CrossBag: A Bag of Tricks for Cross-City Mobility Prediction

JangHyeon Lee
lee04588@umn.edu
University of Minnesota, Twin Cities
Minneapolis, Minnesota, USA

Yao-Yi Chiang
yaoyi@umn.edu
University of Minnesota, Twin Cities
Minneapolis, Minnesota, USA

Abstract

Access to large-scale human trajectory data has significantly advanced the understanding of human mobility (HuMob) behavior for urban planning. However, these data are often concentrated in major cities, leaving smaller or less-monitored areas with limited information, undermining the performance of data-hungry machine learning models for HuMob prediction. This imbalance poses a challenge for cross-city mobility prediction, as many existing models are designed for single-city settings. To address this, we present **CrossBag**, a set of simple yet effective techniques to boost cross-city prediction. These techniques include context-aware spatiotemporal embeddings, masking types, and a progressive knowledge transfer method to incrementally adapt the target model while preserving useful patterns from the source model for stable cross-city transfer. Additionally, we propose a test-time trajectory refinement method using top-K guided beam search to prevent predictors from getting *stuck* in repetitive location predictions. We validate CrossBag on the large-scale multi-city dataset from the HuMob Challenge 2024, achieving a top-10 placement out of over 100 participating teams.

CCS Concepts

• **Computing methodologies → Artificial intelligence**.

Keywords

Human mobility, Transfer learning, Spatiotemporal, Transformer

ACM Reference Format:
JangHyeon Lee and Yao-Yi Chiang. 2024. CrossBag: A Bag of Tricks for Cross-City Mobility Prediction. In *2nd ACM SIGSPATIAL International Workshop on the Human Mobility Prediction Challenge (HuMob'24), October 29-November 1 2024, Atlanta, GA, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3681771.3699935

1 Introduction

Understanding, modeling, and predicting human mobility (HuMob) in urban areas is crucial for transportation planning, disaster response, and urban development [7]. The availability of large-scale movement data from mobile devices has enabled machine learning models to capture complex spatiotemporal dependencies, which are essential for accurate HuMob prediction [5]. Early models, such as spatiotemporal Recurrent Neural Network (ST-RNN) [6] and DeepMove [3], utilize time and location embeddings to account for varying temporal and spatial contexts. Meanwhile, Spatiotemporal Long Short-Term Memory (ST-LSTM) [4] targets short-term predictions within minutes or hours, bridging the gap between long-term recommendations and real-time forecasting. To address the limitations of sequential RNNs and LSTMs, transformer models have been introduced for more efficient long-term time series modeling [9]. However, these models are primarily designed for single-city scenarios, limiting their generalization across different urban contexts. Furthermore, movement data are often concentrated in major cities [13], leaving smaller regions with insufficient data coverage.

This raises a key question: *Can data from other cities improve the predictability of human mobility in a data-limited city?* This challenge, known as cross-city mobility prediction, is complicated because each city has unique road networks and transit systems, resulting in distinct mobility patterns [12]. Additionally, socioeconomic factors [2] such as population density, economic activity, and land use vary significantly across cities, further complicating generalization of mobility trends. For instance, in compact cities, individuals tend to walk shorter distances, whereas in sprawling areas, they travel farther and rely more on vehicles. Such differences make it essential to calibrate single-city models carefully to avoid overfitting to patterns that may not generalize well across diverse urban environments. A natural solution is to transfer knowledge from a data-rich city to a data-limited one using fine-tuning [14]. However, conventional fine-tuning approaches often suffer from catastrophic forgetting [1], where the model loses previously learned knowledge from the source domain while adapting to the target, resulting in weakened generalization.

In response, we present **CrossBag** to enhance cross-city mobility prediction. Our contributions are three-fold: First, we detail the design choices behind our spatiotemporal transformer, including contextual spatiotemporal embeddings and masking to capture richer dependencies across space and time. Second, we introduce Stable Knowledge Transfer (SKT), improving cross-city transfer stability with layer-wise learning rate control and gradual unfreezing to preserve useful prior knowledge. Third, we improve an existing probability suppression method to refine sequences during test-time, called Test-Time Trajectory Refinement (TrajRef), preventing repetitive location predictions. We validate our approach on the HuMob Challenge 2024 dataset.

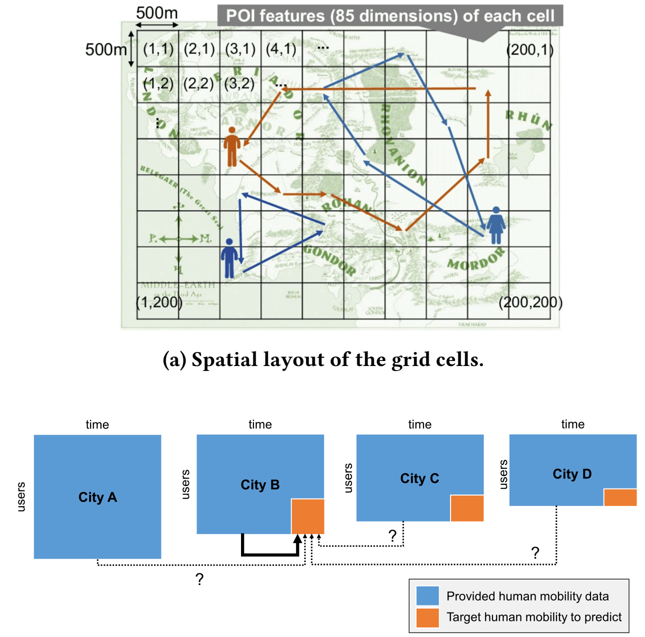
2 Problem Statement

The HuMob Challenge aims to overcome the lack of large-scale, open-source human mobility datasets, a key barrier to advancing human mobility models. The challenge provides the YJMob100K dataset, capturing mobility patterns across four metropolitan areas in Japan [15]. Each area is divided into a 200 x 200 grid of 500m x 500m cells (Figure 1a) over 75 days, discretized into 30-minute

HuMob'24, October 29-November 1 2024, Atlanta, GA, USA

ACM ISBN 979-8-4007-1150-3/24/10
https://doi.org/10.1145/3681771.3699935

intervals. The dataset includes full trajectories for 100,000 individuals in city A and partial trajectories for 25,000, 20,000, and 6,000 individuals in cities B, C, and D, respectively. Each city is also annotated with an 85-dimensional Point-of-Interest (POI) feature vector for each grid cell. The goal is to predict the movement of the 3,000 individuals in Cities B, C, and D for days 61 to 75 (Figure 1b).



(a) Spatial layout of the grid cells.

(b) Can mobility prediction be improved using data from other cities?

Figure 1: Schematic of the cross-city HuMob challenge.

3 Enhancing HuMob Predictors Across Cities

Building on previous work [5, 9, 11], we present **CrossBag**, a collection of complementary techniques to enhance cross-city human mobility models. First, our Spatiotemporal Transformer (STF) captures complex mobility patterns by encoding day, time, spatial coordinates, and POI features, using multi-head attention to preserve long-term temporal dependencies. Second, Stable Knowledge Transfer (SKT) enables effective cross-city adaptation by progressively fine-tuning models, minimizing catastrophic forgetting to retain critical prior knowledge. Finally, Test-Time Trajectory Refinement (TrajRef) addresses repetitive prediction errors by combining top-K sampling and beam search to explore diverse candidate paths for more reliable spatial predictions. Together, CrossBag improves generalization across diverse urban settings (Figure 2).

3.1 Spatiotemporal Transformer (STF)

To encode the spatiotemporal attributes of human mobility data, we construct separate embeddings for day, time, and location coordinates. The *day* embedding (E_{day}) has a category size of 75, representing the total number of days in the dataset, while the *time* embedding (E_{time}) has a category size of 48, corresponding to half-hour intervals over a 24-hour day (24 × 2). The *time difference* embedding ($E_{\Delta t}$) encodes temporal gaps between consecutive observations, capturing variations in sampling intervals to provide context for varying user mobility behaviors. The *location* embeddings (E_x and E_y) reflect the spatial grid cell counts in each direction. We obtain the unified spatiotemporal embedding by summing the individual embeddings:

$$E_{\text{ST}} = E_{\text{day}} + E_{\text{time}} + E_{\Delta t} + E_x + E_y \tag{1}$$

In addition to the spatiotemporal (ST) embeddings (E_{ST}), we incorporate POI features to enrich spatial context. To construct the POI embeddings, we first count the occurrences of each category within each grid cell and apply a logarithmic transformation to normalize these counts, balancing the influence of frequently visited places while preserving their importance. We project these POI feature vectors into the same embedding space as the spatiotemporal features using a linear layer. Finally, we integrate the POI embeddings with the spatiotemporal features by concatenating them to obtain the final embedding:

$$E_{\text{final}} = \text{Concat}(E_{\text{ST}}, E_{\text{POI}}) \tag{2}$$

The inclusion of POI embeddings allows the model to capture fine-grained spatial semantics and consider functional attributes of locations, potentially improving its ability to predict mobility patterns.

We pass the final embedding (E_{final}) into the STF encoder, which captures complex spatiotemporal dependencies in human mobility data. The encoder consists of multiple layers, each containing a multi-head self-attention mechanism and a feed-forward network (FFN). The attention mechanism enables STF to focus on important spatiotemporal patterns. Each encoder layer processes the hidden states sequentially. Once the input passes through L encoder layers, the final hidden state is directed to a task-specific FFN, which has two separate branches to predict the x- and y-coordinates. These branches allow STF to map the spatiotemporal features to separate spatial predictions: $\hat{x} = \text{FFN}_x(H^{(L)})$ and $\hat{y} = \text{FFN}_y(H^{(L)})$.

During training, we optimize the model using a cross-entropy loss function that focuses specifically on the masked x- and y-coordinate outputs. For each batch, we compute the loss solely on the masked entries, which drives the model's weight updates based on the errors in these regions. By focusing only on the masked regions, the model learns to infer missing values based on the context provided by the unmasked data, minimizing:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{CE}}(\hat{x}_{\text{masked}}, x_{\text{true}}) + \mathcal{L}_{\text{CE}}(\hat{y}_{\text{masked}}, y_{\text{true}}), \tag{3}$$

where $\hat{x}_{\text{masked}}, \hat{y}_{\text{masked}}$ are the predicted coordinates for the masked entries, and $x_{\text{true}}, y_{\text{true}}$ are the corresponding ground-truth values.

CrossBag presents two masking strategies: fixed masking and random fixed masking. Fixed masking consistently hides days 61 to 75, focusing on the model's ability to predict a 15-day segment at the end of each user's trajectory. This approach aligns with the competition goal of predicting the final portion of the trajectory, training the model to handle the most up-to-date movements. Random fixed masking selects a random 15-day window within each user's sequence, exposing the model to varied missing segments. For example, random fixed masking might choose days 30 to 45 for one trajectory and days 50 to 65 for another. This variability helps the model handle different temporal segments by learning from varied time windows.

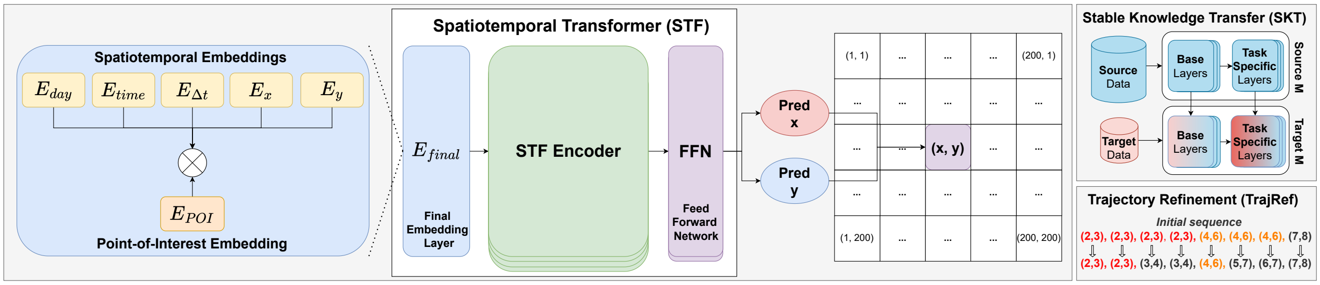


Figure 2: Overview of the pipeline.

3.2 Stable Knowledge Transfer (SKT)

Standard fine-tuning (FT) approaches often face two main issues when adapting a pretrained model from a source domain to a target domain with limited data: overfitting and catastrophic forgetting [1]. Overfitting occurs when the model aggressively adjusts to the small target dataset, losing the model's ability to generalize beyond the training data. Conversely, catastrophic forgetting happens when rapid updates to deeper layers overwrite valuable patterns learned from the source domain. The goal of a typical FT approach is to minimize the loss function $\mathcal{L}_{\text{target}}$ in the target domain:

$$\theta_{\text{target}} = \arg\min_{\theta} \mathcal{L}_{\text{target}}(\theta), \tag{4}$$

where θ represents all the model parameters. However, this direct minimization treats all layers equally, leading to unstable updates. In deep models, lower-level layers capture general features (e.g., spatiotemporal dependencies), while higher-level layers focus on task-specific characteristics. Applying uniform learning rates and optimizations across all layers risks overwriting these core lower-layer features, resulting in poor generalization and suboptimal performance in the target domain.

To address these limitations, we propose **S**table **K**nowledge **T**ransfer (**SKT**) that progressively updates different parts of the model in stages. We first partition the model parameters into two groups: θ_{frozen}, representing the base layers (e.g., embedding and encoder layers) that capture general patterns, and $\theta_{\text{trainable}}$, which consists of the task-specific layers (e.g., final feedforward) responsible for the predictions in the target domain. Initially, we only update $\theta_{\text{trainable}}$ while keeping θ_{frozen} fixed:

$$\theta_{\text{trainable}} = \arg\min_{\theta_{\text{trainable}}} \mathcal{L}_{\text{target}}(\theta_{\text{trainable}}, \theta_{\text{frozen}}). \tag{5}$$

This step allows task-specific layers to quickly adapt to the target domain using a higher learning rate. Meanwhile, the frozen layers retain the broader knowledge from the source domain, such as recurring behavioral patterns. As training progresses, we gradually unfreeze deeper layers. Each time a new layer is unfrozen, it is added to the set of trainable parameters and updated using a much lower learning rate than the task-specific layers. This ensures that newly trained layers are refined slowly, minimizing the risk of disrupting previously learned representations.

3.3 Test-Time Trajectory Refinement (TrajRef)

Models can overfit frequent mobility patterns, such as prolonged stays at high-probability locations like home or work, leading to repetitive predictions over consecutive time steps. As a result, the model can get "stuck" predicting the same location, failing to capture transitions to new places. To mitigate this, previous work introduced a probability suppression technique that reduces the model's confidence in predicting the same location repeatedly by multiplying its probability by 0.9 [9]. While this method discourages the model from getting trapped in repetitive predictions, it does not fully account for the broader sequence context. Consequently, it can over-penalize valid stationary behaviors (e.g., waiting at a bus stop) or lead to abrupt transitions to less realistic locations.

In response, we introduce **TrajRef** that combines top-K sampling and beam search during inference. First, we apply top-K sampling to select a set of the most likely candidates at each time step for (x, y) coordinates. This approach encourages the model to explore multiple high-probability options, rather than restricting it to a single most likely location, thereby reducing the risk of repetitive, overly deterministic predictions. Once top-K sampling narrows down the search space, beam search evaluates these candidates across the entire sequence to identify the most probable trajectory. Beam search keeps a fixed number of top-N sequences and updates them at each time step based on their cumulative probabilities. As the sequence progresses, it discards less likely paths and retains only those that maximize the overall probability, selecting the most plausible movement patterns given the input. The cumulative score for each sequence up to the time step t is defined as:

$$\text{Score}_{\text{beam}}^{(t)} = \sum_{k=1}^{t} \log(P(x_k, y_k)), \tag{6}$$

where $P(x_k, y_k)$ is the joint probability of predicting the coordinate pair (x_k, y_k) at step k. This cumulative score enables beam search to evaluate the entire sequence, ensuring that a candidate with slightly lower probability at a single step can still be selected if it results in a better overall trajectory. The complementary nature of top-K sampling and beam search allows the model to focus on a small set of high-quality candidates at each step while iteratively refining the whole sequence. We use grid search to select K and N, leveraging the parallel processing capabilities of Transformer models to maintain time efficiency.

4 Experimental Setup

4.1 Datasets

To recap, the goal is to predict the movement of the last 3,000 individuals in Cities B (`25,000` total UserIDs (UIDs)), C (`20,000` total UIDs), and D (`6,000` total UIDs) for days 61 to 75. Thus, we only have complete coverage data up to `UID = 21,999` in City B, `UID = 18,999` in City C, and `UID = 2,999` in City D for the full 75 days. City A (`100,000` total UIDs) has complete data for all days. Before creating the final predictions, we validate our approach using smaller training and validation subsets.

For **City A**, the training set includes `0` ≤ `UID` ≤ `79,999` for days 1 to 75 and `80,000` ≤ `UID` ≤ `99,999` for days 1 to 60. The validation set then uses `80,000` ≤ `UID` ≤ `99,999` for days 61 to 75, replicating the scenario of predicting the next 15 days.

For **City B**, the training set includes `0` ≤ `UID` ≤ `18,999` for days 1 to 75 and `19,000` ≤ `UID` ≤ `21,999` for days 1 to 60. The validation set then uses `19,000` ≤ `UID` ≤ `21,999` for days 61 to 75, replicating the scenario of predicting the next 15 days for these 3,000 individuals.

For **City C**, the training set includes `0` ≤ `UID` ≤ `13,999` for days 1 to 75 and `14,000` ≤ `UID` ≤ `16,999` for days 1 to 60. The validation set then uses `14,000` ≤ `UID` ≤ `16,999` for days 61 to 75, replicating the scenario of predicting the next 15 days for these 3,000 individuals.

For **City D**, having 3,000 users for validation is not feasible due to the limited number of users. Therefore, we use `0` ≤ `UID` ≤ `2,499` for training on days 1 to 60 and reserve `2,500` ≤ `UID` ≤ `2,999` for validation on days 61 to 75, replicating the scenario of predicting the next 15 days for these 500 individuals.

4.2 Model Comparisons

We compare three models. The first is a single-city (SC) approach, training separate spatiotemporal transformer (STF) models for each city to capture distinct mobility patterns, serving as our baseline. The second is STF with our proposed Stable Knowledge Transfer (SKT) from the data-rich City A. The third is SKT with our test-time trajectory refinement component (SKT+TrajRef) to handle repetitive prediction issues. For simplicity, we refer to these models as SC, SKT, and SKT+TrajRef in the remainder of this paper.

4.3 Evaluation Metrics

4.3.1 Dynamic Time Warping (DTW). DTW is a distance measure that aligns whole trajectories to minimize the total distance between the corresponding points [10]. The alignment must satisfy three criteria: the start and end points must align, the order of points must be preserved, and each step must move forward in one or both trajectories. Given two trajectories $T_A = (p_1, p_2, \ldots, p_M)$ and $T_B = (q_1, q_2, \ldots, q_N)$, DTW finds the path that minimizes the alignment cost, defined as the sum of distances between matched points:

$$\mathrm{DTW}(T_A, T_B) = \min_P \sum_{i=1}^{N} d(p_i, q_i), \tag{7}$$

where P is the alignment path, N is the number of matched pairs, and $d(p_i, q_i)$ is the Euclidean distance between the matched points. By minimizing the alignment cost, DTW measures the similarity between trajectories, producing a non-negative score where 0 indicates identical trajectories. DTW effectively captures global shape similarity by stretching or compressing segments to match the patterns. However, DTW struggles with trajectories with complex local structures or subtrajectories of different orders. This is because DTW forces a strict start-to-end alignment, resulting in high costs even when global patterns are similar.

4.3.2 GEO-BLEU. GEO-BLEU is a geospatial similarity measure adapted from the BLEU score from natural language processing [8]. It compares the local characteristics of the trajectories by representing them as sequences of n-grams, where each n-gram is a segment of n consecutive points. Let $T_A = (p_1, p_2, \ldots, p_n)$ and $T_B = (q_1, q_2, \ldots, q_n)$ represent corresponding n-grams from trajectories A and B. GEO-BLEU calculates the similarity score $S(T_A, T_B)$:

$$S(T_A, T_B) = \prod_{i=1}^{n} \exp(-\beta \cdot d(p_i, q_i)), \tag{8}$$

where $d(p_i, q_i)$ is the Euclidean distance between points p_i and q_i, and β is a scaling factor. This score ranges from 0 to 1, 1 indicating that the two n-grams are perfectly aligned. GEO-BLEU then aggregates these local similarity scores over all matched n-grams and normalizes them to produce a similarity value between 0 and 1, where a higher score indicates greater similarity. Due to its focus on local n-gram matching, GEO-BLEU is robust to minor global misalignments. Unlike DTW, this makes it ideal for scenarios where trajectories may share local motifs without requiring all predicted and reference trajectories to be globally aligned from start to end.

5 Results & Discussion

	GEO-BLEU (↑)			DTW (↓)		
	CityB	CityC	CityD	CityB	CityC	CityD
SC	0.225	0.232	0.030	39.25	30.74	279.1
SKT	0.281	0.279	0.274	33.42	23.32	34.15
SKT+TrajRef	0.288	0.285	0.281	32.79	22.96	33.80

Table 1: Experimental results on the cross-city challenge.

Table 1 shows that both SKT and SKT+TrajRef outperform the SC models across all evaluated cities, demonstrating the effectiveness of cross-city knowledge transfer. SC models, which train separate models for each city, struggle significantly in data-scarce regions such as City D. The extremely low GEO-BLEU score and high DTW value for City D highlight their inability to capture meaningful patterns due to limited training data. Even in City B and City C, which have larger datasets, SC models still underperform compared to SKT and SKT+TrajRef, indicating that our approach captures shared mobility patterns across cities more effectively than localized models. Overall, SKT achieves better generalization and more accurate predictions, especially in cities with smaller training data.

The improved performance of SKT+TrajRef compared to SKT highlights the benefit of applying trajectory refinement during test-time. By adjusting predicted trajectories, SKT+TrajRef reduces errors such as over-reliance on frequently visited locations, leading to more accurate spatial predictions. This demonstrates that combining cross-city transfer learning with test-time refinement effectively enhances prediction quality across diverse urban environments, addressing both data scarcity and varied mobility patterns.

References

[1] Xinyang Chen, Sinan Wang, Bo Fu, Mingsheng Long, and Jianmin Wang. 2019. Catastrophic forgetting meets negative transfer: Batch spectral shrinkage for safe transfer learning. *Advances in Neural Information Processing Systems* 32 (2019).

[2] Lei Dong, Carlo Ratti, and Siqi Zheng. 2019. Predicting neighborhoods' socioeconomic attributes using restaurant data. *Proceedings of the national academy of sciences* 116, 31 (2019), 15447–15452.

[3] Jie Feng, Yong Li, Chao Zhang, Funing Sun, Fanchao Meng, Ang Guo, and Depeng Jin. 2018. Deepmove: Predicting human mobility with attentional recurrent networks. In *Proceedings of the 2018 world wide web conference*. 1459–1468.

[4] Dejiang Kong and Fei Wu. 2018. HST-LSTM: A hierarchical spatial-temporal long-short term memory network for location prediction.. In *Ijcai*, Vol. 18. 2341–2347.

[5] Yan Lin, Huaiyu Wan, Shengnan Guo, and Youfang Lin. 2021. Pre-training context and time aware location embeddings from spatial-temporal trajectories for user next location prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 4241–4248.

[6] Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. 2016. Predicting the next location: A recurrent model with spatial and temporal contexts. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 30.

[7] Markus Schläpfer, Lei Dong, Kevin O'Keeffe, Paolo Santi, Michael Szell, Hadrien Salat, Samuel Anklesaria, Mohammad Vazifeh, Carlo Ratti, and Geoffrey B West. 2021. The universal visitation law of human mobility. *Nature* 593, 7860 (2021), 522–527.

[8] Toru Shimizu, Kota Tsubouchi, and Takahiro Yabe. 2022. GEO-BLEU: similarity measure for geospatial sequences. In *Proceedings of the 30th International Conference on Advances in Geographic Information Systems* (Seattle, Washington) *(SIGSPATIAL '22)*. Association for Computing Machinery, New York, NY, USA, Article 17, 4 pages. https://doi.org/10.1145/3557915.3560951

[9] Haru Terashima, Naoki Tamura, Kazuyuki Shoji, Shin Katayama, Kenta Urano, Takuro Yonezawa, and Nobuo Kawaguchi. 2023. Human Mobility Prediction Challenge: Next Location Prediction using Spatiotemporal BERT. In *Proceedings of the 1st International Workshop on the Human Mobility Prediction Challenge*. 1–6.

[10] Kevin Toohey and Matt Duckham. 2015. Trajectory similarity measures. *Sigspatial Special* 7, 1 (2015), 43–50.

[11] Leye Wang, Xu Geng, Xiaojuan Ma, Feng Liu, and Qiang Yang. 2019. Cross-city transfer learning for deep spatio-temporal prediction. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*. 1893–1899.

[12] Yu Wang, Tongya Zheng, Yuxuan Liang, Shunyu Liu, and Mingli Song. 2024. Cola: Cross-city mobility transformer for human trajectory simulation. In *Proceedings of the ACM on Web Conference 2024*. 3509–3520.

[13] Ying Wei, Yu Zheng, and Qiang Yang. 2016. Transfer knowledge between cities. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1905–1914.

[14] Qiong Wu, Kaiwen He, Xu Chen, Shuai Yu, and Junshan Zhang. 2021. Deep transfer learning across cities for mobile traffic prediction. *IEEE/ACM Transactions on Networking* 30, 3 (2021), 1255–1267.

[15] Takahiro Yabe, Kota Tsubouchi, Toru Shimizu, Yoshihide Sekimoto, Kaoru Sezaki, Esteban Moro, and Alex Pentland. 2024. YJMob100K: City-scale and longitudinal dataset of anonymized human mobility trajectories. *Scientific Data* 11, 1 (2024), 397.