Supplementary Information for Learning the complexity of urban mobility with deep generative collaboration network

Yuan Yuan[1+], Jingtao Ding[1+*], Depeng Jin[1], and Yong Li[1*]

[1]Beijing National Research Center for Information Science and Technology (BNRist), Department of Electronic Engineering, Tsinghua University, Beijing, P. R. China
[*]To whom correspondence should be addressed; Emails: liyong07@tsinghua.edu.cn, dingjt15@tsinghua.org.cn.
[+]Yuan Yuan and Jingtao Ding contribute equally to this work.

S1. Model detail

In this section, we elaborate on the detailed design of the proposed framework. The notations used in this paper are summarized in Supplementary Table 1.

S1.1 Data processing and feature extraction

Multilevel urban mobility data includes individual trajectories and population flows. An *individual trajectory* can be represented by a spatial-temporal movement sequence $X = [x_1, x_2, ..., x_T]$. The maximum sequence length T is decided by both the overall duration and time interval of records. The t_{th} element x_t is a movement record defined as a tuple $(t_0 + t \times \Delta, l_t)$, where t_0 denotes the starting timestamp of X, Δ denotes the record time interval and $l_t \in L$ denotes the location of interest. We also define a set of non-overlapping regions of interest $r \in G_r$, which are geographical polygons based on administration partition. The *population flow* $F(t_1, t_2, r, r')$ is the total number of people moving from region r to region r' during the period $t_1 \sim t_2$, which is obtained by aggregating massive individual trajectories during this specified period.

Next, we extract features from raw data to represent locations and regions. To characterize each location l_i, we consider two options. The first option is to directly use the one-hot location ID for representing locations and train corresponding embeddings in an end-to-end way. The second option uses important features related to its location attractiveness and urban functions, respectively. The first feature is visitation popularity which counts the aggregated number of user visits. We extract this feature by aggregating visit counts of all Point-of-Interests (POIs) located within l_i, which can be obtained from third-party data companies like SafeGraph (https://www.safegraph.com/). The second feature is a POI distribution which counts the POI number grouped by categories like the resident, business, transport, etc. The relative proportion between different POI categories reflects the urban function of a location. It can also be easily accessed from open geographic databases like OpenStreetMap (https://www.openstreetmap.org/). As for representing a region r, similarly, we use the features of population size and POI distribution, of which the former can be obtained from the National Population Census. To reduce the noise of continuous variables and minimize the abnormal effects, we convert the above three continuous features into discrete characteristics by mapping each value into one of ten categories based on the total percentage.

S1.2 Preliminary on generative adversarial imitation learning

We consider a Markov decision process (MDP) $< S, A, P, R, \gamma >$. S is the state space, A is the action space, $P : S \times A \mapsto [0, 1]$ denotes the state transition, $R : S \times A \mapsto \mathbb{R}$ denotes the reward function, which evaluates the utility of taking action under the state, and $\gamma \in [0, 1]$ is a discount factor, which determines to what extent future rewards are considered. An agent takes actions following the policy $\pi : S \rightarrow A$. Specifically, at the t_{th} step with state $s_t \in S$, an agent takes the action $a_t \in A$ according to the action probability $\pi(a_t|s_t)$. After taking action, it receives the reward $r_t \in R$ based on s_t and a_t. We use an expectation with respect to a policy π to denote the expectation with respect to the trajectory generated by π, i.e., $\mathbb{E}_\pi[r] = \mathbb{E}[\sum_{t=1}^{\infty} \gamma^t r(s_t, a_t)]$, where $s_0 \sim p_0$, $a_t \sim \pi(\cdot|s_t)$, and $s_{t+1} \sim P(\cdot|s_t, a_t)$. We also use $\hat{\mathbb{E}}_\tau$ to denote empirical expectation with respect to trajectory samples τ. To accommodate the multi-agent setting, we use bold variables without subscript n to denote the concatenation of all variables for all agents. For example, Π denotes the joint policy, $\mathbf{r}$ denotes global rewards and $\mathbf{a}$ denotes the actions of all agents.

Given a sequence of state-action pairs E defining expert behavior π_E, i.e., $E = \{(s,a)|\forall s, a \sim \pi_E(\cdot|s)\}$, the goal of generative adversarial imitation learning (GAIL) is directly learning optimal policy function π from this demonstration data. Specifically, GAIL fits data distributions in E by generative adversarial training, that is, alternating between optimizing a discriminative classifier D and a generative model G. Note that G generates state-action pairs according to learned policy π. Mathematically, the above optimization process can be formulated as finding a saddle point (π, D) as follows[13]:

$$\min_{\pi} \max_{D} \mathbb{E}_{\pi}\left[\log(D(s,a))\right] + \mathbb{E}_{\pi_E}\left[\log(1-D(s,a))\right] - \lambda H(\pi) \ , \tag{1}$$

where $H(\pi) = \mathbb{E}_{\pi}[-\log \pi(a|s)]$ is the γ-discounted causal entropy of the policy π[5]. The objective of D is to maximize its likelihood of distinguishing state-action pairs (s,a) generated by G from the expert demonstration in E, while G tries to minimize this likelihood, i.e., leaning the optimal policy π that imitates the expert policy π_E. In the terminology of reinforcement learning, the former refers to recovering the expert's reward function and the latter refers to running reinforcement learning on this learned reward function to search for the optimal policy.

To solve this problem, GAIL uses two function approximation for D and π, i.e., a ω-parameterized discriminator network $D_\omega : S \times A \to (0,1)$ and a θ-parameterized policy network π_θ. The optimization of Equation (1) with respect to the discriminator D_ω is

$$\max_{\omega} \mathbb{E}_{\pi}\left[\log(D_\omega(s,a))\right] + \mathbb{E}_{\pi_E}\left[\log(1-D_\omega(s,a))\right] \ , \tag{2}$$

which adopts widely used gradient descent methods like Adam. As for its counterpart, minimizing Equation (1) with respect to π_θ can be seen as searching the optimal policy $\pi_\theta(s,a)$ with a reward function $\log(D_\omega(s,a))$, which requires using policy gradient methods like Proximal Policy Optimization (PPO)[23]. In general, PPO finds the optimal π_θ by constantly updating it towards directions suggested by an advantage function $A_\pi(s,a)$[12] that replaces $\log(D_\omega(s,a))$ and measures the advantage of taking a specific action a in state s, over randomly selecting an action according to $\pi_\theta(\cdot|s)$. Specifically, $A_\pi(s,a)$ equals to difference between the state-action value function $Q_\pi(s,a)$ and the state value function $V_\pi(s)$, i.e., $A_\pi(s,a) = Q_\pi(s,a) - V_\pi(s)$. The optimization of Equation (1) with respect to the policy network π_θ is

$$\max_{\theta} \mathbb{E}_{\pi_{\text{old}}}\Big[\min\big(\text{Ra}(\theta)A_\pi(s,a), \text{Clip}(\text{Ra}(\theta), 1-\varepsilon, 1+\varepsilon)A_\pi(s,a)\big)\Big] \ , \tag{3}$$

where $\text{Ra} = \pi_\theta(a|s)/\pi_{\theta_{\text{old}}}(a|s)$ is the probability ratio of taking action a at state s between the updated policy π_θ and the old policy $\pi_{\theta_{\text{old}}}$. The second term, $\text{Clip}(\text{Ra}(\theta), 1-\varepsilon, 1+\varepsilon)A_\pi(s,a)$, clips the probability ratio to keep it inside of the interval $[1-\varepsilon, 1+\varepsilon]$. The overall objective is the minimum of the clipped and unclipped objective, with a penalty for having too large of a policy update. As for the calculation of the advantage function $A_\pi(s,a)$, it can be estimated by one-step temporal difference error using a parameterized critic network V_ϕ, i.e.,

$$A_\pi(s,a) \approx A_\phi(s,a) = r(s,a) + \gamma V_\phi(s') - V_\phi(s) \ . \tag{4}$$

Note that the first two terms approximate $Q_\pi(s,a)$ given the next-step state $s' \sim P(\cdot|s,a)$. The critic network V_ϕ is optimized using Monte-Carlo policy evaluation that minimizes the following mean squared error between the empirical mean return $\hat{V}_\pi(s)$ and the function approximation $V_\phi(s)$,

$$\begin{aligned} &\min_{\phi} \mathbb{E}_{\pi}\left[(\hat{V}_\pi(s) - V_\phi(s))^2\right] \ , \\ &\text{where } \hat{V}_\pi(s) = \hat{\mathbb{E}}_\tau\Big[\sum_{k=1}^{T-t} \gamma^{k-1} r_{t+k} \mid s_t = s\Big] \ , \\ &\text{and } r_{t+k} = r(s_{t+k}, a_{t+k}) = \log(D_\omega(s_{t+k}, a_{t+k})), \ a_{t+k} \sim \pi_\theta(\cdot|s_{t+k}) \ . \end{aligned} \tag{5}$$

S1.3 PPO-based generator update

The current mobility policy $\pi_\theta(a_t|o_t)$ of the generator are evaluated by two critics, i.e., $V_{\phi_i}(o_t)$ and $Q_{\phi_p}(o_t, a_t)$, with respect to its ability to capture mobility regularity and its further impact on population mobility, respectively. The advantage function for model optimization using PPO in Equation (3) is as follows:

$$A_\phi(o_t, a_t) = (1-\eta)(r(o_t, a_t) + \gamma V_{\phi_i}(o_{t+1})) + \eta Q_{\phi_p}(o_t, a_t) - V_{\phi_i}(o_t), \tag{6}$$

where η is the trade-off weight controlling the collaboration of the two critics.

S1.4 Model training and generation

We train our model for hundreds of iterations to learn the skills of multilevel urban mobility generation. The overall framework is trained in an adversarial manner, where we train the policy function modeled by the generator network and the reward function modeled by the discriminator network alternatively. In each iteration, we collect training samples of a few hundred episodes by generating individual trajectories. We first train the discriminator by a binary classification to distinguish between real state-action pairs and generated state-action pairs. The loss function is defined as $L_{dis} = -\frac{1}{n}\sum_{i=1}^{N} y_i \log p_i + (1-y_i)\log(1-p_i)$. Then we update the parameters of the framework using Proximal Policy Optimization (PPO)[23]. Algorithm 1 illustrates the training process of the overall framework.

With a well-trained model, we perform model inference to generate urban mobility. We first generate movement trajectories of individuals and then obtain mobility flows by aggregating individuals' movements. The generation algorithm is summarized in Algorithm 2.

S1.5 Location representation design for model transferability

To adapt to transferring between different cities, we remove the ID embedding approach used in previous experiments. Instead, we use a number of input attributes to represent each location, and the input vector of the location is $x_i = \texttt{Concat}[x_i^1, x_i^2]$, which is a concatenation of these features: x_i^1 and x_i^2 are the popularity percentile (characterized by visitation frequency) and POI distribution of the location l_i, respectively. Then the input vectors x_i are fed into the same feed-forward neural network, which has 10 hidden layers of dimension 64 with the LeakyReLu function. The output dimension of the feed-forward neural network is 32, which acts as the location embedding. The spatial representation is obtained from location features, which contain attributes describing the properties of the location. Compared with ID embeddings, the representation method becomes transferable across different cities. Supplementary Fig. 4 illustrates the designed location representation approach for model transferability.

S2. Experiment details

S2.1 Experimental settings

Our selected hyper-parameters covering basic network architecture and training process are listed in Supplementary Table 2. For each city, we preprocess the corresponding mobility records and urban context information to construct a training dataset of 20,000 instances. Generally, we train DeepMobility for 2000 iterations, and we train the generator for 5 epochs in every iteration. We monitor the training process by generating 50,000 individual trajectories and the resulting flows every 10 iterations and evaluate their data quality. The JS divergence and the CPC improve (decreases and increases, respectively) with training, meaning that DeepMobility learns the complex data distribution behind urban mobility. The training stabilizes after 600 iterations with the discriminator accuracy around 50%, suggesting that the DeepMobility is not subject to overfitting. For each validation, we choose the generator with the highest CPC and use it to simulate a population containing the same number of individuals as in our collected data, and their corresponding one-week movements throughout the urban region. Note that to reproduce long-term human mobility patterns, such as ultra-slow diffusion of gyration radius (r_g) shown in Supplementary Fig. 10, we also generate one-year movements of 200 individuals with heterogeneous r_g.

S2.2 Baseline approaches

Continuous-time Random Walk[6]. Continuous-time Random Walk (CTRW) is a generalization of a random walk, where the waiting time between two jumps is a stochastic variable. Brockmann et al. propose to use a two-parameter continuous-time random walk model to describe human travel for the first time. They have found that the waiting time δt and travel distances r both universally follow the pow-law distribution:

$$P(r) \sim r^{-(1+\beta)}\ , \tag{7}$$

$$P(\Delta t) \sim \Delta t^{-(1+\alpha)}\ , \tag{8}$$

where $\beta = 0.59 \pm 0.02, \alpha = 0.60 \pm 0.03$. When an agent jumps from the current location, we sample the next location with the probability proportional to $P(r)$.

Exploration and Preferential Return (EPR)[26]. EPR model[26] is proposed to explain the emergence of observed scaling laws[6]. This model has the following mechanisms:

- The waiting time Δt follows the distribution $P(\Delta t) \sim \Delta t^{-(1+\beta)}\exp(-\Delta t/\tau)$.
- For location selection, the probability of visiting a new location is $P_{new} = \rho S^{-\gamma}$, where S is the total number of distinct locations that have been visited. Correspondingly, the probability of returning to previously visited locations is $1 - \rho S^{-\gamma}$, which is the return phase.

- If an agent decides to explore a new location, the probability of different locations is associated with the geographic distance from the current location r_{ij}, i.e., $p(i,j) \sim \frac{1}{r_{ij}^2}$. Then, the number of distinct locations S is increased by one.
- If an agent decides to return to a previously visited location, the probability is proportional to the historical visitation frequency f_i.

Recency-EPR (r-EPR)[4]. r-EPR model is another extension of EPR model, which incorporates the concept of recency. Specifically, this model also considers recently-visited locations not only frequently-visited locations. The r-EPR model has the following mechanisms:

- The waiting time Δt follows the distribution $P(\Delta t) \sim \Delta t^{-(1+\beta)} \exp(-\Delta t/\tau)$.
- For location selection, the probability of visiting a new location is $P_{new} = \rho S^{-\gamma}$, which is the explore phase. Correspondingly, the probability of returning to previously visited locations is $1-\rho S^{-\gamma}$, which is the return phase.
- If an agent decides to explore a new location, the probability of different locations is associated with the geographic distance from the current location r_{ij}, i.e., $p(i,j) \sim \frac{1}{r_{ij}}^2$. Then, the number of distinct locations S is increased by one.
- If an agent decides to return to a previously visited location, it is with the probability $1-\alpha$ to visit the i_{th} last visited location from a Zipf's law: $p(i) \sim K_s(l_i)^{-\eta}$, where $K_s(l_i)$ denotes the recency-based rank. At the same time, it is also with the probability α to visit the location following the distribution of visitation frequencies: $p(i) \sim K_f(l_i)^{-\eta}$, where $K_f(l_i)$ denotes the frequency-based rank.

Density-EPR (d-EPR)[19,18]. d-EPR model is another extension of EPR model, which takes the population density into consideration. The d-EPR model has the following mechanisms:

- The waiting time Δt follows the distribution $P(\Delta t) \sim \Delta t^{-(1+\beta)} \exp(-\Delta t/\tau)$.
- For location selection, the probability of visiting a new location is $P_{new} = \rho S^{-\gamma}$, which is the explore phase. Correspondingly, the probability of returning to previously visited locations is $1-\rho S^{-\gamma}$, which is the return phase.
- If an agent decides to explore a new location, the probability of different locations follows the gravity model with probability $p(i,j) \sim \frac{1}{N} \frac{n_i n_j}{r_{ij}^2}$, where n_i is the population size, r_{ij} is the distance between location i and location j.
- If an agent decides to return to a previously visited location, the probability is proportional to the historical visitation frequency f_i.

Memory-Preferential Random Walk (MPRW)[30]. This model claims that the visitation frequencies to different locations follow Zipf's law. When considering the memory effect, it considers the probability of visiting a location is inversely proportional to the rank, which is formulated as follows:

$$A_i = 1 + \frac{\lambda}{r_i} \,, \tag{9}$$

where λ characterizes the memory effect and the r_i denotes that target location i is the r_{th} newly visited location in the movement trajectory. If the target location i has never been visited, the value of $p(i)$ keeps unity. In addition, this model assumes that the location attractiveness is proportional to the population size, and the attractiveness from location i to location j is modeled as $\frac{m_i}{W_{ji}}$, where W_{ji} counts the total population belonging to the circular region, whose center is the location j. The radius of the circular region is the distance between locations i and j. As a result, The transition probability of traveling from location i to j can be written as:

$$p(i,j) \sim \frac{m_i}{W_{ji}} (1 + \frac{\lambda}{r_i}) \,. \tag{10}$$

TimeGeo[14]. It is also an extension of EPR model, which redesigns the temporal choice mechanism to capture the circadian propensity to travel. It has the following mechanisms:

- It models the global movement circadian rhythm at the population level, i.e.,$P(t)$, which is measured as the ratio of movements in the time interval t.
- It first examine whether the individual is at home and then decides whether to move. For individuals at home, the probability to move is $n_w P(t)$, while for individuals at home, the probability to move is $\beta_1 n_w P(t)$.
- Each time the individual chooses to move, it will decide whether to go home. If not going home, it adopts a rank-based exploration and preferential return model.

Long short-term memory (LSTM)[11]. It is a typical type of recurrent neural networks that is capable of learning order dependence in sequences effectively. Specifically, it applies a multi-layer long short-term memory recurrent neural network to encode an input sequence. We train this model by maximizing likelihood estimation on observed individual trajectories, and then perform trajectory generation by making predictions regressively.
Sequence Generative Adversarial Nets (GAN)[31]. GAN models a data generator as a stochastic parametrized policy used in reinforcement learning (RL). Besides, the RL reward is obtained from the Generative Adversarial Nets (GAN)[7] discriminator by judging the complete sequence. In its policy gradient, the model employs Monte Carlo (MC) search to approximate the state-action value by passing back reward signals to intermediate state-action steps. Specifically, the optimization of the generator in GAN is to minimize the expected reward:

$$J(\theta) = \mathbb{E}[R_T|s_0,\theta] = \sum_{y_1\in\mathscr{Y}} G_\theta(y_1|s_0)\dot{Q}_{D_\phi}^{G_\theta}(s_0,y_1)\ . \tag{11}$$

The discriminator is trained as follows:

$$\min_{\phi} -\mathbb{E}_{Y\sim p_{data}}[\log D_\phi Y] - \mathbb{E}_{Y\sim G_\theta}[\log(1-D_\phi(Y))]\ . \tag{12}$$

S2.3 Evaluation metrics.

To measure generation realism, we compare the urban trajectories generated by DeepMobility with the empirical data at both individual and population levels. Individual human movements can be characterized by the following fundamental metrics[3,28,17]: (I1) Jump length, the distance between two consecutive locations visited by an individual, (I2) Weekly trip distance, the accumulative distance traveled by an individual during a week, (I3) Radius of gyration, the characteristic distance traveled by an individual away from the starting point, and (I4) Stay duration, the time spent at a location, (I5) Daily Visited Locations, the number of locations visited by an individual during a day. The first three metrics depict spatial patterns of human mobility, while the rest two capture mobility features from a temporal perspective. To quantify the differences between empirical distributions of data and the model simulation, we use the Jensen-Shannon divergence (JSD) and the Kolmogorov–Smirnov (KS) test[14,9], which are defined as follows:

$$\mathrm{JSD}(p||q) = \frac{1}{2}D_{\mathrm{KL}}(p\|\frac{p+q}{2}) + \frac{1}{2}D_{\mathrm{KL}}(q\|\frac{p+q}{2})\ ,\ D_{\mathrm{KL}} = \int_{-\infty}^{\infty} p(x)\log(\frac{p(x)}{q(x)})dx\ , \tag{13}$$

$$\mathrm{KS}(p||q) = \sup_x |P(x)-Q(x)|\ , \tag{14}$$

where H is the Shannon entropy, P and Q are cumulative distributions of p and q, and $\sup_x$ denotes the maximum distance between $P(x)$ and $Q(x)$. JSD and KS are always positive and are bound by $[0,1]$, with 0 indicating a perfect match between two distributions and 1 suggesting no overlap between two distributions. At the population level, the mobility flows between regions are calculated as how many people move from one region to another within a period of time. Therefore we use (P1) the Common Part of Commuters (CPC), which is a well-established metric to quantify the similarity between generated flows y_g and real flows y_r[25,21], and (P2) the Mean Absolute Error (MAE) to calculate the distance between the real and the generated flows[16,17]. The two metrics are defined as follows:

$$\mathrm{CPC} = \frac{2\sum_{i,j,k}\min(y_g(\tau_k,r_i,r_j),y_r(\tau_k,r_i,r_j))}{\sum_{i,j,k}y_g(\tau_k,r_i,r_j)+y_r(\tau_k,r_i,r_j)}\ , \tag{15}$$

$$\mathrm{MAE} = \frac{1}{N\times N\times K}\sum_{k=1}^{K}\sum_{i=1}^{N}\sum_{j=1}^{N}\left|y_g(\tau_k,r_i,r_j)-y_r(\tau_k,r_i,r_j))\right|\ , \tag{16}$$

where $y(\tau_k,r_i,r_j)$ denotes the population flow from region r_i to region r_j during the period τ_k. Each day is divided into 4 periods: $[0:00\sim 6:00)$, $[6:00\sim 12:00)$, $[12:00\sim 18:00)$, and $[18:00\sim 24:00)$. CPC is always positive and bounded by $[0,1]$ with 1 indicating the two are identical and 0 suggesting no overlap.

S3. Supplementary results

S3.1 Complete performance comparison with baseline approaches

We implement the above baseline approaches and evaluate their performance regarding the individual level and population level. For EPR-based models, including EPR, r-EPR, d-EPR, and MPRW, mobility trajectories are with stochastic time intervals between consecutive jumps. We fill in the corresponding position for time intervals with the last location to align it with other mobility trajectories, as waiting time denotes the time interval that the individual stays at the location.

Supplementary Table 3 and Table 4 show the generation performance on multilevel objectives where our method outperforms all the baseline methods with significant improvements. TimeGeo achieves the second-best performance in many scenarios, implying the necessity to incorporate temporal choice. In addition, GAN which leverages LSTM as the generator delivers better performance than the solely LSTM model, which indicates that the long-term rewards should be considered rather than only considering step-wise optimization. Supplementary Fig. 5 and Fig. 6 visualize the generated population flows between regions of real cases, generated by our model and baselines, respectively. Specifically, we select the two baselines TimeGeo and GAN, without the loss of generality as they are representative of physical models and deep generative models. We can observe that population flows generated by our model exhibit the most similar spatial distribution with observed flows, which successfully avoids both overestimation and underestimation. We can find that our model is capable of capturing the scaling law of jump length, where the number of movements decreases with the increasing distance. However, TimeGeo overestimates the low flows and underestimates the high flows, failing to capture the dense flows at the city center. Besides, GAN obviously overestimates population flows.

S3.2 More experimental results on the daily life patterns reproduced in generated trajectories

Supplementary Fig. 7 demonstrates the home distribution between real data and generated data. Specifically, each individual's home is identified as the location with the highest ratio at night time[14]. Without the loss of generality, we select Beijing and Senegal, where Beijing is representative of Chinese cities. As we can observe, the home distribution extracted from generated urban mobility is remarkably similar to the real case. Take Beijing as an example, the famous residential areas, including the Huilongguan district and the Tiantongyuan neighborhood, are successfully identified by our model.

When considering typical patterns of human mobility, one way is to construct mobility-pattern-based networks, i.e., mobility motifs, for each individual[3]. Mobility motifs are referred to as some simple movement patterns that repeatedly occur in the individual trajectory. A motif is defined as a successive sequence of locations with the same initial and last locations and represents the regularities underlying daily movements[22]. From the four empirical datasets, nine typical distinct motifs are identified. It was found that more than 95% of the recorded movements made by all individuals can be described with these 9 motifs. We count the motif distribution in Supplementary Fig. 8, where the frequency is shown in the log scale to show the infrequent motifs clearer. As we can observe, for all four datasets, our framework exhibits the same mobility motifs and has nearly indistinguishable frequencies with empirical results. The motifs with node sizes 2 and 3 are reproduced with higher propensities successfully. These results indicate that the underlying regularities existing in our daily mobility can be successfully captured. Due to the important role of motifs in determining microscopic mobility characteristics, the agreement between observed motifs and generated motifs further validates our model's effectiveness at the individual level.

S3.3 More experimental results on mobility laws

Sublinear growth in the number of unique locations visited

Supplementary Fig. 10a-d shows the number of distinct locations $S(t)$ visited by an individual versus time t (up to one year) in a log–log plot. We observe consistent power-law growths $S(t) \sim t^{\mu}, \mu < 1$, where larger r_g has slightly more visited locations. Different from exponential growths, the power-law increasing patterns with $\mu < 1$ can be attributed to the memory effect that individuals have a decreasing tendency to visit new locations[26].

Ultraslow diffusion process

Moreover, we find that individual trajectories (up to one year) generated by DeepMobility not only reproduce the fat-tailed distribution of $P(r_g)$, but also capture the ultraslow growth pattern of r_g in Supplementary Fig. 10e-h, in line with findings revealed by previous studies[10,26].

Limits of predictability in human mobility

We examine the degree of predictability of our generated mobility. Following previous work[27], we calculate three entropy measures: (i) The random entropy S_i^{rand}, (ii) the temporal-uncorrelated entropy S_i^{unc}, and (iii) the actual entropy S_i. Supplementary Fig. 11 a-d illustrates the distributions of three entropy measures, $P(S_i^{rand})$, $P(S_i^{unc})$, and $P(S_i)$, which shows the similar distributions as the real-world human mobility data. We find that there exists a significant shift of $P(S_i)$ compared with $P(S_i^{rand})$. The observation indicates that the individuals' real mobility uncertainty is much lower than the uncertainty only focusing on the number of visited locations, which is in line with previous findings[27].

Power law distribution of the region-to-region trip number

Prior work on population-level urban mobility has observed that the number of trips, i.e., aggregated flows, between two regions follows a power law distribution[30]. Our generated mobility data at the population level again reproduces this pattern, as illustrated in Supplementary Fig. 13.

Statistical properties of origin-destination demand networks across different cities

Prior work on urban mobility networks has discovered very similar statistical patterns across different cities, in terms of node degree, node-level flow (total flow of one node including inflow and outflow) and origin-destination flow[20]. Our generated mobility networks, with locations represented as nodes and mobility flows represented as weighted directed edges, across four different cities also exhibit similar power-law-like scaling patterns that remain stable (Supplementary Fig. 14).

S3.4 Detailed experimental results of geographical transferability

We study model transferability by training it with a dataset of one city and testing it on another city, whose mobility data is not used in the training process. This kind of transferability enables us to train a model in a city with urban mobility data available and apply it to generate complete urban mobility for less-developed cities with limited urban mobility. An embedding layer transforms the location attributes and region attributes into embeddings, and then the state encoder and trajectory decoder in our model operate on location and region embeddings. The number of model parameters is independent of both the number of locations and the number of regions. Therefore, our model is naturally suited for transfer between different cities, since we can directly utilize a pretrained model in new cities without introducing new model parameters.

Supplementary Table 5 demonstrates the complete results of model transferability. We compare the performance of our model with physical models, the parameters of which are also transferable across different cities. As we can observe, our model also achieves the best performance in the transfer setting.

S3.5 Privacy preservation evaluation

To prove that the urban mobility generated by our framework does not leak individual privacy, we perform experiments from three aspects.

- **Uniqueness testing.**[8] This measure evaluates whether the generated mobility trajectories are completely identical to the original trajectories. It highlights the extent to which the model directly generates copies instead of brand-new data.
- **Membership inference attack.**[15,24] If the generated data does not reveal the identities of users from the original data, it should not be possible to use the generated data to reidentify users in the training set. For this purpose, we use the framework of membership inference attack[15,24]. Stronger privacy protection leads to a lower attack success rate.
- **Differential privacy.**[1,15] A model is differentially private if for any pair of training datasets and that differ in the record of a single user, it holds that: $M(z;D) \leq e^{\varepsilon} M(z;D') + \delta$, which means one can hardly distinguish whether any individual is included in the original dataset or not by looking at the output. It is a rigorous mathematical definition of privacy.

Uniqueness testing.

To prove that the realistic generated mobility trajectory is not a simple copy of the real trajectory but a brand-new trajectory, we perform a uniqueness testing of it by comparing it with the real data. We randomly select generated trajectories and compare them with all the real trajectories from the training set. The two trajectories are aligned in the time dimension one by one and determine whether the locations at the corresponding time points are exactly the same. The overlapping ratio is defined as the ratio of the number of identical locations to the total trajectory length. Next, we choose the real trajectory that is most similar to generate one, that is, the one with the highest overlapping ratio. We calculate the overlapping ratio distribution of all the generated trajectories with the most similar real trajectories mentioned before. The results can also be extended by considering more similar trajectories, e.g., the top3 and top5 similar real trajectories. As shown in Supplementary Fig. 15 **a-d**, for both the Beijing, Shenzhen, and Senegal datasets, more than 80% of the generated mobility trajectories cannot find any real trajectories that have more than 40% overlapping ratio with them. For the Shanghai dataset, more than 80% of the generated mobility trajectories overlap with real trajectories with an overlapping ratio of less than 20%. The results demonstrate that, while capturing mobility patterns, our framework indeed learns to generate brand-new and unique trajectories rather than simply copying.

Membership inference attack.

Given a deep learning model and an individual record, the goal of the attack is to determine whether this record was included in the training set or not. We follow the attack settings as[24,15], where the attacker's access to the deep learning model is that they can obtain the model's output. To improve the attack performance, we estimate the individual information leakage using powerful machine learning models, which are trained to predict whether an individual is in the training set. To control the impact of classification methods, four common-used classification algorithms are included: Logistic Regression(LR), Support Vector Machine(SVM), K-nearest neighbors algorithm(KNN) and Random Forest(RF). The positive samples are those individuals in the training data, while the negative samples are not. The input feature is the overlapping ratios of multiple runs. The metric is the success rate, which is the percentage of successful trials in guessing whether a sample is in the training set[24,15]. Stronger privacy protection leads to a lower success rate. Supplementary Fig. 15 **f** shows the attack results. On the four empirical datasets, including Beijing, Shanghai, Shenzhen, and Senegal, the attack success rate is less than 0.52, which means that attackers can hardly infer whether individuals are in the training set only with the information of the generated urban mobility. The results suggest the robustness of our framework to membership inference attacks.

Differential privacy.

A model M is (ε, δ) differentially private if for any pair of datasets D and D' that differ in the activity data of a single user, it holds that:

$$M(z;D) \leq e^{\varepsilon} M(z;D') + \delta$$

For the output z, $M(z,D)$ denotes the probability distribution of z with the data D as the input. Smaller values of ε and δ give more privacy. In our experiment, we examine the privacy budget of our proposed model from the perspective of changes in the overlapping ratio. Specifically, the overlapping ratio of each individual under the conditions that this individual is contained in the training set or not is modeled by two Gaussian distributions respectively, which are then regarded as $M(z,D)$ and $M(z,D')$ to calculate the privacy budget ε. For each user, we calculate the ε using TensorFlow Privacy[1,2]. The cumulative distribution of ε is illustrated in Supplementary Fig. 15 **e**. We can observe that without any additional privacy-preserving mechanism, our model is able to have a max privacy budget $\varepsilon < 0.03$, which is typically considered as a reasonable operating point for generative models[15], e.g., Apple uses a privacy budget of 4.0[1]. The privacy budget can be further enhanced by introducing DP-SGD[1] or DP-GAN[15,29].

References

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. "Deep learning with differential privacy". In: *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*. 2016, pp. 308–318.

[2] Galen Andrew, Steve Chien, and Nicolas Papernot. *TensorFlow privacy*. 2019.

[3] Hugo Barbosa, Marc Barthelemy, Gourab Ghoshal, Charlotte R James, Maxime Lenormand, Thomas Louail, Ronaldo Menezes, José J Ramasco, Filippo Simini, and Marcello Tomasini. "Human mobility: Models and applications". In: *Physics Reports* 734 (2018), pp. 1–74.

[4] Hugo Barbosa, Fernando B de Lima-Neto, Alexandre Evsukoff, and Ronaldo Menezes. "The effect of recency to human mobility". In: *EPJ Data Science* 4 (2015), pp. 1–14.

[5] Michael Bloem and Nicholas Bambos. "Infinite time horizon maximum causal entropy inverse reinforcement learning". In: *53rd IEEE conference on decision and control*. IEEE. 2014, pp. 4911–4916.

[6] Dirk Brockmann, Lars Hufnagel, and Theo Geisel. "The scaling laws of human travel". In: *Nature* 439.7075 (2006), pp. 462–465.

[7] Antonia Creswell, Tom White, Vincent Dumoulin, Kai Arulkumaran, Biswa Sengupta, and Anil A Bharath. "Generative adversarial networks: An overview". In: *IEEE signal processing magazine* 35.1 (2018), pp. 53–65.

[8] Yves-Alexandre De Montjoye, César A Hidalgo, Michel Verleysen, and Vincent D Blondel. "Unique in the crowd: The privacy bounds of human mobility". In: *Scientific reports* 3.1 (2013), pp. 1–5.

[9] Jie Feng, Zeyu Yang, Fengli Xu, Haisu Yu, Mudan Wang, and Yong Li. "Learning to simulate human mobility". In: *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*. 2020, pp. 3426–3433.

[10] Marta C Gonzalez, Cesar A Hidalgo, and Albert-Laszlo Barabasi. "Understanding individual human mobility patterns". In: *nature* 453.7196 (2008), pp. 779–782.

[11] Alex Graves and Alex Graves. "Long short-term memory". In: *Supervised sequence labelling with recurrent neural networks* (2012), pp. 37–45.

[12] Mance E Harmon and Stephanie S Harmon. "Reinforcement Learning: A Tutorial." In: (1997).

[1] https://www.apple.com/privacy/docs/Differential_Privacy_Overview.pdf

[13] Jonathan Ho and Stefano Ermon. "Generative adversarial imitation learning". In: *Advances in neural information processing systems* 29 (2016).

[14] Shan Jiang, Yingxiang Yang, Siddharth Gupta, Daniele Veneziano, Shounak Athavale, and Marta C González. "The TimeGeo modeling framework for urban mobility without travel surveys". In: *Proceedings of the National Academy of Sciences* 113.37 (2016), E5370–E5378.

[15] Zinan Lin, Alankar Jain, Chen Wang, Giulia Fanti, and Vyas Sekar. "Using GANs for sharing networked time series data: Challenges, initial promise, and open questions". In: *Proceedings of the ACM Internet Measurement Conference*. 2020, pp. 464–483.

[16] Zhicheng Liu, Fabio Miranda, Weiting Xiong, Junyan Yang, Qiao Wang, and Claudio Silva. "Learning geo-contextual embeddings for commuting flow prediction". In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 34. 01. 2020, pp. 808–816.

[17] Massimiliano Luca, Gianni Barlacchi, Bruno Lepri, and Luca Pappalardo. "A survey on deep learning for human mobility". In: *ACM Computing Surveys (CSUR)* 55.1 (2021), pp. 1–44.

[18] Luca Pappalardo, Salvatore Rinzivillo, and Filippo Simini. "Human mobility modelling: exploration and preferential return meet the gravity model". In: *Procedia Computer Science* 83 (2016), pp. 934–939.

[19] Luca Pappalardo, Filippo Simini, Salvatore Rinzivillo, Dino Pedreschi, Fosca Giannotti, and Albert-László Barabási. "Returners and explorers dichotomy in human mobility". In: *Nature communications* 6.1 (2015), p. 8166.

[20] Meead Saberi, Hani S Mahmassani, Dirk Brockmann, and Amir Hosseini. "A complex network perspective for characterizing urban travel demand patterns: graph theoretical analysis of large-scale origin-destination demand networks". In: *Transportation* 44 (2017), pp. 1383–1402.

[21] Markus Schläpfer, Lei Dong, Kevin O'Keeffe, Paolo Santi, Michael Szell, Hadrien Salat, Samuel Anklesaria, Mohammad Vazifeh, Carlo Ratti, and Geoffrey B West. "The universal visitation law of human mobility". In: *Nature* 593.7860 (2021), pp. 522–527.

[22] Christian M Schneider, Vitaly Belik, Thomas Couronné, Zbigniew Smoreda, and Marta C González. "Unravelling daily human mobility motifs". In: *Journal of The Royal Society Interface* 10.84 (2013), p. 20130246.

[23] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. "Proximal policy optimization algorithms". In: *arXiv preprint arXiv:1707.06347* (2017).

[24] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. "Membership inference attacks against machine learning models". In: *2017 IEEE symposium on security and privacy (SP)*. IEEE. 2017, pp. 3–18.

[25] Filippo Simini, Gianni Barlacchi, Massimilano Luca, and Luca Pappalardo. "A Deep Gravity model for mobility flows generation". In: *Nature communications* 12.1 (2021), pp. 1–13.

[26] Chaoming Song, Tal Koren, Pu Wang, and Albert-László Barabási. "Modelling the scaling properties of human mobility". In: *Nature physics* 6.10 (2010), pp. 818–823.

[27] Chaoming Song, Zehui Qu, Nicholas Blumm, and Albert-László Barabási. "Limits of predictability in human mobility". In: *Science* 327.5968 (2010), pp. 1018–1021.

[28] Jinzhong Wang, Xiangjie Kong, Feng Xia, and Lijun Sun. "Urban human mobility: Data-driven modeling and prediction". In: *Acm Sigkdd Explorations Newsletter* 21.1 (2019), pp. 1–19.

[29] Chugui Xu, Ju Ren, Deyu Zhang, Yaoxue Zhang, Zhan Qin, and Kui Ren. "GANobfuscator: Mitigating information leakage under GAN via differential privacy". In: *IEEE TIFS* 14.9 (2019), pp. 2358–2371.

[30] Xiao-Yong Yan, Wen-Xu Wang, Zi-You Gao, and Ying-Cheng Lai. "Universal model of individual and population mobility on diverse spatial scales". In: *Nature communications* 8.1 (2017), p. 1639.

[31] Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. "Seqgan: Sequence generative adversarial nets with policy gradient". In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 31. 1. 2017.

Supplementary Algorithms

Algorithm 1 DeepMobility

Input: Expert individual trajectories X_E, expert population flows F, threshold N_P

1: initialize π_θ, D_ϕ, v_I, and Q_P with random parameters, initialize the trajectory set $X_P = \{\}$
2: **for** $i \leftarrow 1, 2, ..., N$ **do**
3: Generate a batch of individual trajectories $X_i \sim \pi_{\theta_i}$, $X_P = X_P \cup X_i$
4: **Train discriminator (step 5):**
5: Update parameters of D_ϕ based on Adam Optimizer with the gradient

$$\hat{\mathbb{E}}_{X_i}[\nabla_\phi \log(D_\phi(o,a)] + \hat{\mathbb{E}}_{X_E}[\nabla_\phi \log(1 - D_\phi(o,a)]$$

6: **for each** state-action pair (o,a) **do**
7: Compute reward $r(o,a)$, individual value $v_I(o)$, population-aware value $Q_p(o,a)$
8: **end for**
9: **Train individual-level and population-level critics (step 10-14):**
10: Update parameters of the individual-level critic according to Equation (5)
11: **if** Count$(X_P) \geq N_P$ **then**
12: Update parameters of the population-level critic
13: $X_P = \{\}$
14: **end if**
15: **Train generator (step 16-17):**
16: Calculate A according to Equation (6)
17: Update parameters of π_θ via PPO method

$$\hat{\mathbb{E}}[\nabla_\theta \min(\frac{\pi_{\theta_{old}}}{\pi_\theta} A, clip(\frac{\pi_{\theta_{old}}}{\pi_\theta}, 1-\varepsilon, 1+\varepsilon)A)] - \lambda \nabla_\theta H(\pi_\theta)$$

18: **end for**

Algorithm 2 Urban mobility generation (generating one individual as an example)

Input: Parameter of the policy function π_θ, start time $t = 0$, trajectory length T
Output: An individual trajectory $X = [x_1, x_2, ..., x_T], x_i = (r_i, l_i)$

1: initialize $X = \{x_1\}$ randomly, $t = 1$
2: **while** $t < T$ **do**
3:     **First stage (step 4-12):**
4:     Calculate IndividualDecision $P_I = \pi_{\theta_I}(a_t | X_{<t})$
5:     Calculate PopulationDecision $P_P = \pi_{\theta_P}(a_t | \tilde{\mathbf{F}}_{l_{t-1}})$
6:     Estimate state uncertainty u_t
7:     Sample a decision mechanism DM according to a Bernoulli distribution parameterized by uncertainty

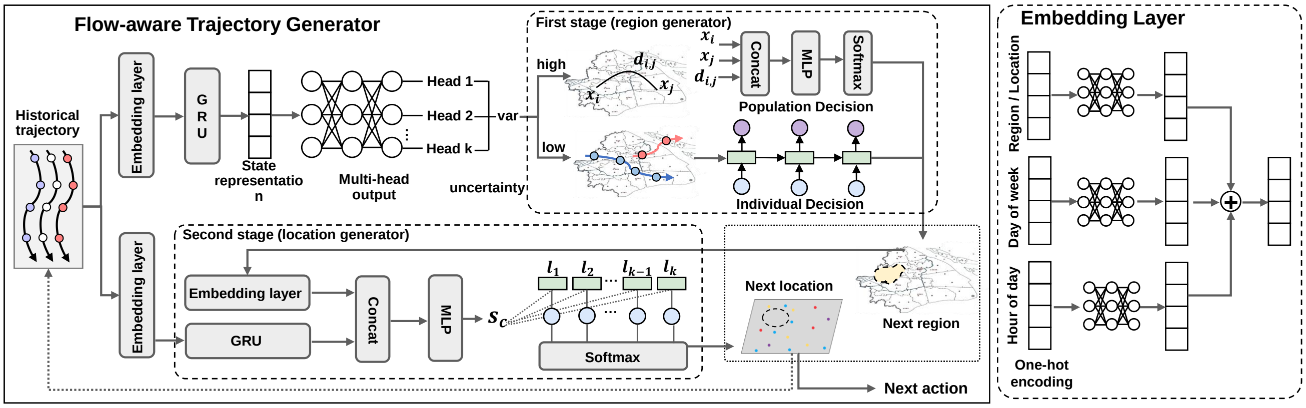
$$\pi(a_t | o_t) = \begin{cases} P_I, \ \texttt{Prob} = 1 - u_t \\ P_P, \ \texttt{Prob} = u_t \end{cases}$$

8:     **if** $DM =$ IndividualDecision **then**
9:         Sample the next region r_{t+1} according to probability distribution P_I
10:     **else**
11:         Sample the next region r_{t+1} according to probability distribution P_P
12:     **end if**
13:     **Second stage (step 14-15):**
14:     Calculate the attention weights
15:     Sample a location according to the attention weights α:

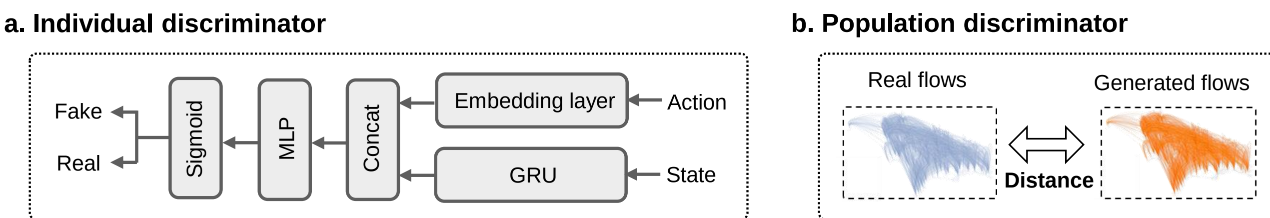
$$l_{t+1} \sim PN(N : \alpha_0, \alpha_1, ..., \alpha_{N_{l,r}}), \text{ where } N_{l,r} \text{ is the number of locations in region } r_{t+1}$$

16:     $X = X \cup \{(r_{t+1}, l_{t+1})\}$
17:     $t = t + 1$
18: **end while**

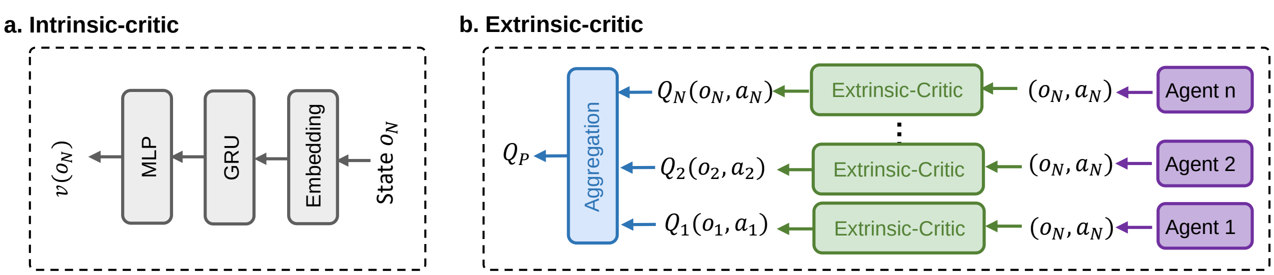
Supplementary Figures



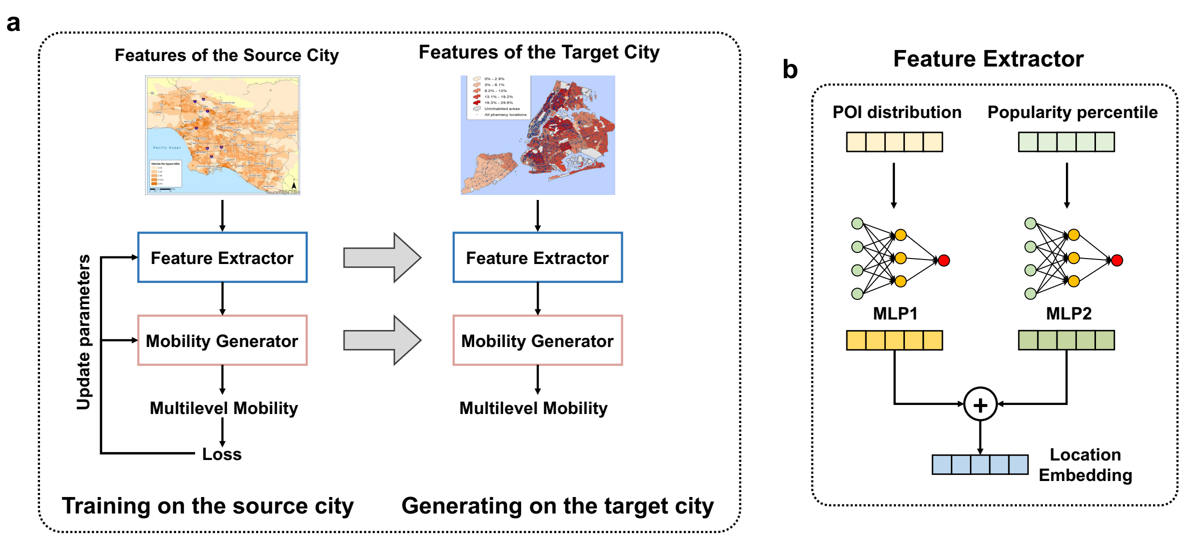
Supplementary Fig. 1. Network architecture of the global-aware trajectory generator. It consists of two stages: region generation and location generation. The embedding layer employs three MLPs to encode features of location ID, day of the week, and hour of the day.



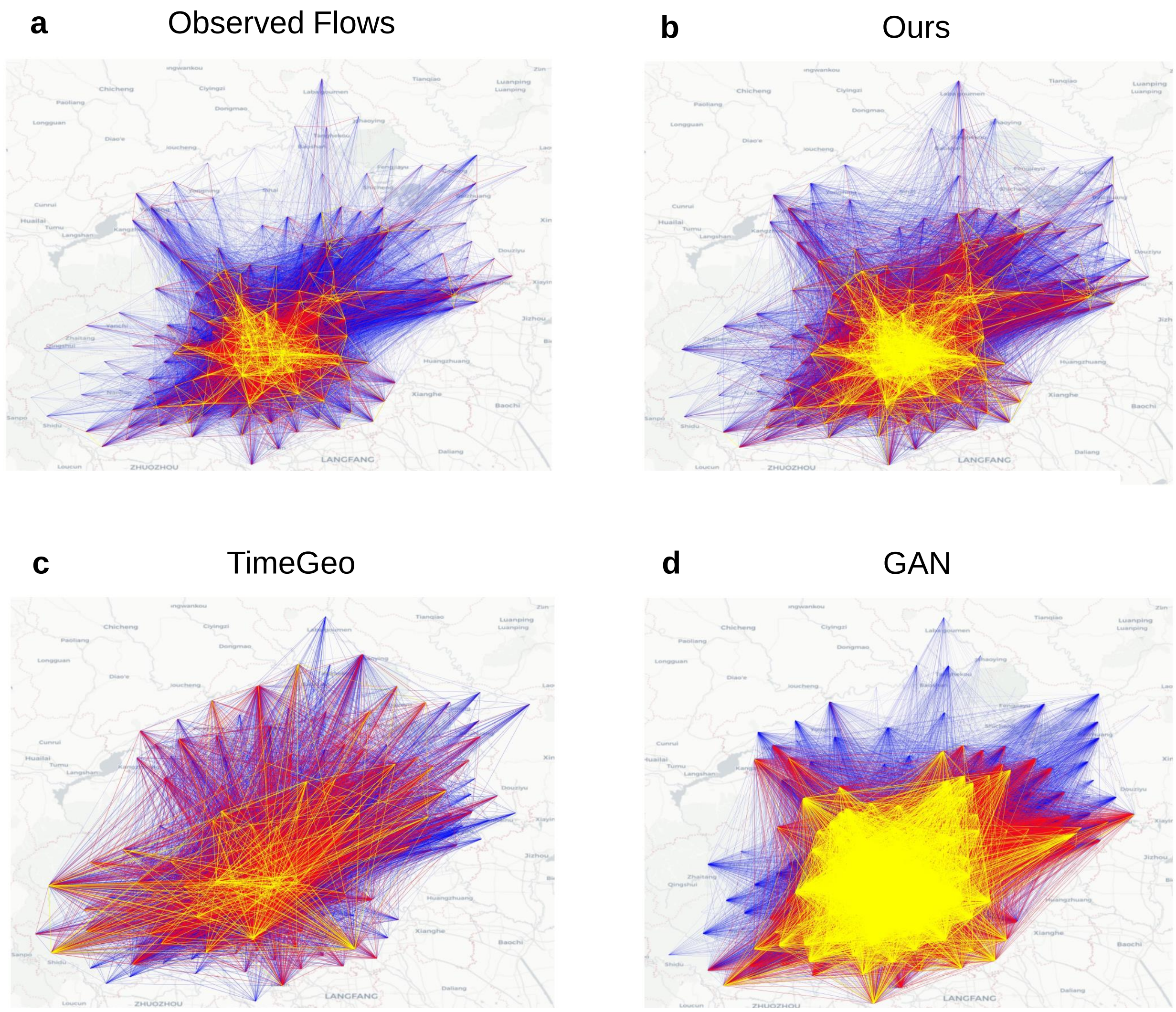
Supplementary Fig. 2. Network architecture to model multilevel reward functions.



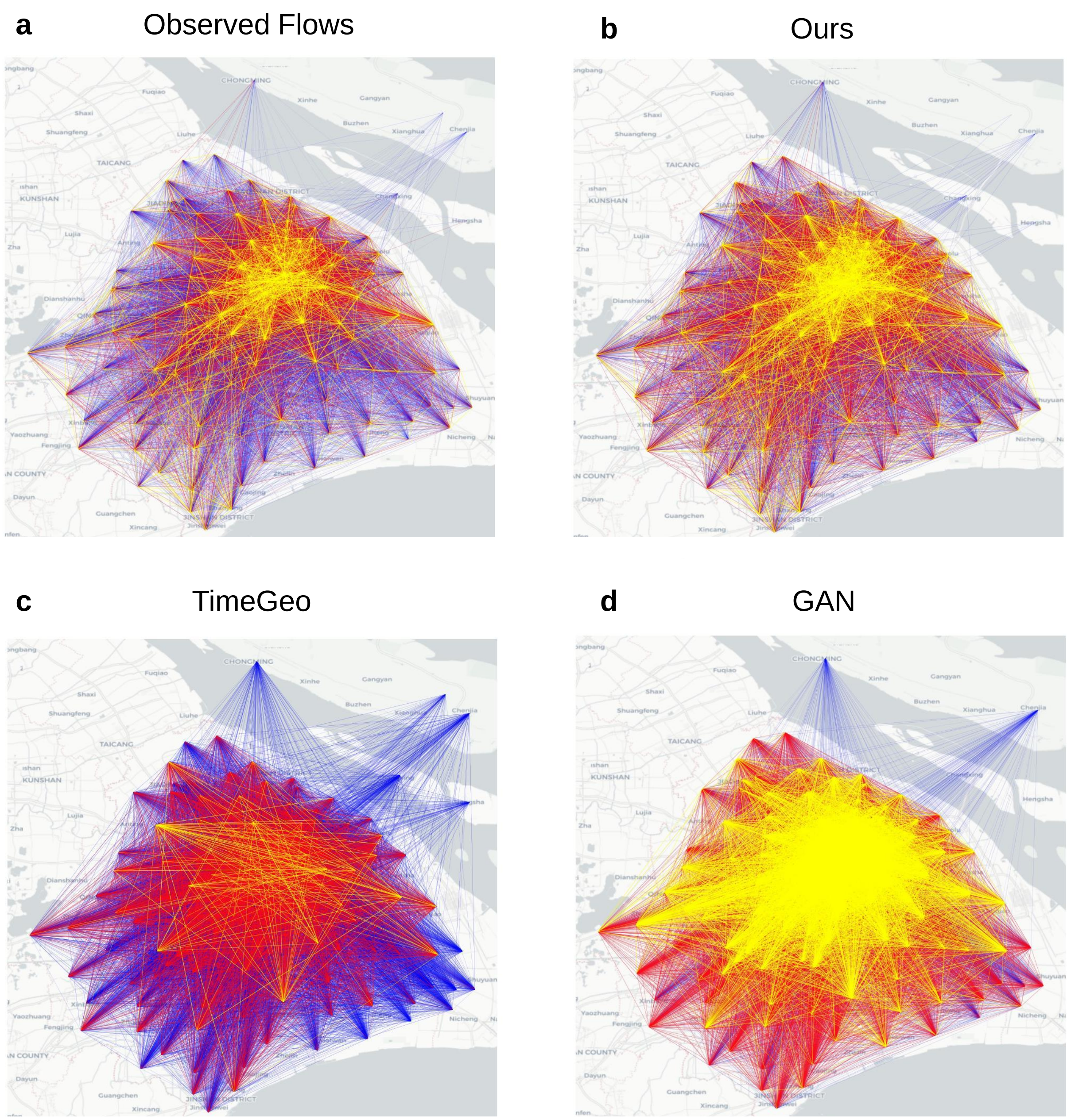
Supplementary Fig. 3. Network architecture of the *individual-level critic* and *population-level critic*.



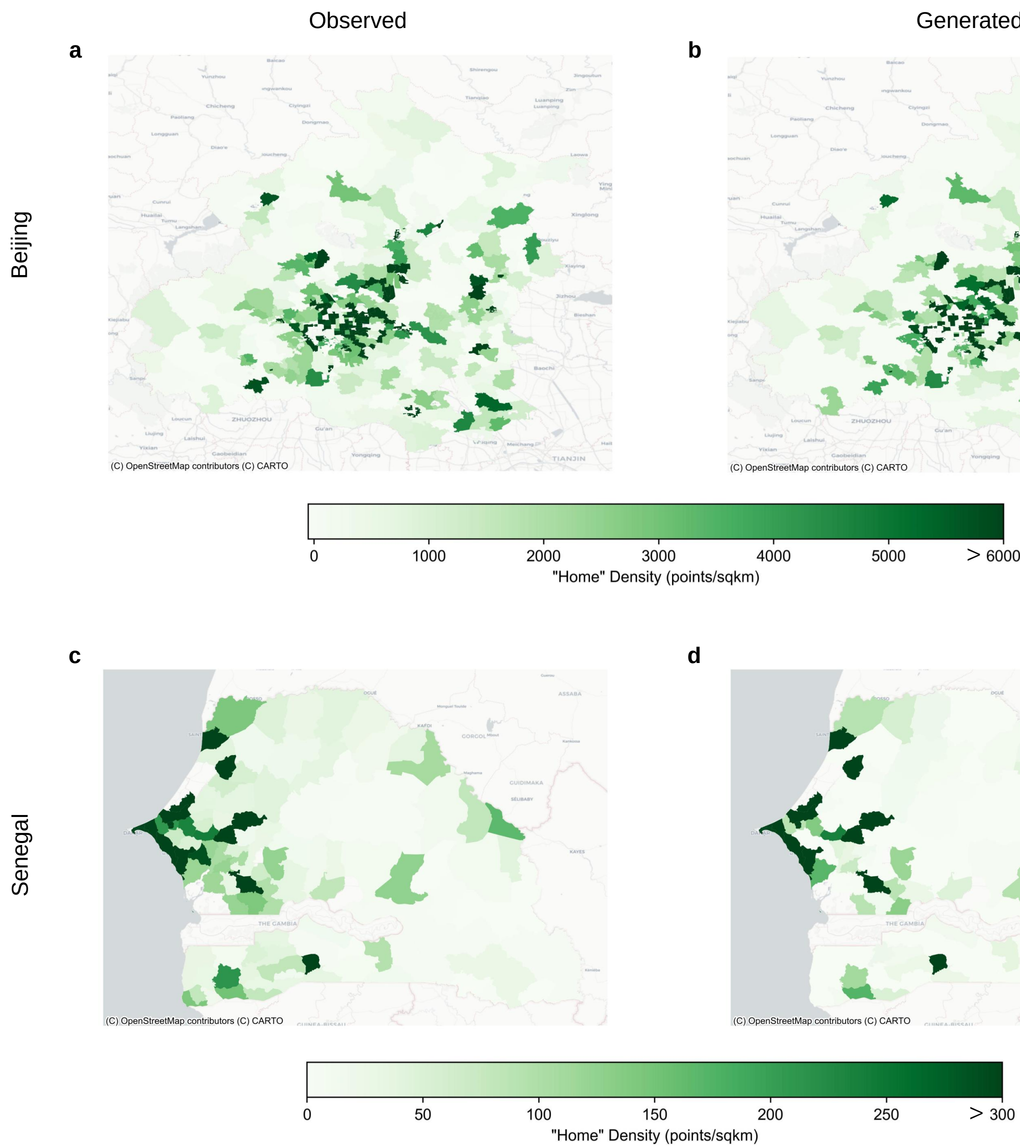
Supplementary Fig. 4. **a. Model transfer from the source city to the target city. b. The network architecture of the used feature extractor to obtain transferable location representation.**



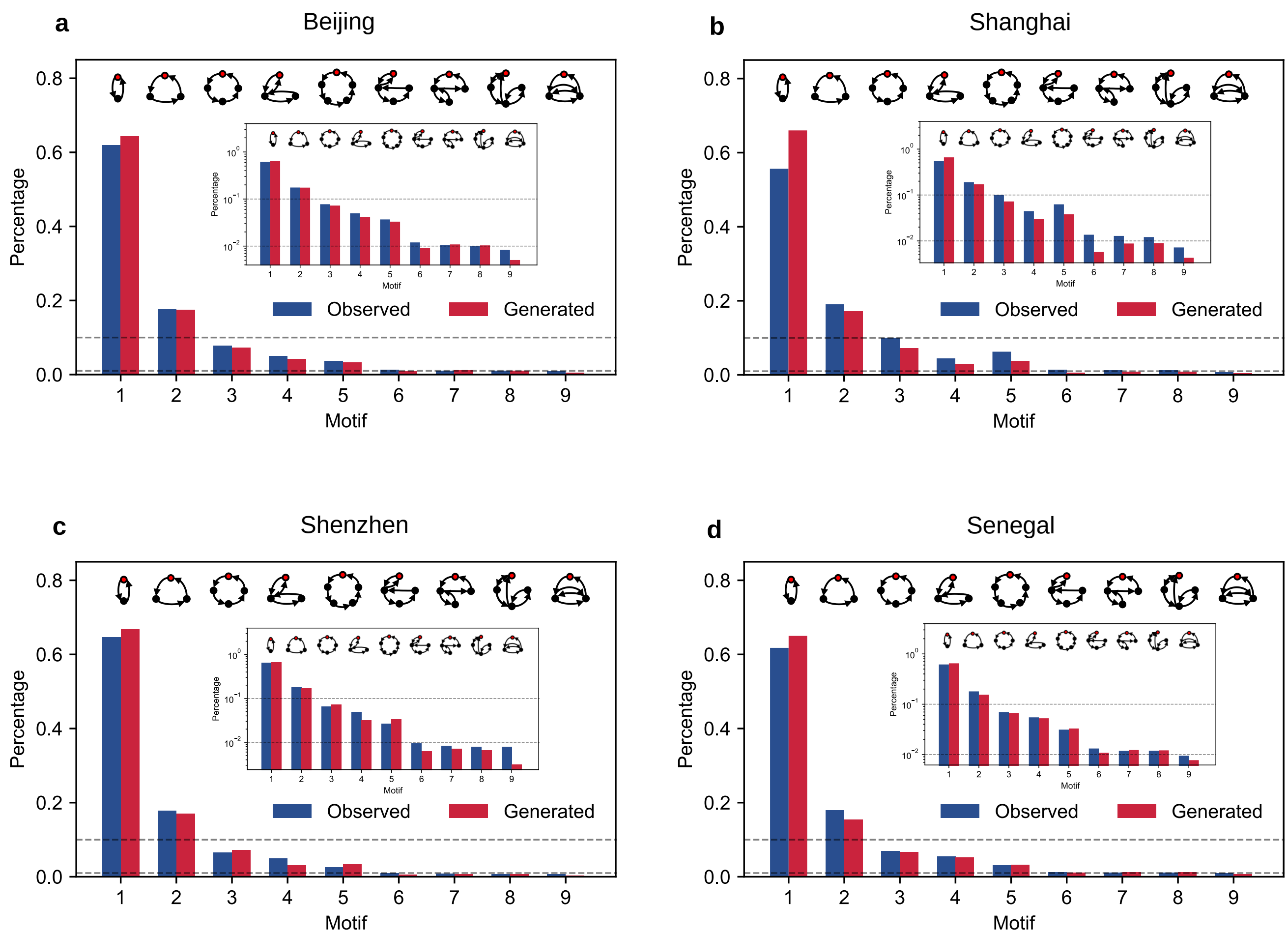
Supplementary Fig. 5. Observed flows versus generated flows of Beijing. Visualization of real population flows (**a**), the flows generated by our framework (DeepMobility, panel **b**), the flows generated by the TimeGeo (panel **c**), and the flows generated by the GAN (panel **d**) on 318 regions of interest in Beijing, China. Colored edges represent observed flows (**a**) or generated flows (**b,c**) between two regions. The colors from blue to yellow denote the increase in population flows. DeepMobility captures the overall spatial distribution of population flows more accurately than TimeGeo and GAN.



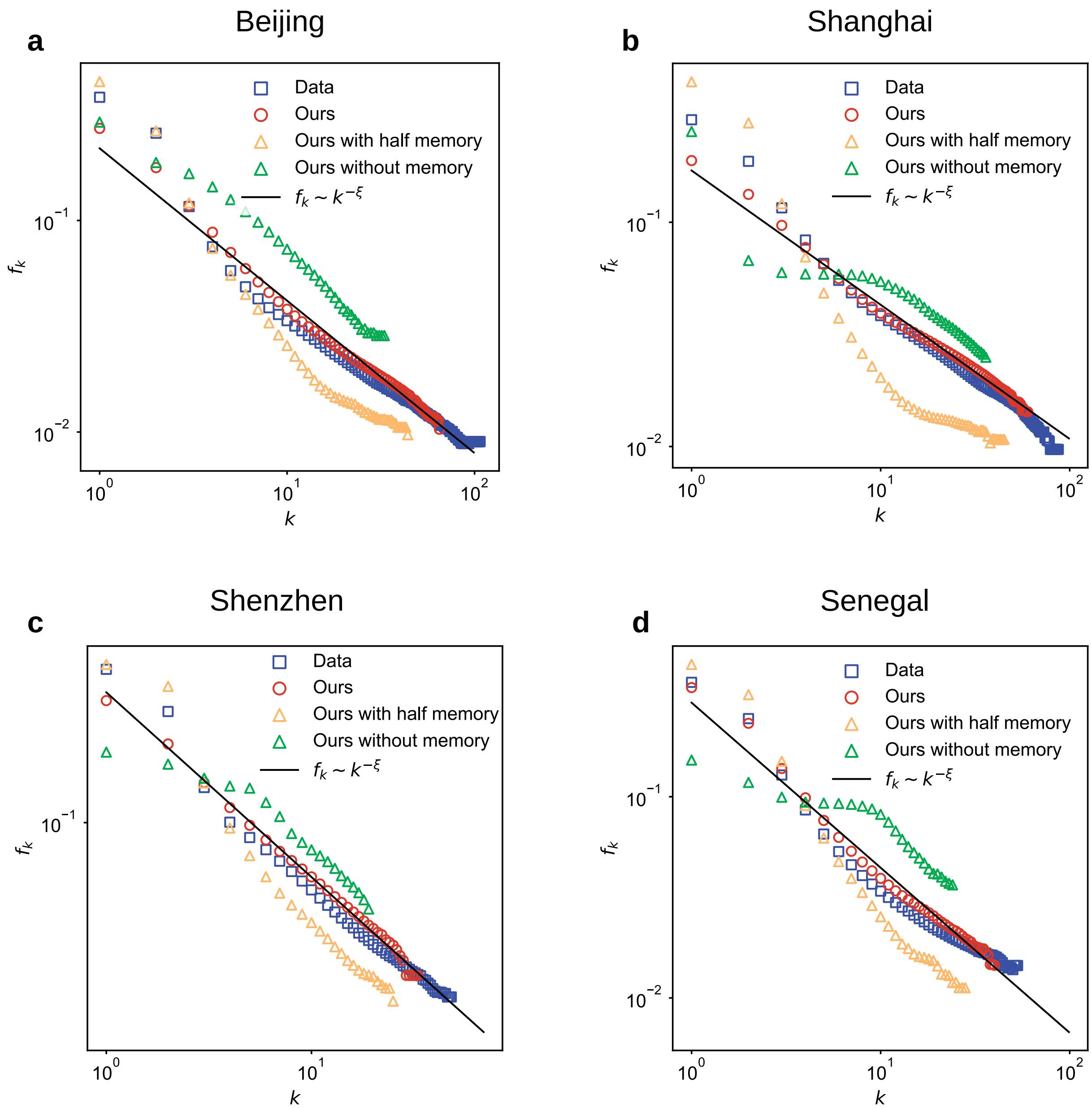
Supplementary Fig. 6. Observed flows versus generated flows of Shanghai. Visualization of real population flows (**a**), the flows generated by our framework (DeepMobility, panel **b**), the flows generated by the TimeGeo (panel **c**), and the flows generated by the GAN (panel **d**) on 200 regions of interest in Shanghai, China. Colored edges represent observed flows (**a**) or generated flows (**b,c**) between two regions. The colors from blue to yellow denote the increase in population flows. DeepMobility captures the overall spatial distribution of population flows more accurately than TimeGeo and GAN.



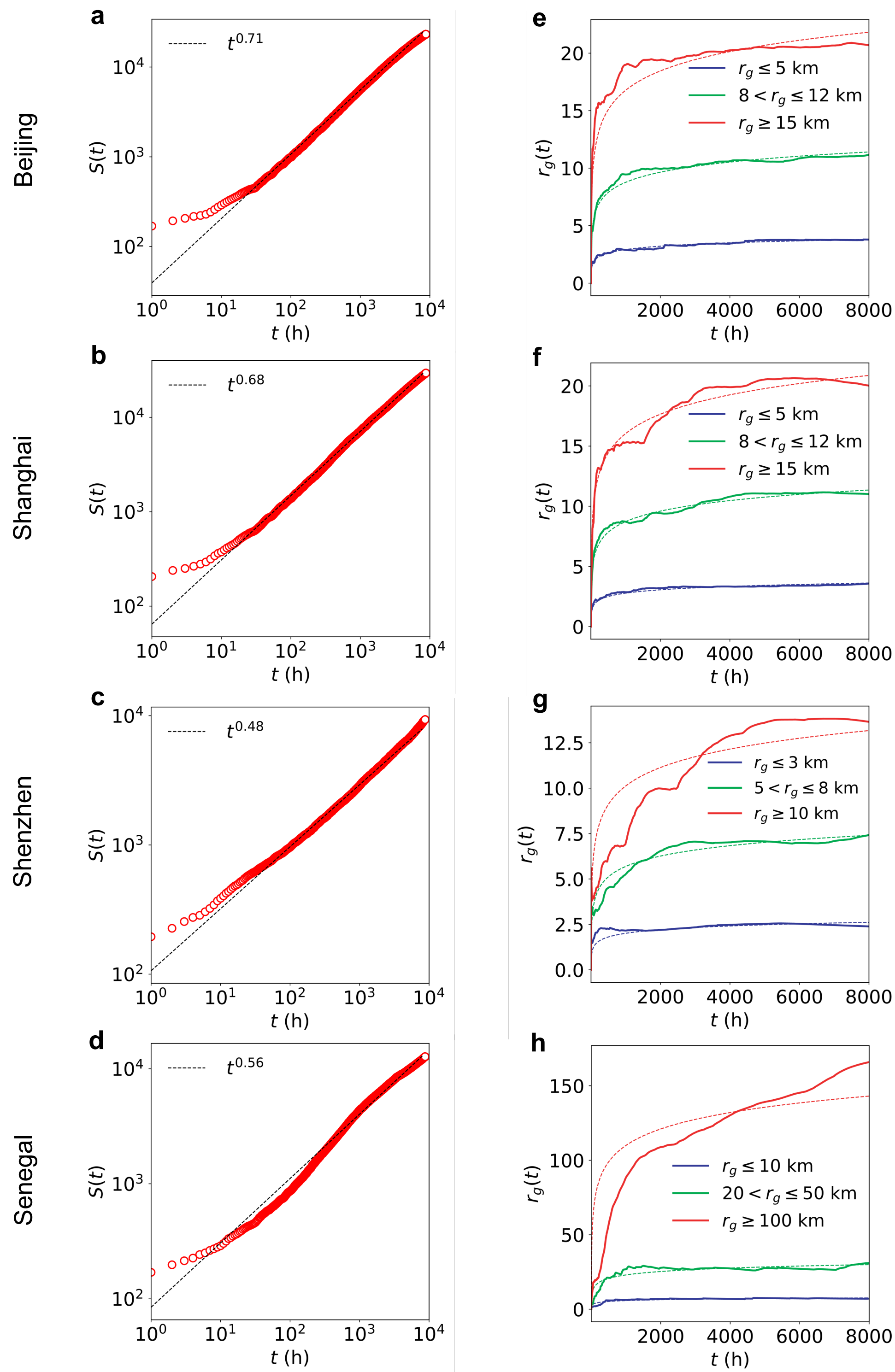
Supplementary Fig. 7. The distribution of home locations in Beijing and Senegal at the region resolution, versus the distribution of identified home locations in the generated urban mobility data. The home is inferred as an individual's most common location between 11 p.m. and 6 a.m.



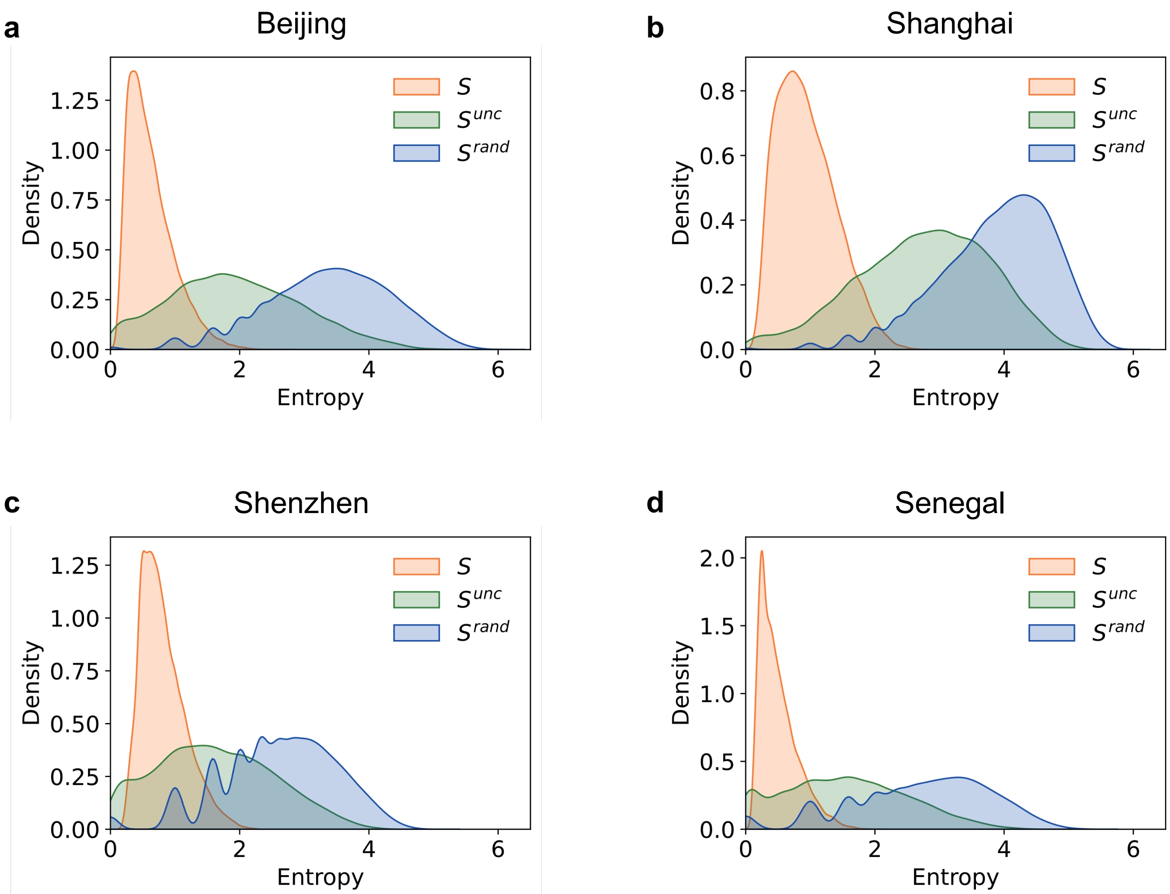
Supplementary Fig. 8. Most frequented motifs associated with individual's mobility pattern. Each individual's movement trajectory is constructed as a network, where the nodes represent locations visited and directed edges represent movement transitions between the locations. From the four empirical datasets, nine typical distinct motifs are identified following previous works[30,14]. It was found that more than 95% of the recorded movements made by all individuals can be described with these 9 motifs. Some more possible mobility networks are not considered to be motifs as they occur less than 0.1%. The occurrence frequencies of motifs from real-world trajectories and generation outcomes show a generally strong agreement with each other, with JSD statistics of 0.025, 0.084, 0.047, and 0.029 in four cities. The model is able to capture well the higher probability of motifs with 2 or 3 nodes.



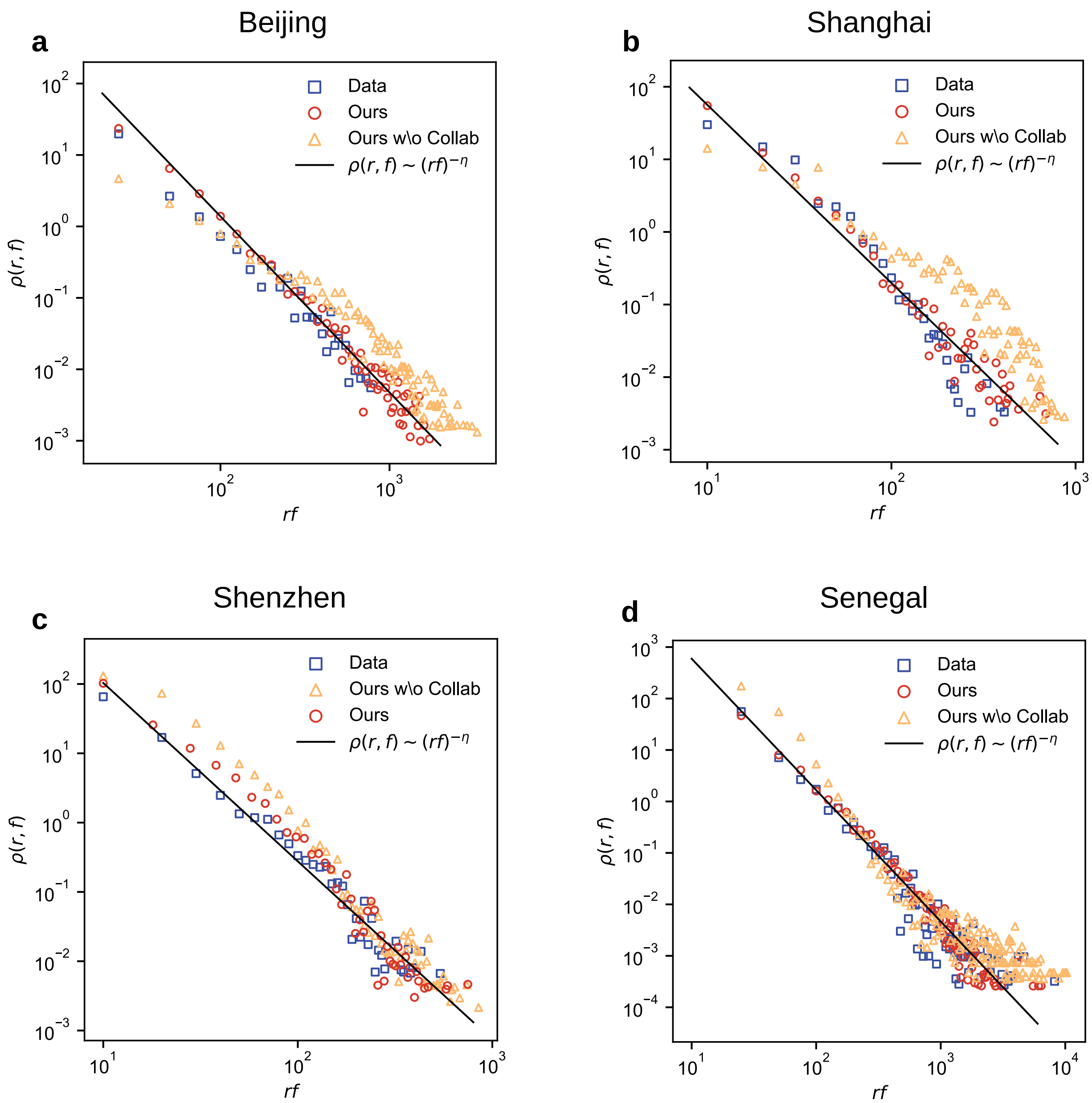
Supplementary Fig. 9. The visitation distribution comparison between DeepMobility and its variants with the reduced memory effect. There are two variants: (1) only retaining a half-length of the visitation history (yellow color), and (2) totally removing the visitation history (green color). With a half-length of the visitation history, individuals' behaviors become more concentrated, i.e., a steeper distribution, however, when totally leaving out the visitation history, individuals exhibit more random visitations, i.e., a more uniform distribution.



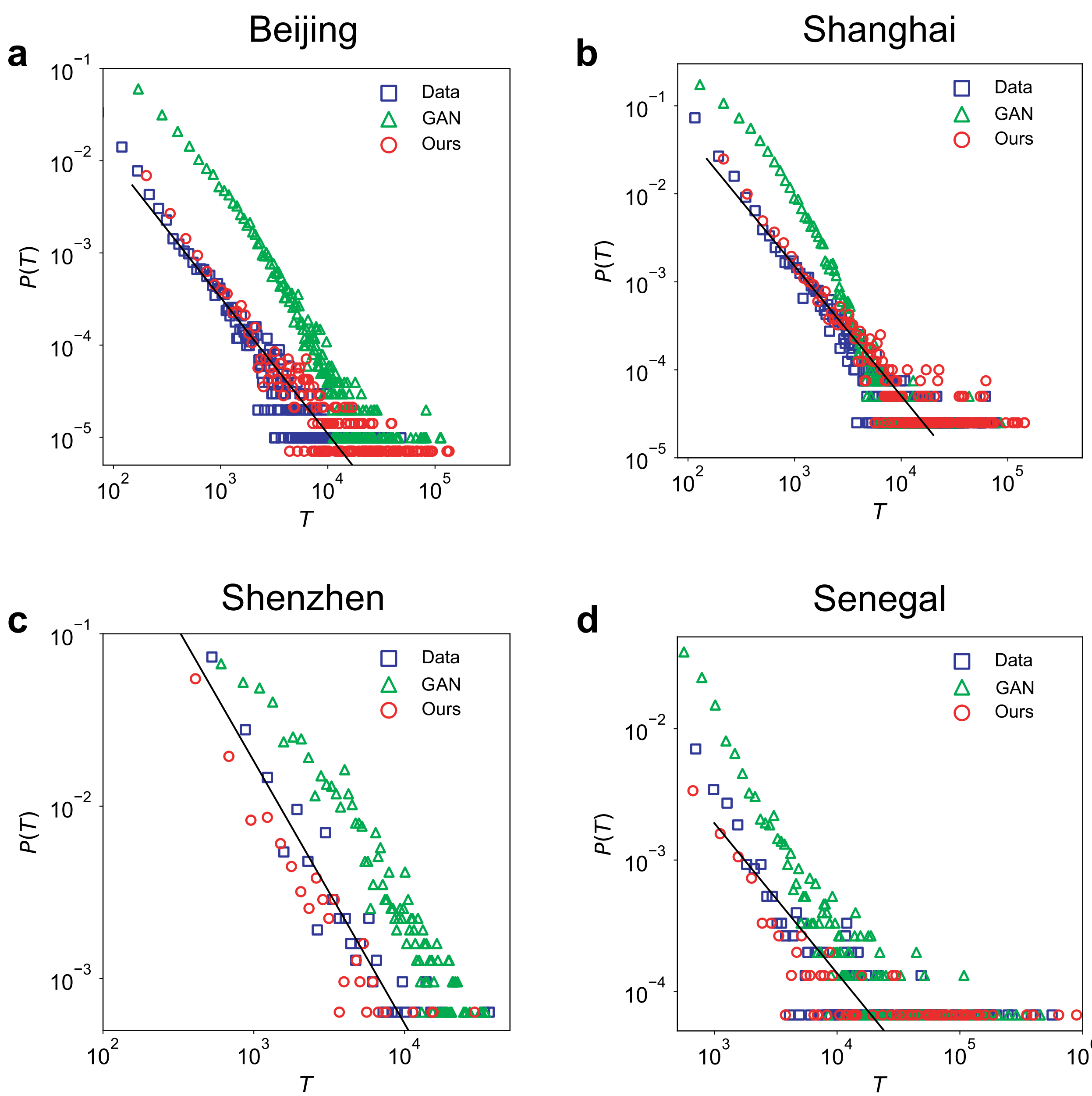
Supplementary Fig. 10. Sublinear growth of the distinct visited location number and ultraslow growth of the radius of gyration. a-d Total number of visited distinct locations $S(t)$ within time t in a log-log plot. The black dashed line represents the fitting result of t, $R^2 = \{0.990, 0.997, 0.986, 0.987\}$. **e-h** The evolution of the radius of gyration $r_g(t)$ versus time t for different r_g groups. To compare the increase characteristics for different $r_g(t)$. They are all ultraslow growths within t for different r_g groups.



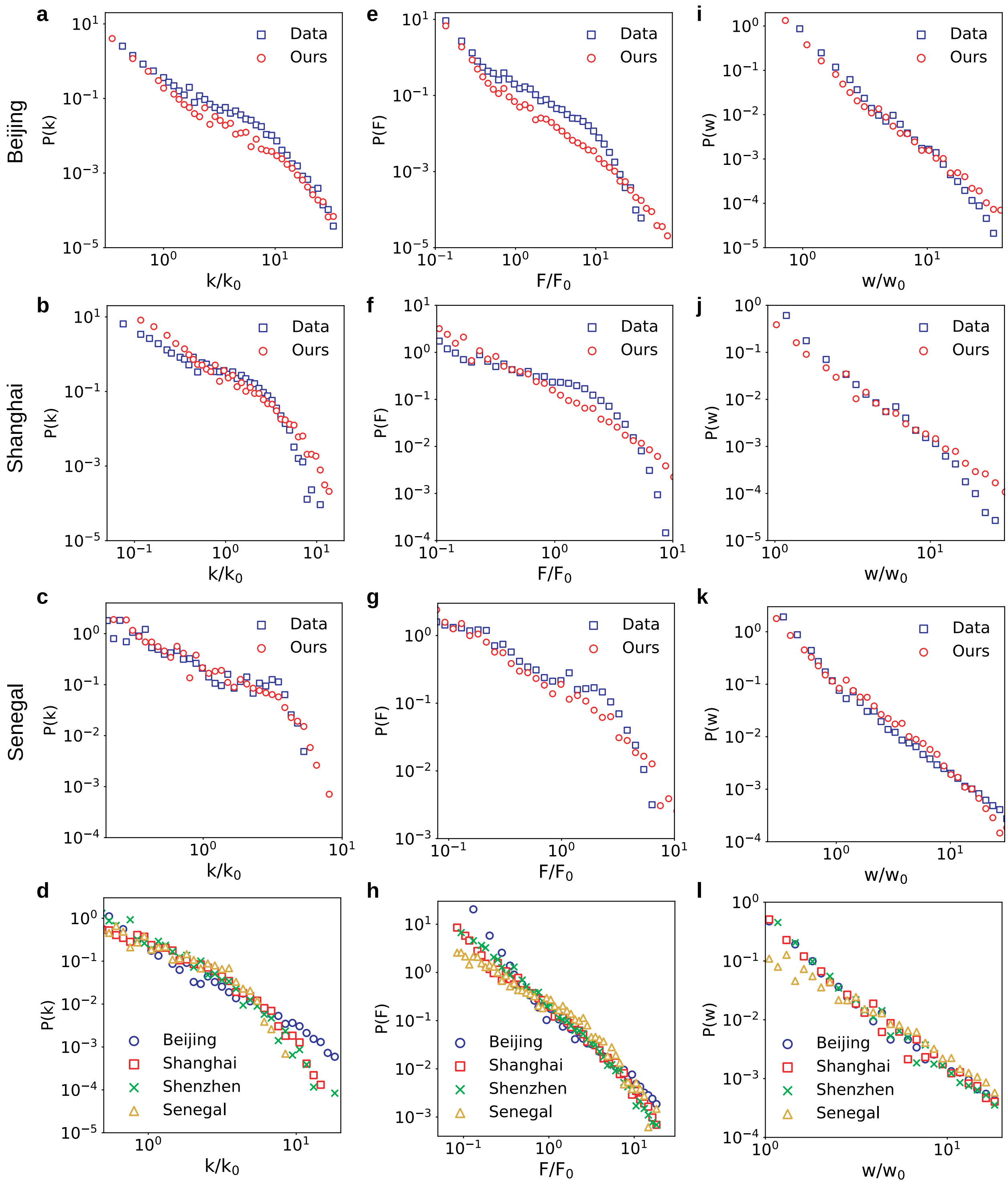
Supplementary Fig. 11. The distribution of three entropy measures: the actual entropy S, the random entropy S^{rand}, and the uncorrelated entropy S^{unc} across users in four cities.



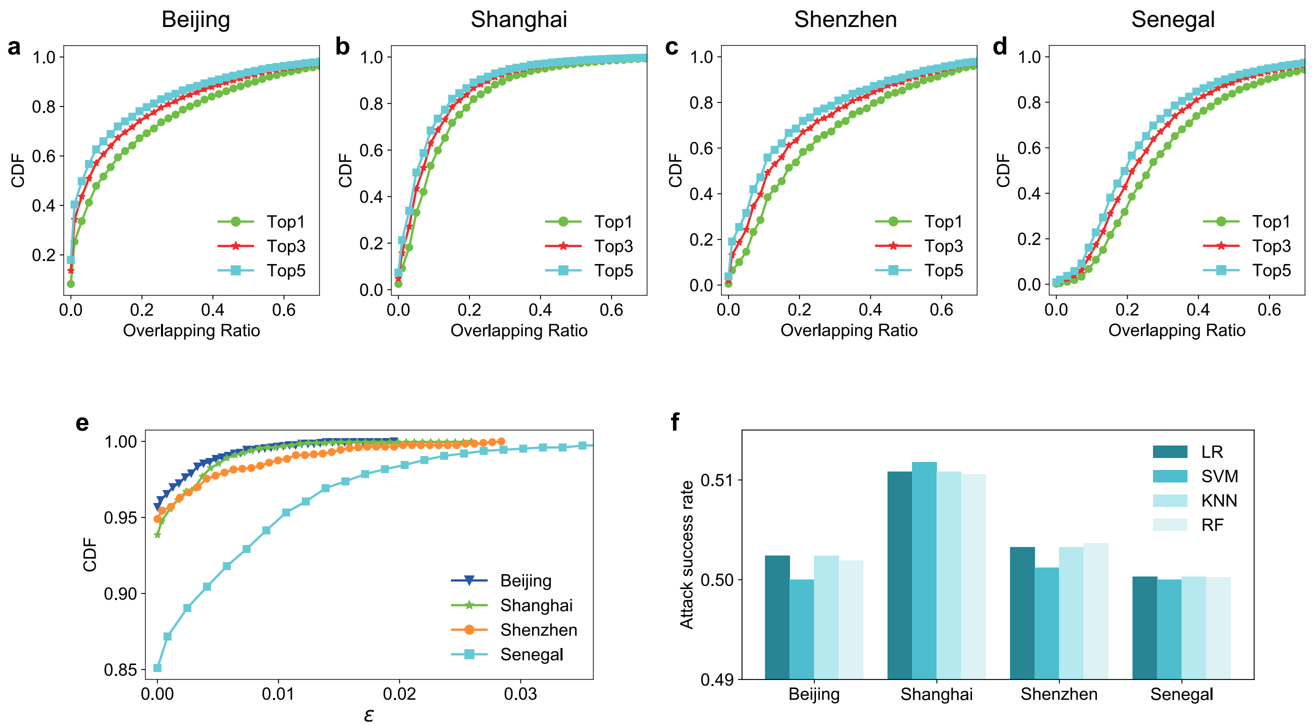
Supplementary Fig. 12. **The results of distance-frequency scaling for DeepMobility removing collaborative learning**.



Supplementary Fig. 13. Empirical results versus generation results regarding the number of trips, T, between two regions. The black lines represent a power-law fit to the empirical data. The green labels show the result of GAN.



Supplementary Fig. 14. Statistical properties of the trip networks, where nodes represent locations. **a-d** node degree k, **e-h** node-level flow F, and **i-l** orgin-destination flow w normalized by the mean value k_0, F_0, and w_0.



Supplementary Fig. 15. Privacy evaluation in terms of uniqueness testing, differential privacy and membership inference attack. The overlapping ratio between two mobility trajectories is used to represent the uniqueness of the generated trajectory. **a-d** Accumulative distributions of the overlapping ratio associated with the most similar trajectory (top1), the most 3 similar trajectories(top3), and the most 5 similar trajectories (top5). For the four empirical datasets, more than 80% generated trajectories cannot find any real trajectory with overlapping ratio more than 0.4. **e** Accumulative distributions of ε for the four empirical datasets. ε-differential privacy measures the privacy loss if any sample is removed from the training datasetm, thus a lower ε denotes a better privacy preserving performance. **f** Success rate of membership inference attack. Four widely-adopted classification models are used to eliminate the impact of different attack approaches. For all classification models on the four empirical datasets, the success rates are all less than 0.52, indicating it is difficult to reidentify individuals in the training set.

Supplementary Tables

Notation	Description
U	The set of users.
G_r	The set of geographical regions.
L	The set of geographical locations.
T	The set of time slots.
X_u	The user trajectory for user $u \in U$.
F	The population flows between regions.
S	The state space.
A	The action space.
π_θ	The policy function.
D_ϕ	The individual-level discriminator.
D_P	The population-level discriminator.
R	The reward function.

Supplementary Table. 1. A list of commonly used notations.

Group	Hyper-parameter	Values
Basic network	Number of MLP layers (PopulationDecision)	10
	Number of MLP layers (remaining)	4
	Hidden size of MLP layers	64
	Activation of MLP layers	LeakyReLU
	Number of GRU layers	1
	Hidden size of GRU layers	64
	Activation of GRU Cell	Tanh
	Embedding size of locations	32
	Embedding size of regions	16
	Embedding size of day of week	8
	Embedding size of hour of day	16
PPO Training	Threshold N_P	50,000
	Total training iterations	2000
	PPO epochs	5
	Minibatch size of PPO	2048
	Discriminator epochs	3
	Minibatch size of discriminator	2048
	Discount (γ)	0.99
	GAE parameter (λ)	0.95
	Clipping parameter ε	0.2
Overall training	Optimizer	Adam
	Learning rate	0.0001
	Generation batch size	128

Supplementary Table. 2. Hyper-parameters for DeepMobility

	Individual-level metrics					Population-level metrics	
Model	dailyloc	jump length	trip distance	duration	radius	MAE	CPC
Beijing							
CTRW	0.207	0.487	0.603	0.154	0.549	148.1	8.751e-5
EPR	0.477	0.485	0.301	0.352	0.548	148.2	0.002
r-EPR	0.476	0.484	0.302	0.351	0.548	148.2	0.001
d-EPR	0.476	0.483	0.297	0.352	0.547	148.1	0.002
MPRW	0.446	0.451	0.206	0.330	0.518	154.0	3.478e-4
TimeGeo	0.119	<u>0.071</u>	<u>0.155</u>	<u>0.034</u>	<u>0.124</u>	140.6	0.153
LSTM	0.329	0.192	0.529	0.058	0.430	189.9	0.303
GAN	<u>0.098</u>	0.174	0.360	0.046	0.343	<u>127.7</u>	<u>0.357</u>
DeepMobility	**0.021**	**0.058**	**0.023**	**0.026**	**0.044**	**22.68**	**0.787**
Shanghai							
CTRW	0.305	0.494	0.501	0.198	0.523	326.9	0.007
EPR	0.560	0.488	0.121	0.387	0.513	284.3	0.045
r-EPR	0.561	0.491	0.117	0.385	0.513	284.4	0.045
d-EPR	0.560	0.392	0.071	0.387	0.237	276.4	0.237
MPRW	0.561	0.491	<u>0.067</u>	0.384	0.458	292.8	0.015
TimeGeo	<u>0.203</u>	0.203	0.247	0.055	<u>0.186</u>	<u>246.6</u>	0.237
LSTM	0.445	0.218	0.497	0.047	0.489	426.6	0.283
GAN	0.240	<u>0.201</u>	0.459	<u>0.034</u>	0.446	326.9	<u>0.324</u>
DeepMobility	**0.003**	**0.029**	**0.008**	**0.005**	**0.025**	**63.12**	**0.686**
Shenzhen							
CTRW	0.189	0.522	0.506	0.118	0.617	2064.9	0.008
EPR	0.631	0.516	0.255	0.271	0.581	1597.8	0.087
r-EPR	0.628	0.518	0.255	0.270	0.582	1599.2	0.086
d-EPR	0.632	0.454	0.104	0.271	0.409	1546.3	0.114
MPRW	0.624	0.522	0.257	0.267	0.567	1720.9	0.017
TimeGeo	**0.028**	<u>0.058</u>	<u>0.134</u>	**0.004**	<u>0.085</u>	<u>1415.9</u>	<u>0.216</u>
LSTM	0.605	0.276	0.501	0.157	0.570	4155.5	0.099
GAN	0.237	0.256	0.357	0.123	0.434	2146.6	0.210
DeepMobility	<u>0.070</u>	**0.046**	**0.019**	<u>0.015</u>	**0.073**	**253.0**	**0.560**
Senegal							
CTRW	0.195	0.485	0.432	0.164	0.477	1176.7	0.007
EPR	0.453	0.055	0.094	0.374	0.022	767.8	0.074
r-EPR	0.453	0.054	0.094	0.373	**0.022**	767.7	0.074
d-EPR	0.453	<u>0.052</u>	0.087	0.374	0.034	769.5	0.074
MPRW	0.439	0.487	0.198	0.346	0.245	<u>648.7</u>	0.036
TimeGeo	<u>0.134</u>	0.093	<u>0.082</u>	0.042	0.103	732.0	<u>0.359</u>
LSTM	0.385	0.117	0.424	0.041	0.401	1607.9	0.281
GAN	0.175	0.105	0.386	<u>0.028</u>	0.368	1127.7	0.350
DeepMobility	**0.010**	**0.037**	**0.019**	**0.006**	<u>0.034</u>	**223.0**	**0.650**

Supplementary Table. 3. Comparison of the performance, regarding the generation fidelity from both individual and population perspectives. The baselines approaches consist of Continuous-time Random Walk (CTRW), exploration and preferential return model (EPR), variants of EPR models including r-EPR, d-EPR, and TimeGeo, Markov, Long short-term memory (LSTM), Sequence Generative Adversarial Nets (GAN). The individual-level evaluation uses JSD-based mobility metrics, including the number of daily visited locations (dailyloc), the displacements between two locations (jump length), accumulative distance traveled by an individual in a week (trip distance), staying duration at one location (duration), and radius of gyration (radius). We also provide a population evaluation of the goodness of each model in terms of Common Part of Commuters (CPC) and Mean Absolute Error (MAE). Regardless of the evaluation perspectives and metrics, DeepMobility significantly outperforms other models in all cities considered.

	Individual-level metrics				
Model	dailyloc	jump length	trip distance	duration	radius
Beijing					
CTRW	0.419	0.999	0.998	0.379	0.999
EPR	0.914	0.988	0.555	0.716	0.994
r-EPR	0.913	0.992	0.555	0.714	0.994
d-EPR	0.914	0.989	0.522	0.716	0.991
MPRW	0.911	0.992	0.525	0.711	0.987
TimeGeo	0.350	**0.168**	0.363	0.222	0.280
LSTM	0.553	0.871	0.983	0.361	0.988
GAN	0.210	0.870	0.908	0.355	0.973
DeepMobility	**0.187**	0.324	**0.078**	**0.108**	**0.186**
Shanghai					
CTRW	0.688	0.993	0.969	0.531	0.999
EPR	0.969	0.957	0.193	0.747	0.895
r-EPR	0.969	0.962	0.186	0.745	0.895
d-EPR	0.970	0.740	0.277	0.747	0.556
MPRW	0.967	0.921	0.190	0.743	0.890
TimeGeo	0.510	0.218	0.588	0.332	0.500
LSTM	0.836	0.590	0.973	0.336	0.872
GAN	0.609	0.565	0.903	0.319	0.830
DeepMobility	**0.095**	**0.166**	**0.152**	**0.062**	**0.135**
Shenzhen					
CTRW	0.458	0.997	0.984	0.369	0.999
EPR	0.995	0.964	0.583	0.618	0.941
r-EPR	0.994	0.967	0.583	0.616	0.941
d-EPR	0.995	0.817	0.384	0.618	0.729
MPRW	0.992	0.968	0.617	0.615	0.954
TimeGeo	0.263	**0.193**	0.264	**0.059**	**0.351**
LSTM	0.915	0.640	0.979	0.509	0.917
GAN	0.679	0.617	0.883	0.487	0.860
DeepMobility	**0.247**	0.294	**0.185**	0.127	0.378
Senegal					
CTRW	0.463	0.996	0.996	0.463	0.994
EPR	0.989	0.290	0.432	0.746	0.170
r-EPR	0.988	0.288	0.432	0.745	0.170
d-EPR	0.988	0.290	0.413	0.746	0.209
MPRW	0.986	0.962	0.666	0.744	0.919
TimeGeo	0.406	0.387	0.359	0.289	0.423
LSTM	0.757	0.435	0.908	0.282	0.781
GAN	0.515	0.405	0.839	0.268	0.746
DeepMobility	**0.155**	**0.273**	**0.078**	**0.063**	**0.159**

Supplementary Table. 4. Comparison of the KS statistics, regarding the individual-level generation fidelity

From Beijing to Shanghai	Individual-level metrics					Population-level metrics	
Model	dailyloc	jump length	trip distance	duration	radius	MAE	CPC
From Beijing to Shanghai							
CTRW	0.305	0.494	0.501	0.198	0.523	326.9	0.007
EPR	0.560	0.488	0.121	0.387	0.513	284.3	0.045
r-EPR	0.561	0.491	0.118	0.385	0.513	284.4	0.045
d-EPR	0.560	0.392	<u>0.071</u>	0.387	0.237	276.4	0.072
TimeGeo	<u>0.255</u>	<u>0.061</u>	0.294	<u>0.078</u>	<u>0.196</u>	<u>254.6</u>	<u>0.191</u>
DeepMobility	**0.196**	**0.059**	**0.025**	**0.046**	**0.002**	**121.3**	**0.402**
From Beijing to Shenzhen							
CTRW	0.190	0.522	0.506	0.117	0.617	2072.3	0.008
EPR	0.688	0.518	0.201	0.333	0.171	1681.0	0.026
r-EPR	0.690	0.519	0.201	0.342	0.170	1680.7	0.026
d-EPR	0.685	0.488	**0.138**	0.335	0.116	1663.0	0.037
TimeGeo	**0.036**	<u>0.059</u>	0.167	**0.003**	0.085	<u>1390.5</u>	<u>0.245</u>
DeepMobility	<u>0.398</u>	**0.056**	<u>0.164</u>	<u>0.094</u>	**0.079**	**341.8**	**0.464**
From Shanghai to Beijing							
CTRW	<u>0.207</u>	0.442	0.584	0.154	0.548	169.2	0.014
EPR	0.476	0.441	**0.237**	0.334	0.537	154.3	0.034
r-EPR	0.476	0.440	0.238	0.332	0.537	154.2	0.047
d-EPR	0.445	0.449	<u>0.227</u>	0.334	<u>0.537</u>	<u>147.5</u>	0.036
TimeGeo	**0.111**	<u>0.440</u>	0.553	**0.028**	0.545	152.1	<u>0.175</u>
DeepMobility	0.324	**0.067**	0.370	<u>0.093</u>	**0.299**	**226.5**	**0.301**
From Shanghai to Shenzhen							
CTRW	0.190	0.522	0.506	0.117	0.617	2072.3	0.008
EPR	0.634	0.515	0.225	0.271	0.545	1598.2	0.086
r-EPR	0.633	0.518	0.225	0.271	0.545	1599.7	0.086
d-EPR	0.634	0.453	**0.089**	0.271	0.373	1546.7	0.114
TimeGeo	<u>0.155</u>	<u>0.059</u>	0.167	**0.004**	<u>0.187</u>	<u>1376.6</u>	<u>0.262</u>
DeepMobility	**0.132**	**0.055**	<u>0.093</u>	<u>0.052</u>	**0.116**	**612.7**	**0.423**
From Shenzhen to Beijing							
CTRW	0.191	0.444	0.554	0.146	0.560	108.3	0.018
EPR	0.424	0.447	0.145	0.279	0.453	101.0	0.047
r-EPR	0.421	0.447	0.145	0.278	0.454	101.0	0.047
d-EPR	0.424	0.443	0.144	0.279	0.451	100.9	0.050
TimeGeo	<u>0.123</u>	<u>0.443</u>	**0.108**	<u>0.051</u>	**0.124**	<u>98.26</u>	<u>0.144</u>
DeepMobility	**0.078**	**0.052**	<u>0.125</u>	**0.044**	<u>0.146</u>	**89.84**	**0.334**
From Shenzhen to Shanghai							
CTRW	0.296	0.493	0.587	0.190	0.527	202.4	0.008
EPR	0.502	0.487	0.136	0.327	0.516	183.9	0.055
r-EPR	0.499	0.489	0.135	0.327	0.517	184.1	0.054
d-EPR	0.502	0.392	**0.024**	0.328	0.255	174.5	0.106
TimeGeo	<u>0.228</u>	<u>0.066</u>	0.257	<u>0.084</u>	<u>0.182</u>	<u>164.3</u>	<u>0.173</u>
DeepMobility	**0.089**	**0.022**	<u>0.048</u>	**0.025**	**0.035**	**131.8**	**0.306**

Supplementary Table. 5. Performance comparison in terms of the model transferability. Transfer experiments are carried out on Chinese cities, including Beijing, Shanghai, and Shenzhen. Model parameters are learned with data from the source city and models are evaluated on the target city. Three Chinese cities constitute six source-target city pairs.