spatial coordinates. Consequently, we use purchase records for three item categories from Amazon [25], namely Digital Music(DM), Appliances(AP) and Beauty(BY). For each item, we use the user reviews as the events in a sequence with the time of the written review as the event time and the rating (1 to 5) as the corresponding mark. As in this case, we do not have a spatial density function \( p_d(\Delta t, k) \), we change the fusion equation 4 to include the predicted time of next purchase as:

\[
s_k = s_k + \alpha \cdot w_f \Delta t, k+1, \tag{8}
\]

We consider Digital Music(\( \mathcal{S} = 12k \)) as origin and Appliances(\( \mathcal{S} = 7k \)) and Beauty(\( \mathcal{S} = 6k \)) as target. From the results in Table 3, we note that even in the absence of spatial flows, Reformd outperforms other baselines across all metrics.

6 CONCLUSION AND FUTURE WORK

In this paper, we present Reformd, a novel method for transferring mobility knowledge across regions by sharing the spatial and temporal NFs for continuous-time checkin prediction. As a future work, we plan to incorporate meta-learning based transfer [14].
REFERENCES

[1] Emmanuel Bacry, Iacopo Mastromatteo, and Jean-François Muzy. 2015. Hawkes processes in finance. In Market Microstructure and Liquidity.
[2] Dumitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In ICML.
[3] Louise Barkhuus and Anind K. Dey. 2003. Location-Based Services for Mobile Telephony: a Study of Users Privacy Concerns. In Interact.
[4] E Barouch, GM Kaufman, and ML Glasser. 1986. On sums of lognormal random variables. Studies in Applied Mathematics 75. 1 (1986), 37–55.
[5] Yile Chen, Cheng Long, Gao Cong, and Chenliang Li. 2020. Context-aware Deep Model for Joint Mobility and Time Prediction. In WSDM.
[6] Chen Cheng, Haigun Yang, Michael R Lyons, and Irwin King. 2013. Where you like to go next: Successive point-of-interest recommendation. In IJCAI.
[7] Junho Cho, Seth A. Myers, and Jure Leskovec. 2011. Friendship and mobility: User movement in location-based social networks. In KDD.
[8] Daryl J Daley and David Vere-Jones. 2007. An introduction to the theory of point processes: volume II: general theory and structure. Springer Science & Business Media.
[9] Dufresne Daniel. 2004. The log-normal approximation in financial and other computations. In Advances in Applied Probability.
[10] Nan Du, Hanjun Dai, Rakshit Trivedi, Utkarsh Upadhyay, Manuel Gomez-Rodriguez, and Le Song. 2016. Recurrent marked temporal point processes: Embedding event history to vector. In KDD.
[11] Mehrdad Farajtabar, Jiachen Yang, Xiaojing Ye, Huan Xu, Rakshit Trivedi, Elias Khalil, Shuang Li, Le Song, and Hongyuan Zha. 2017. Fake news mitigation via point process based intervention. In ICML.
[12] Jie Feng, Yong Li, Chao Zhang, Funing Sun, Fanchao Meng, Ang Guo, and Depeng Jin. 2018. DeepMove: Predicting Human Mobility with Attentional Recurrent Networks. In WWW.
[13] Flavio Figueiredo, Bruno Ribeiro, Jussara A. Almeida, and Christos Faloutsos. 2016. TribeFlow: mining & predicting user trajectories. In WWW.
[14] Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In ICML.
[15] Vinayak Gupta, Srikantha Bedathur, Sourangsu Bhattacharya, and Abir De. 2021. Learning Temporal Point Processes with Intermittent Observations. In AISTATS.
[16] Alan G Hawkes. 1971. Spectra of some self-exciting and mutually exciting point processes. In Biometrika.
[17] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In ICLR.
[18] Ankita Likhyani, Vinayak Gupta, PK Srijith, P Deepak, and Srikantha Bedathur. 2020. Modeling Implicit Communities from Geo-tagged Event Traces using Spatio-Temporal Point Processes. In WISE.