Information Fusion 127 (2026) 103766

Contents lists available at ScienceDirect

Information Fusion

journal homepage: www.elsevier.com/locate/inffus

Traveller: Travel-pattern aware trajectory generation via autoregressive diffusion models

Yuxiao Luo [a,b], Songming Zhang [c], Kang Liu [b], Yang Xu [a,d,e], Ling Yin [b,c,*]

[a] Department of Land Surveying and Geo-informatics, The Hong Kong Polytechnic University, Hong Kong, China
[b] Shenzhen Institutes of Advanced Technology (SIAT), Chinese Academy of Sciences, 518055, Shenzhen Guangdong, China
[c] Faculty of Computer Science and Control Engineering, Shenzhen University of Advanced Technology, Shenzhen Guangdong, China
[d] The Hong Kong Polytechnic University Shenzhen Research Institute, Shenzhen Guangdong, China

ARTICLE INFO

Keywords:
Trajectory generation
Synthetic data
Human mobility simulation
Geo AI

ABSTRACT

Trajectory Generation (TG) enables realistic simulation of individual movements for applications such as urban management, transportation planning, epidemic control, and privacy-preserving mobility analysis. However, existing TG methods, particularly unconditional diffusion models, struggle with spatiotemporal fidelity as they often overlook some travel patterns that are critical in an individual's mobility behavior, such as recurrent location visits, movement scope, and temporal regularities. In this work, we propose the Autoregressive Diffusion Model for Travel-Pattern Aware Trajectory Generation (**Traveller**), a novel approach that integrates autoregressive travel-pattern modeling (AR-TempPlan) with diffusion-based trajectory generation (TravCond-Diff) to produce realistic and context-aware movement patterns. By leveraging the spatial anchor and temporal modes of visiting different locations, we derive an individual's particular travel pattern as spatiotemporal constraints for guided trajectory generation. Building on this, AR-TempPlan generates a mask location sequence as the temporal modes, planning location transitions over time, while TravCond-Diff leverages this planning signal and home location, the spatial anchor, to guide spatial generation through a discrete diffusion process. Experiments on real-world datasets demonstrate that Traveller with the dual guidance mechanism enables the production of high-fidelity and individual trajectories that effectively capture complex human mobility behaviors while preserving privacy. The code and data are available at https://github.com/YuxiaoLuo0013/Traveller.

1. Introduction

As central nodes of human society, cities have undergone profound transformations propelled by cultural, economic, political, and technological progress [1–4]. With the acceleration of urbanization, cities are expanding in scale and growing increasingly complex in function, presenting significant challenges for effective analysis, planning, and governance [5–7]. To meet these challenges, Urban Computing has emerged as a promising paradigm that harnesses sensing technologies and large-scale computing infrastructures to derive actionable insights from the massive data streams generated across urban environments [8]. A core component of Urban Computing is the integration of heterogeneous and large-scale datasets from multiple domains, also known as Cross-Domain Data Fusion [9]. These datasets include environmental sensor readings, infrastructure usage, human mobility data, among others [10–12]. Among these, human mobility data, particularly fine-grained individual trajectories that describe detailed daily movements such as commuting from home to work and stopping at specific locations, are especially valuable for capturing the dynamic interplay between people and urban spaces. However, acquiring such high-quality, large-scale mobility data remains costly and is often constrained by privacy regulations and policy limitations [13,14]. To address the scarcity of high-quality individual-level mobility data, Trajectory Generation (TG), a synthetic data technology, offers a solution by acting as both a data generator and a mobility simulation engine, enabling research and applications in extensive areas such as digital twin simulations [15], crime risk assessment [16], transportation optimization [17], and epidemic control [18], while ensuring privacy through synthetic data [19,20].

Building on their success in image and video generation [21–23], generative models have also shown significant promise in TG. GAN-based approaches, such as SeqGAN [24], have been applied to simulate spatiotemporal trajectories, leveraging their ability to model sequen-

* Corresponding author.
E-mail addresses: yuxiao.luo@connect.polyu.hk (Y. Luo), sm.zhang1@siat.ac.cn (S. Zhang), kangliu@siat.ac.cn (K. Liu), yang.ls.xu@polyu.edu.hk (Y. Xu), yinling@siat.ac.cn (L. Yin).

https://doi.org/10.1016/j.inffus.2025.103766
Received 17 April 2025; Received in revised form 12 July 2025; Accepted 18 September 2025
Available online 29 September 2025

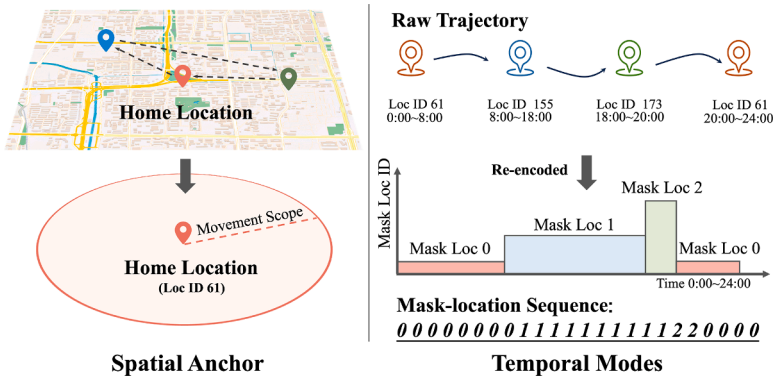

Fig. 1. Illustration of the proposed travel pattern, incorporating the home location as a spatial anchor and the re-encoded mask-location sequence as temporal modes.

tial dependencies [19,25]. Similarly, diffusion models [26,27] have emerged as powerful tools for generating mobility trajectories, demonstrating their effectiveness in accurately capturing complex spatial movement patterns [28–30]. These models leverage powerful deep learning architectures to capture complex spatiotemporal dependencies, offering significant potential for human mobility modeling. Despite their potential, these approaches still face challenges in aligning with the real-world travel behaviors of urban residents.

First, unconditional generation methods like TraGDM [28] generated trajectories as the overall distribution without incorporating some important individual travel patterns, such as movement constraints and temporal regularities, leading to spatial and temporal inconsistencies. **Second**, some methods [31,32] attempt to enhance trajectory generation by incorporating trajectory semantics, which can be categorized into exact activity types (e.g., staying at home, working, shopping, etc.) or inferred basic activity sequences such as home-work-other patterns. For instance, ActSTD [31] jointly generated spatiotemporal trajectories and exact activity labels. However, large-scale individual mobility datasets such as mobile phone positioning data typically contain only raw spatiotemporal information (e.g., coordinates and timestamps) without semantic annotations indicating the activities conducted at each location. In this context, relying on exact activity types often requires extensive manual labeling (e.g., from travel surveys), which is both labor-intensive and time-consuming, making it difficult to scale and update in real time [3]. Alternatively, Act2Loc [32] inferred a simplified activity sequence to guide spatial selection, but still relied on real data during the generation process, which raises potential privacy concerns.

To address these challenges, we propose the Autoregressive Diffusion Model for Travel-Pattern Aware Trajectory Generation, called **Traveller**, a novel TG framework specifically designed to align synthetic trajectories with real-world travel patterns. In this framework, as shown in Fig. 1, we first derive a particular travel pattern of individual trajectories from the trajectory data itself without requiring external semantic labels, represented by the home location and the re-encoded mask location sequence. These two components serve as the spatial anchor and the temporal modes. Based on the defined travel pattern, combining the advantage of the Autoregressive model (AR) in capturing temporal dynamics with the diffusion model's strength in spatial modeling, Traveller utilizes a travel-pattern-guide generation. Specifically, the Autoregressive Temporal Planning Model (**AR-TempPlan**) generates temporal modes as conditions, planning location transitions over time to reflect real individual travel behaviors. Meanwhile, the Travel-Pattern Conditioned Diffusion Model (**TravCond-Diff**) guided by the temporal condition and utilizing the home location as a spatial anchor, excels in spatial modeling through a spatial discrete choice diffusion process. This process is denoised by Travel-Pattern Diffusion Transformer (TravDiT) Block with MeanClamp operation to align with spatiotemporal conditions. This dual-guidance mechanism ensures that the generated trajectories not only respect the inherent temporal order and align with the spatial distribution of real-world mobility patterns but also preserve privacy by relying entirely on patterns learned from generative models, without accessing sensitive individual-level data in the generation process. In summary, our key contributions can be summarized as follows:

- We propose **Traveller**, the first trajectory generation framework that integrates autoregressive planning and diffusion-based spatial modeling to effectively capture spatiotemporal dynamics in human mobility.
- We define an effective travel pattern at an individual level, and introduce a Travel-Pattern Diffusion Transformer (TravDiT) Block that leverages the home location as a spatial anchor and integrates temporal modes, ensuring alignment with real-world individual mobility patterns.
- Extensive experiments on real-world datasets demonstrate that our method achieves state-of-the-art performance in terms of spatiotemporal fidelity and comparable utility to real data in downstream tasks such as next-location prediction.

2. Related work

Synthetic trajectory generation. Current studies on trajectory generation tasks can generally be divided into mechanism-driven and deep-learning (DL) approaches. Mechanism-driven models [33,34] focus on capturing the statistical patterns of human mobility using a compact set of parameters that have physical relevance. For example, the Exploration and Preferential Return (EPR) model [35] simulated movement by balancing exploration and revisitation behaviors. According to the EPR model, Barbosa et al. [33] and Alessandretti et al. [36] introduced more intricate individual behavior mechanisms, social mechanisms, and geospatial features. However, these methods rely on fixed assumptions, limiting their ability to model the complexity of human mobility. In addition, DITRAS [37] combined a Markov-based diary generator with an EPR-based trajectory generator. However, it relied on a binary indicator (i.e., staying at home or not) to generate activity diaries, ignoring other activities such as work or other, which limits its ability to comprehensively model travel behavior in the temporal domain.

Recent research has primarily concentrated on advanced DL methods like Generative Adversarial Networks (GANs) [19,38], and Variational Autoencoders (VAEs) [39,40] for complex spatiotemporal trajectory modeling. For example, MoveSim [38] applied SeqGAN [24] to capture mobility patterns, while VOLUNTEER [40] modeled user behavior and mobility actions using a two-layered VAE. Inspired by the success of diffusion models in image generation [21,27,41], researchers have introduced them to trajectory synthesis. DiffTraj [29] pioneered diffusion-based trajectory generation, while TrajGDM [28] employed cross-entropy loss to train a continuous diffusion model for discrete trajectories. Recent works have explored control mechanisms in trajectory generation. For example, ControlTraj [30] and Diff-RNTraj [42]

use road networks as spatial constraints, which are suitable for vehicle trajectories that emphasize road segments and driving dynamics. In contrast, human mobility focuses on spatiotemporal activity patterns such as stays, trips, and activity chains. As a result, vehicle-based constraint methods cannot be directly applied to human mobility, which often relies on coarser data sources like cellular base stations and different modeling objectives. Other studies incorporate user attributes (e.g., age, occupation) for controllable generation [43], but such profile data is often sensitive and difficult to obtain due to privacy concerns. Furthermore, these diffusion models often lack explicit incorporation of individual-level travel patterns such as recurrent visits and temporal dynamics, resulting in reduced spatial and temporal coherence in the generated trajectories.

Diffusion model with autoregressive transformation. Diffusion models have recently emerged as a powerful class of generative models, achieving state-of-the-art results in various domains such as image synthesis [26,27,41], text generation [44–46], and time series forecasting [47,48]. These models generate data through iterative denoising, capturing complex distributions with high fidelity. However, sequential tasks like video or text generation require additional modeling to ensure temporal consistency and motion dynamics. To address this, recent works have integrated Autoregressive (AR) transformers with diffusion models, leveraging AR's sequential modeling capability to improve temporal coherence. For video generation, MarDini [49] combined masked auto-regression with diffusion for temporal consistency, while Diffusion Forcing [50] trained causal sequence models to denoise variable-length sequences with adaptive noise levels. In the domain of text generation, AR-Diffusion [51] introduced a left-to-right noise schedule to ensure that each token is generated conditioned on previously generated tokens. Block Diffusion [52] interpolated between discrete denoising diffusion and autoregressive models, achieving state-of-the-art performance among diffusion-based methods on language modeling benchmarks and enabling the generation of sequences of arbitrary length. Inspired by the diffusion model with autoregressive transformation, we introduce AR-TempPlan in Traveller to model temporal planning, guiding TravCond-Diff for realistic trajectory generation. This design enables more realistic and behaviorally consistent trajectory generation by aligning spatial transitions with plausible temporal dynamics observed in human mobility patterns.

3. Preliminary

3.1. Problem statement

Definition 1 (Individual trajectory)**.** The individual trajectory of a person is formally defined as a sequence of K stay points $\mathbf{s} = [l_1, l_2, \dots, l_K]$ recorded from the equal-interval time slot (e.g., a 1-hour time slot), where each record l_k denotes the location identifier of the k-th stay point selected from a set of possible locations $\mathbb{L}$ (e.g., grid ID, community ID, and POI ID).

Definition 2 (Travel pattern)**.** The travel pattern of an individual defines some particular spatial and temporal conditions shaping her/his movement behavior. It consists of two key components:

- Spatial anchor: The home location l_h is used as the spatial anchor, constraining an individual's movement scope and providing the recurrent visiting point for spatial consistency.
- Temporal modes: A re-encoded mask-location sequence $\mathbf{Z}$, which plans temporal travel modes by capturing transitions of staying locations. As illustrated in Fig. 1, each unique location in a trajectory is assigned a mask ID, representing the order of visits rather than the location's unique identifier. Visits to the same location share the same mask ID, preserving the temporal structure of the trajectory. Note that the order of visits is not a unique identifier of the visited locations.

Problem statement (Trajectory generation). The primary objective of this research is to develop a travel pattern-aware trajectory generative model G capable of generating synthetic individual trajectories aligned with real-world movement patterns. Formally, the problem is defined as follows: Given a real-world individual trajectory dataset for training, $\mathbf{S} = [\mathbf{s}^1, \mathbf{s}^2, \dots, \mathbf{s}^n]$, the goal of G is to generate a set of realistic individual trajectories $\hat{\mathbf{S}} = [\hat{\mathbf{s}}^1, \hat{\mathbf{s}}^2, \dots, \hat{\mathbf{s}}^m]$, where each $\hat{\mathbf{s}} = [\hat{l}_1, \hat{l}_2, \dots, \hat{l}_K]$ is a sequence of individual trajectory.

3.2. Diffusion model for trajectory generation

Diffusion models are a type of generative model renowned for their efficacy in generating high-quality, realistic samples. Typically, these models have two key phases: the forward phase and the reverse denoising phase.

Forward process. The forward diffusion process progressively applies a Gaussian perturbation to the original samples. Suppose we have an original vectorized trajectory $\mathbf{X}_0$, the forward noising can be defined as the following Markov Chains that aim to transition the data to a noisy vectorized trajectory, denoted as $\mathbf{X}_T$:

$$q(\mathbf{X}_{1:T} \mid \mathbf{X}_0) = \prod_{t=1}^{T} q\left(\mathbf{X}_t \mid \mathbf{X}_{t-1}\right), \tag{1}$$

$$q(\mathbf{X}_t \mid \mathbf{X}_{t-1}) = \mathcal{N}\left(\mathbf{X}_t; \sqrt{(1-\beta_t)}\mathbf{X}_{t-1}, \beta_t \mathrm{I}\right). \tag{2}$$

where $\beta \in (0,1)$ represents the noise schedule, and I represents the identity matrix. To facilitate gradient-based optimization, we can compute $\mathbf{X}_t$ by utilizing the reparameterization technique such that $\mathbf{X}_t = \sqrt{\bar{\alpha}_t}\mathbf{X}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon$, and $\bar{\alpha}_t = \prod_{t=1}^{T}(1-\beta_t)$ where $\epsilon \sim \mathcal{N}(0, \mathrm{I})$. At the final step of the forward diffusion process, the noise-dominated $\mathbf{X}_T$ conforms to a standard Gaussian distribution.

Reverse process. The reverse process of our diffusion model is to reconstruct $\mathbf{X}_0$ from $\mathbf{X}_T$ based on step-by-step denoising and conditional information $\mathbf{c}$. In the context of TG, this conditional information $\mathbf{c}$ is expressed through individuals' travel patterns. Formally, the reverse denoising process is defined by:

$$p_\theta(\hat{\mathbf{X}}_{0:T-1} \mid \hat{\mathbf{X}}_T, \mathbf{c}) = \prod_{t=1}^{T} p_\theta\left(\hat{\mathbf{X}}_t \mid \hat{\mathbf{X}}_{t-1}, \mathbf{c}\right), \tag{3}$$

$$p_\theta(\hat{\mathbf{X}}_{t-1} \mid \hat{\mathbf{X}}_t, \mathbf{c}) : = \mathcal{N}\left(\hat{\mathbf{X}}_{t-1}; \mu_\theta(\hat{\mathbf{X}}_t, t, \mathbf{c}), \sum_\theta(\hat{\mathbf{X}}_t, t, \mathbf{c})\right), \tag{4}$$

where $\mu_\theta(\hat{\mathbf{X}}_t, t, \mathbf{c})$ and $\sum_\theta(\hat{\mathbf{X}}_t, t, \mathbf{c})$ are the learned mean and variance at each step t, respectively. For example, μ_θ and $\sum_\theta$ are typically obtained by the architecture of U-Net or Transformer. Following the complete denoising process, we can directly generate a high-quality $\hat{\mathbf{X}}_0$ in the continuous latent space from Gaussian noise by T denoising steps. Diffusion models have been utilized for trajectory data generation; however, prior approaches have fundamentally overlooked the incorporation of individual travel patterns.

4. Methodology

In this section, we provide an overview of the design of Traveller, an autoregressive diffusion model. As shown in Fig. 2, Traveller consists of two components: the Autoregressive Temporal Planning Model (AR-TempPlan) and the Travel-Pattern Conditioned Diffusion Model (TravCond-Diff). The details are described in the following subsections.

4.1. Autoregressive temporal modes generation for temporal planning

Given its success in NLP [53,54] and video generation [55], the autoregressive model (AR) excels in sequential modeling by generating step-by-step outputs while preserving temporal dependencies and coherence. Therefore, we propose the Autoregressive Temporal Planning

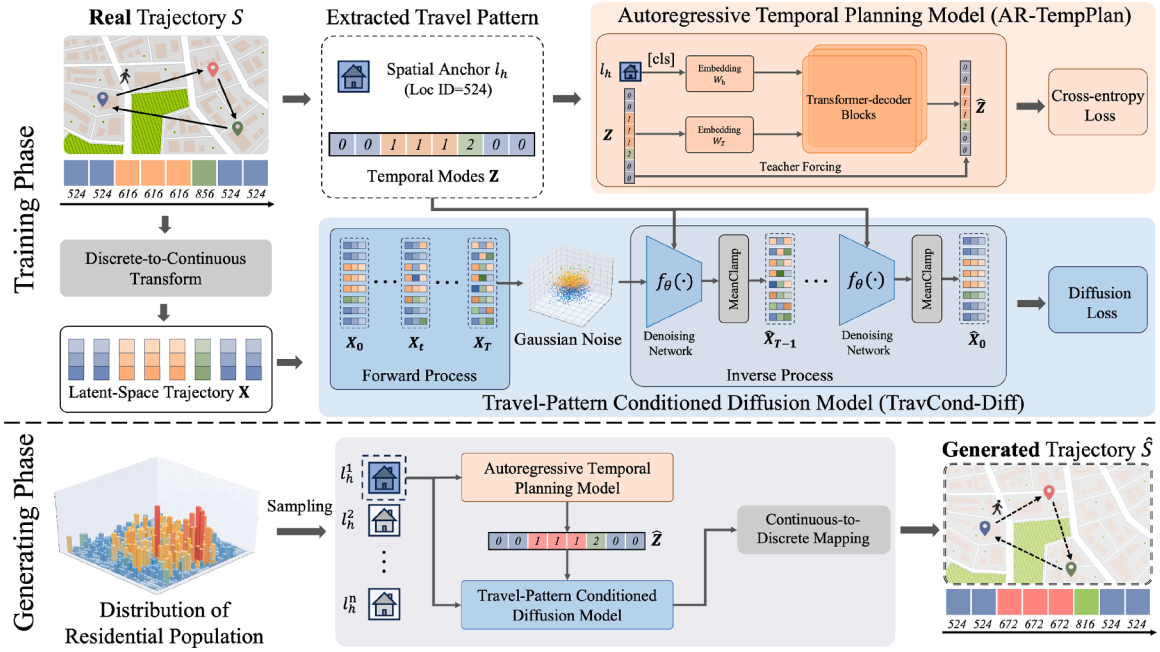

Fig. 2. Overview of the Traveller framework, divided into training and generating phases. Spatial anchors (home location) and temporal modes are derived from the raw dataset to train AR-TempPlan and TravCond-Diff. During generation, the sampled home location from the population distribution guides AR-TempPlan in planning temporal modes. These planning signals, along with the spatial anchor, then condition TravCond-Diff to generate spatiotemporal trajectories.

Model (AR-TempPlan), which formulates the trajectory temporal planning as a next-token prediction task. Specifically, AR-TempPlan leverages the home location as a context constraint and generates temporal modes as a planning signal that guides the conditions of travel patterns.

Given an individual trajectory $\mathbf{s}$ with K stay points, the home location $l_h \in \mathbb{L}$ is first identified as the location with the longest nighttime stay following the rule-based method [34]. The home location here anchors individual activities and provides a key initial condition for AR-TempPlan to iteratively generate sequences. Also, we represent $\mathbf{s}$ as a one-hot encoded sequence $\mathbf{Z} \in \{0,1\}^{K\times D}$, where K is the trajectory length and D is the total number of unique locations visited. This transformation serves as a planning signal that provides essential guidance for downstream tasks to produce realistic spatiotemporal trajectories that align with the observed temporal patterns of movements.

By defining temporal modes $\mathbf{Z} = [z_1, z_2 \ldots z_K]$ and home location l_h, we use the autoregressive objective to maximize the following likelihood:

$$\mathcal{L}(\mathbf{Z} \mid l_h) = \sum_{i=1}^{t} \log P(z_i \mid z_{1:i-1}, l_h; \phi), \tag{5}$$

where the conditional probability P is modeled using a neural network with parameters ϕ. The conditional probability P in AR-TempPlan is modeled using a Transformer decoder [56], which is well-suited for sequence modeling due to its ability to capture long-range dependencies within data. This decoder utilizes a multi-headed self-attention mechanism on the input context tokens, which is then followed by position-wise feedforward layers to generate an output distribution across target tokens. Then, we designate the home location l_h as the classification token (`cls_token`), which is concatenated with $\mathbf{Z}$ to form the initial input $\mathbf{H}_0$ for the Transformer Decoder. The subsequent hidden states $\mathbf{H}_i, \forall i \in [1, M]$ are updated iteratively using decoder layers, allowing the model to capture sequential dependencies. The process is formalized as follows:

$$\mathbf{H}_0 = \text{Concat}(\mathbf{W}_h \cdot l_h, \mathbf{W}_z \cdot \mathbf{Z}) + \mathbf{W}_{pos} + W_{time}, \tag{6}$$

$$\mathbf{H}_i = \text{Transformer-decoder Block}(\mathbf{H}_{i-1}), \forall i \in [1, M], \tag{7}$$

where $\mathbf{W}_h$ and $\mathbf{W}_z$ are the embedding matrices for the home location l_h and temporal modes $\mathbf{Z}$, respectively. $\mathbf{W}_{pos}$ is the position embedding matrix of the transformer model, and $\mathbf{W}_{time}$ encodes temporal features such as hour-of-day or day-of-week for considering explicit temporal semantics. After passing through N Decoder Layers, the probability distribution over the predicted sequence is computed as $P(\hat{\mathbf{Z}}) = \text{Softmax}(\mathbf{H}_M)$. Finally, the predicted sequence $\hat{\mathbf{Z}}$ is obtained by sampling from $P(\hat{\mathbf{Z}})$, ensuring diversity in planning the temporal patterns of individual movements.

4.2. Travel-pattern conditioned diffusion for trajectory generation

While the Transformer architecture has demonstrated exceptional performance across a range of diffusion models [57,58], it still lacks the awareness of spatiotemporal context specific to individual trajectory data. To address this, we propose the Travel-Pattern Conditioned Diffusion Model (TravCond-Diff).

Discrete-to-continuous trajectory latent spaces. As mentioned in Definition 1, individual trajectories are depicted in a discrete form by using distinct space units, making it challenging to directly employ continuous diffusion models [59] in this context. To address this, we transform the distinct location IDs of the raw trajectory $\mathbf{s}$ into continuous latent spaces $\mathbf{X}$ using an embedding matrix $\mathbf{W}_s$. This transformation is formally defined as $\mathbf{X} = \mathbf{W}_s \cdot \mathbf{s}$, where $\mathbf{W}_s \in \mathbb{R}^{D\times C}$ denotes the pretrained embedding matrix. Notably, we here use a BERT-like pretraining framework [60] to obtain $\mathbf{W}_s$ rather than employing an end-to-end training approach [44,45]. This is because our preliminary experiments indicate that pre-trained embeddings yield superior performance compared to end-to-end training, making it a more effective choice for this task. In particular, we treat the discrete trajectory points as a sequence of tokens and employ Transformer encoders to pre-train the embeddings using a masked token objective. In this approach, the model learns to reconstruct the sequence by predicting the masked tokens based on the contextual information from the surrounding trajectory points, and the learned embedding weights $\mathbf{W}_s$ are stored to represent the embedding of each distinct location in the trajectory.

Travel-pattern diffusion transformer block. Building on the success of the Transformer family, TravCond-Diff incorporates TravDiT Block, a

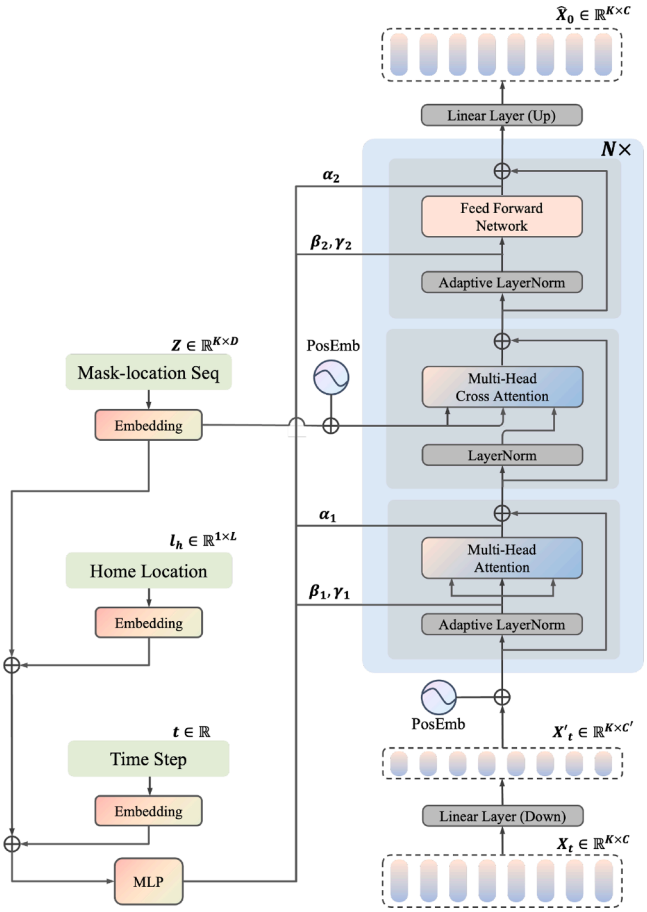

Fig. 3. The pipeline of Travel-Pattern Diffusion Transformer (TravDiT) Block.

customized module inspired by the Diffusion Transformer (DiT) [41], as its backbone architecture. As shown Fig. 3, we first transform the condition $\mathbf{c}$ in Eq. 3 into the following travel-pattern conditions: $l_h, \mathbf{Z}$, and the noise timesteps t into embedding space represented as $\mathbf{l}'_h$,$\mathbf{Z}'$, and $\mathbf{t}'$, respectively. Considering the efficiency of generation, we apply a linear projection to reduce the dimensionality of noisy trajectory embedding $\mathbf{X}_t \in \mathbb{R}^{K\times C}$ to $\mathbf{X}_t^{low} \in \mathbb{R}^{K\times C'}$. This compression helps to reduce computational overhead while preserving essential conditional information for guiding the reverse diffusion process. Position embeddings are incorporated into the inputs $\mathbf{X}_t^{low}$ and $\mathbf{Z}'$.

Then, in each TravDiT Block, the Adaptive Normalization Layer (AdaLN) [61] is utilized to acquire these condition embeddings as guidance during the denoising process. Specifically, we directly add the embedded three travel-pattern conditions, $\mathbf{l}'_h$,$\mathbf{Z}'$, and $\mathbf{t}'$ to the model, and then use a multi-layer perceptron (MLP) to regress the module parameters in TravDiT Block:

$$\gamma_1, \beta_1, \alpha_1, \gamma_2, \beta_2, \alpha_2 = \mathrm{MLP}(\mathbf{l}'_h + \mathbf{Z}' + \mathbf{t}'), \tag{8}$$

where γ_1, γ_2 and β_1, β_2 are the scale and shift factors in the AdaLN layer, while α_1, α_2 serve as dynamic scaling parameters for the TravDiT Block's output. Subsequently, the normalized output from AdaLN is refined through a Multi-Head Self-Attention (MHSA) mechanism to capture complex dependencies and enhance the representation. Regarding the input $\mathbf{X}_t^{low}$, noise is added to $\mathbf{X}$ based on the t steps. This process can be formalized as:

$$\mathbf{X}'_t = \alpha_1 \cdot \mathrm{MHSA}(\mathrm{AdaLN}(\mathbf{X}_t^{low}; \gamma_1, \beta_1)) + \mathbf{X}_t^{low}. \tag{9}$$

To further enhance temporal alignment and control, a Multi-Head Cross-Attention (MHCA) mechanism is incorporated into the TravDiT Block, positioned between the MHSA layer and the feed-forward network (FFN). The MHCA module interacts with the condition embedding $\mathbf{Z}'$ after applying layer normalization (LN), enabling the effective integration of temporal dynamics into the representation. The combined process can be formalized as:

$$\mathbf{X}''_t = \mathrm{MHCA}(\mathrm{LN}(\mathbf{X}'_t), \mathbf{Z}', \mathbf{Z}') + \mathbf{X}'_t, \tag{10}$$

where $\mathbf{X}''_t$ is the output of MHCA mechanism, incorporating temporal dynamics from $\mathbf{Z}'$ and the residual connection with $\mathbf{X}_t$. The scale and shift parameters γ_2 and β_2 dynamically modulate $\mathbf{X}''_t$ via AdaLN layer, while α_2 scales the FFN output. The residual connection within FFN step ensures stability and preserves essential information, leading to the final refined representation as follows:

$$\hat{\mathbf{X}}_0^{low} = \alpha_2 \cdot \mathrm{FFN}(\mathrm{AdaLN}(\mathbf{X}''_t; \gamma_2, \beta_2)) + \mathbf{X}''_t. \tag{11}$$

Finally, the compressed latent trajectory representation $\hat{\mathbf{X}}_0^{low} \in \mathbb{R}^{K\times C'}$ is further mapped back to the original embedding space $\mathbb{R}^{K\times C}$ via a learnable up-linear projection. This design enables the TravDiT Block to effectively and efficiently guide denoising according to the travel-pattern conditions, resulting in the generation of high-quality, context-aware trajectories.

MeanClamp operation to reduce rounding errors. Ideally, the predicted $\hat{\mathbf{X}}_0$ should be generated conditionally based on the temporal modes $\mathbf{Z}$, ensuring alignment with the corresponding location. However, we find it challenging to achieve alignment with the temporal condition when directly using $\hat{\mathbf{X}}_0$ generated by $f_\theta(\mathbf{X}_t, t, l_h, \mathbf{Z})$ to select the final discrete location. The reason for this phenomenon is that the random noise of diffusion causes rounding errors when converting the predicted location embedding $\hat{\mathbf{X}}$ back into a specific location ID. To reduce rounding errors in predicted trajectories, we introduce the MeanClamp operation into the denoising process. MeanClamp is designed to ensure alignment with the temporal condition $\mathbf{Z}$ by enforcing consistency in the final predicted locations. Specifically, for any two points i and j in the trajectory where $\mathbf{Z}_i = \mathbf{Z}_j$, the predicted embeddings $\hat{\mathbf{X}}_{0,i}$ and $\hat{\mathbf{X}}_{0,j}$ are averaged, producing a unified embedding for the corresponding location. This mechanism effectively mitigates inconsistencies caused by random noise and enhances predicted performance.

4.3. Consistent trajectory generation

Generating trajectories with high spatiotemporal consistency that align seamlessly with individual travel patterns is crucial for accurate and realistic trajectory modeling. Built on latent-space trajectory embedding and the proposed TravDiT block, our diffusion model allows for consistent generation of individual trajectories based on the process introduced in Sec. 3.2.

Training process. In the training process, our framework first utilizes derived travel-pattern constraints l_h and $\mathbf{Z}$ from the real data as guidelines. Then following Li et al. [45], we enforce $\hat{\mathbf{X}}_0 = f_\theta(\mathbf{X}_t, t, l_h, \mathbf{Z})$ to explicitly model $\mathbf{X}_0$ at each diffusion timestep, ensuring that $\mathbf{X}_0$ quickly learns to precisely center around the corresponding location embedding. With MeanClamp operation, the training objective of TravCond-Diff can be formalized as:

$$\mathcal{L}(\theta) = \sum_{t=1}^{T} \mathbb{E}_{\mathbf{X}_t} \left\| \texttt{MeanClamp}(\hat{\mathbf{X}}_0, \mathbf{Z}) - \mathbf{X}_0 \right\|^2. \tag{12}$$

Generation process. In the generation process, the model generates $\mathbf{X}_{t-1}$ by conditioning on both the current state $\mathbf{X}_t$ and the adjusted embeddings from $\texttt{MeanClamp}(\hat{\mathbf{X}}_0, \mathbf{Z})$, following the conditional distribution $q(\mathbf{X}_{t-1} \mid \mathbf{X}_t, \texttt{MeanClamp}(\hat{\mathbf{X}}_0, \mathbf{Z}))$. This averaging step enforces alignment among points with identical mask-location IDs, effectively reducing the impact of noise introduced during the diffusion process and mitigating rounding errors when mapping $\hat{\mathbf{X}}_0$ back to discrete location IDs. To further accelerate the generation process, we adopt the Denoising Diffusion Implicit Model (DDIM) [27], which enables deterministic and fewer-step

Table 1
Basic information regarding the trajectories of the two datasets.

Datasets	Beijing	Shenzhen
Source	Tencent	China Unicom
Duration	Oct 1, 2019 - No. 31, 2019	No. 1, 2021 - No. 7, 2021
#Users	100,000	200,000
#Trajectories (Short Term)	872,508	600,000
#Trajectories (Long Term)	290,836	200,000
#Girds	900	2236

sampling compared to the standard stochastic diffusion process. At $t = 0$, the final discrete location IDs are determined by calculating the cosine similarity between the denoised representation $\hat{\mathbf{X}}$ and the location embedding matrix $\mathbf{W}_s$. The location IDs with the highest cosine similarity score are selected as the final generated trajectory $\hat{\mathbf{s}}$.

5. Experiments

5.1. Experiment settings

Datasets. To comprehensively evaluate the performance of our model, we conduct our experiments using two real-world mobility datasets from Beijing and Shenzhen, two megacities in China with distinct urban forms and mobility dynamics. Beijing features a concentric urban structure and population fluctuation with relatively stable patterns, while Shenzhen exhibits a polycentric urban structure organized along an east-west development axis, forming a belt-like spatial layout, with more heterogeneous and multi-scalar mobility behavior [62]. This selection allows our model to be tested across diverse urban configurations. To capture mobility behaviors at multiple temporal resolutions, we resample individual trajectories into two fixed-length temporal scales: short-term (24 hours, one day) and long-term (168 hours, one week). Importantly, the one-week setting includes both weekday and weekend patterns, allowing the model to learn and generate richer mobility dynamics that reflect real-life routines and irregularities. Furthermore, one-week trajectories can be viewed as realistic behavioral blocks, which can be easily repeated or extended to simulate longer-term mobility scenarios. This makes our framework well-suited for urban computing tasks that require continuous synthetic trajectories, such as epidemic spread simulation, urban infrastructure planning, or digital twin modeling. We split these datasets into training, validation, and test sets randomly, with a ratio of 6:1:3. Detailed information about the datasets is provided in Table 1.

- **Beijing:** This dataset was collected from Tencent, a social networking platform, and contains anonymized and de-identified information on users' movement patterns [63]. The mobility data includes 100,000 users, primarily covering the Beijing area from October 1, 2019, to December 31, 2019. Our analysis focuses on trajectories within the five-ring area of Beijing, encompassing 900 spatial units ($1km^2$ each).
- **Shenzhen:** This dataset was obtained from Smart Steps[1], a big data company affiliated with China Unicom, one of China's major telecommunications operators with extensive population coverage. The data capture anonymized movements of 200,000 individuals over one week, from November 1 , to November 7, 2021. For analysis, the city is divided into 2236 spatial units, each covering $1km^2$.

Baselines. To evaluate the spatiotemporal fidelity of trajectories generated by the proposed Traveller, we compare the performance of Traveller with eight state-of-the-art baselines for mobility generation, including two mechanistic models and six deep-learning models.

- **W-EPR**[64]. This model extends the Exploration and Preferential Return (EPR) framework by incorporating spatial heterogeneity and a rank-based distance decay function to simulate human mobility. It effectively replicates mobility patterns at both individual and collective scales, offering robust predictions across various resolutions and highlighting the integration of factors at multiple levels.
- **DITRAS**[37]. This computational model analyzes human mobility by combining an individual's mobility diary with a trajectory generator based on exploration and preferential return mechanisms. It converts mobility diaries into geographic trajectories, enabling the extraction of movement patterns and mobility behaviors from large-scale location data.
- **FC-LSTM**[65]. It is a widely used discriminative model for sequence-to-sequence tasks. It generates trajectories by first sampling a home location from the density distribution of the population, followed by a step-by-step generation process.
- **SeqGAN**[24]. This model is for sequence generation that combines GAN with reinforcement learning, using a policy gradient to train the generator. It employs two GRU networks as the generator and discriminator. Unlike traditional GANs, which generate representations from randomly sampled latent space, SeqGAN begins trajectory generation with a trainable embedding of a start token
- **Movesim**[38]. It is a GAN-based state-of-the-art method for human mobility simulation. It samples the starting point of a trajectory from the population density distribution, with the remaining trajectory generated by a modified prediction model called SeqNet. Training is guided by a discriminator using reinforcement learning.
- **TrajGDM**[28]. It is a state-of-the-art diffusion-based method for trajectory generation that employs a trajectory encoder to map trajectories into a latent space. A trajectory generator, combining Transformer and LSTM blocks, generates trajectories from Gaussian noise. Unlike continuous diffusion models, TrajGDM uses an end-to-end SoftMax Cross-Entropy loss to convert the continuous latent space into discrete locations via a trajectory decoder.
- **DiffTraj**[29]. It is a UNet-based diffusion model designed to capture spatiotemporal features for trajectory generation. Unlike other models, it generates trajectories directly in longitude and latitude format. These trajectories are then mapped to the corresponding spatial units for performance evaluation.
- **COLA** [66]. It is a Transformer-based model designed to simulate human mobility by leveraging cross-city trajectories. The model incorporates both city-specific and city-invariant modules to capture heterogeneous and shared mobility patterns, and employs a lightweight post-hoc calibration strategy to address the challenge of long-tail location distributions.

Evaluation metrics. To evaluate the quality of the generated data, we focus on whether the generated trajectories align with the statistical features of real trajectories. We follow the common practice [12,38] and use six movement metrics to assess the quality of the generated data, illustrating the spatiotemporal characteristics of a trajectory dataset.

- ***Distance*** measures the travel distance for each movement in the trajectory.
- ***Radius*** tracks the gyration radius centered on the trajectory's center.
- ***Duration*** calculates the time spent at each location.
- ***LocNum*** records the number of locations visited by each individual per day.
- ***G-rank*** indicates the visit frequencies of the top 100 locations.
- ***I-rank*** indicates the visit frequencies of the top 10 locations for an individual.

We conducted a quantitative evaluation of the difference between the generated trajectories and actual trajectories using Jensen-Shannon (JS) divergence across the six metrics mentioned above. The formula for JS divergence is presented below:

$$D_{JS}(p,q) = \frac{1}{2} D_{KL}\left(p||\frac{p+q}{2}\right) + \frac{1}{2} D_{KL}\left(q||\frac{p+q}{2}\right), \tag{13}$$

[1] http://www.smartsteps.com

Table 2
The hyperparameters of Traveller.

Parameter	Value
Trajectory Length	24 or 168
Location Embedding	512
Hidden Dim	128
Feedforward Dim	2048
Num Blocks (AR-TempPlan)	2
Num Blocks (TravCond-Diff)	4
Num Heads	8
Diffusion Steps	2000
DDIM Steps	50
beta	0.00085~0.012

where $D_{KL}(p||q)$ is KL divergence equal to $\sum_{x\in X} p(x)\log(\frac{p(x)}{q(x)})$, p and q are two distributions, respectively. The JS divergence ranges from 0 to 1, and a lower value suggests a closer resemblance between the two probability distributions, indicating a higher fidelity of the synthetic trajectory.

In addition to these statistical metrics, we also introduce a ***Diversity*** metric to evaluate the richness and variation of generated trajectories. Specifically, we compute the number of unique trajectories in both real and generated datasets, and define the diversity error as:

$$Diversity = \frac{\left|N_{gen} - N_{real}\right|}{N_{real}}, \tag{14}$$

where N_{gen} and N_{real} denote the number of unique trajectories in the real and generated datasets, respectively. A lower *Diversity* indicates better preservation of diversity, suggesting that the generative model avoids producing overly repetitive or homogeneous samples.

Implemetation details. We begin by using a pre-trained embedding layer to map discrete location IDs into continuous embedding vectors with a dimension of $C = 512$. For model architecture, we adopt a Transformer-based design. Specifically, AR-TempPlan consists of 2 Transformer blocks with a hidden dimension of 128, while TravCond-Diff is composed of 4 Transformer blocks with the same hidden dimension. For the diffusion process, we use a diffusion step with 2000 for training, and DDIM steps of 50 are applied to accelerate sampling. The noise schedule follows a squared linear scaling. All parameters for the transformer modules and the diffusion process, including the number of heads, hidden dimensions, and diffusion settings, are summarized in Table 2, carefully selected to optimize model performance while maintaining computational efficiency. Our model is trained with the Adam optimizer [67] with an initial learning rate of $5e-4$. We also tune the learning rate with the cosine decay schedule. The batch size is set to 512 for the short-term dataset and 128 for the long-term dataset due to the limitation of GPU memory usage. The default training process is 200 epochs with 30 epochs of early stopping. We evaluate the performance every 10 epochs and save the best model with the best performance on the validation as the final testing. All experiments are repeated 5 times, implemented in PyTorch [68], and conducted on a single NVIDIA RTX A5000 24GB GPU, which is sufficient for all our experiments.

5.2. Main results

Table 3 presents a comparative evaluation of the spatiotemporal fidelity of generated trajectories across all methods. Overall, Traveller consistently outperforms all baseline models in both short-term and long-term scenarios on two real-world urban mobility datasets, demonstrating its superior capability in capturing realistic human movement patterns.

The use of spatially anchored guidance in Traveller leads to substantial improvements in the spatial fidelity of the generated trajectories, particularly in metrics such as *Displacement* and *Radius*. For instance, on the Beijing dataset, Traveller achieves the lowest JS divergence for *Displacement* (0.0023) and *Radius* (0.0040), outperforming both mechanistic models like DITRAS (*Displacement*: 0.0247, *Radius*: 0.2280) and deep learning models like DiffTraj (*Displacement*: 0.0027, *Radius*: 0.0068). Moreover, benefiting from the temporal-mode guidance, Traveller demonstrates significant improvements in temporal metrics, especially under the long-term setting. For example, on the long-term Shenzhen dataset, it outperforms comparison models with notably lower JS divergence scores in *Duration* (0.0016) and *LocNum* (0.0004). Due to the dominance of home locations in the data, discriminative models such as FC-LSTM often generate trajectories that remain at the home location, resulting in better performance in the population-level metric *G-rank*. However, this behavior fails to capture the realistic diversity of individual mobility. In contrast, Traveller achieves competitive *G-rank* scores while maintaining a more faithful representation of actual travel behaviors. Furthermore, Traveller attains better results in *I-rank*, which reflects individual-level location preferences, indicating that it not only captures global spatial patterns but also preserves personalized mobility characteristics.

In addition to movement statistical alignment, we also evaluate the diversity of generated trajectories to measure the relative difference in the number of unique trajectories between the generated and real datasets. As shown in Table 3, Traveller achieves the closest diversity to that of the real data in both datasets, demonstrating its ability to generate varied and representative trajectories. This alignment is essential for ensuring that the synthetic data faithfully reflects the structural complexity of real-world human mobility.

In summary, Traveller demonstrates a strong ability to generate spatiotemporally consistent trajectories with high diversity fidelity, consistently outperforming state-of-the-art baselines across multiple metrics and datasets.

5.3. Ablation study

To assess the significance of key design elements in Traveller, we performed an ablation study on both the Beijing and Shenzhen datasets. In this study, we systematically removed the spatial anchor and the temporal modes, MeanClamp operation, and compression in TravDiT. We also assessed the performance of a baseline model using a single Transformer that directly generates trajectories without the TravCond-Diff component. Additionally, we compared different location embedding strategies, including random initialization and end-to-end training, against Traveller's pre-trained embedding approach.

The results, presented in Table 4, show a substantial decrease in performance when these elements are excluded. Specifically, removing the home location condition leads to a significant increase in JS divergence for both temporal and spatial metrics, with a notable impact on the *Radius*, which is related to individual movement scale. Similarly, eliminating the temporal modes results in a marked deterioration, particularly in *Duration*, which reflects the temporal movement pattern. Eliminating the MeanClamp operation further weakens temporal alignment, causing the generated trajectories to deviate from expected time-based behaviors. Additionally, removing the embedding compression module negatively affects performance. This is because our model aims to generate discrete location IDs, and operating in a lower-dimensional embedding space helps reduce overfitting to noise during denoising.

When compared to the autoregressive Transformer model (w/o TravCond-Diff), it generates trajectories with relatively high spatiotemporal fidelity, particularly in learning the temporal patterns. However, this model struggles to capture spatial movement patterns, such as *Radius*. In contrast, our proposed approach, which combines the strengths of both autoregressive and diffusion models with travel-pattern guid-

Table 3
Performance comparison of different trajectory generative approaches, where lower results are better. **Bold** denotes best results and underline denotes the second-best results.

Method		Beijing							Shenzhen						
		Displacement	*Radius*	*Duration*	*LocNum*	*G-rank*	*I-rank*	*Diversity*	*Displacement*	*Radius*	*Duration*	*LocNum*	*G-rank*	*I-rank*	*Diversity*
Short-term (24 Hours)	W-EPR	0.1472	0.3613	0.1688	0.4938	0.0002	0.1189	1.2174	0.1524	0.3045	0.2463	0.5418	0.0001	0.1290	1.0915
	DITRAS	0.0247	0.2280	0.0108	0.2555	0.0045	0.0181	1.2073	0.0189	0.1452	0.0135	0.2122	0.0014	0.0193	1.0643
	FC-LSTM	0.0211	0.1976	0.2347	0.1974	**0.0001**	0.0635	0.9925	0.0179	0.1993	0.1354	0.2037	0.0001	0.0614	0.9795
	SeqGAN	0.0874	0.4299	0.1219	0.3891	0.0757	0.1214	0.2934	0.0167	0.2009	0.0568	0.2090	0.0724	0.0390	0.3131
	Movesim	0.0878	0.3545	0.1244	0.2908	0.0193	0.1030	0.4022	0.0208	0.2201	0.0638	0.2580	0.0162	0.0493	0.4833
	TrajGDM	0.0438	0.2578	0.0441	0.2276	0.0060	0.0613	1.1427	0.0319	0.2368	0.0654	0.2173	0.0096	0.0629	0.8664
	DiffTraj	0.0027	0.0068	0.0114	0.0117	0.0001	0.0029	0.1279	0.0192	0.0209	0.0241	0.0244	0.0002	0.0033	0.0479
	COLA	0.0134	0.1162	0.0332	0.1168	0.0003	0.0360	0.8355	0.0125	0.1316	0.0599	0.1385	0.0002	0.0385	0.8718
	Traveller	**0.0023**	**0.0040**	**0.0006**	**0.0002**	0.0004	**0.0001**	**0.0293**	**0.0017**	**0.0030**	**0.0014**	**0.0002**	**0.0001**	**0.0003**	**0.0298**
Long-term (168 Hours)	W-EPR	0.1518	0.3899	0.1682	0.6427	0.0002	0.1262	0.4166	0.1508	0.3413	0.2508	0.6473	**0.0001**	0.1452	0.2286
	DITRAS	0.0147	0.1825	0.0143	0.2207	0.0040	0.0299	0.3996	0.0117	0.1181	0.0178	0.1552	0.0009	0.0289	0.1669
	FC-LSTM	0.0098	0.0865	0.3452	0.0267	**0.0002**	0.0186	0.9415	0.0166	0.1830	0.3021	0.1544	0.0007	0.0679	0.9644
	SeqGAN	0.0167	0.2009	0.0568	0.2090	0.0724	0.0390	0.2584	0.0148	0.1326	0.0507	0.1426	0.2246	0.0287	0.1792
	Movesim	0.0172	0.4256	0.0878	0.4505	0.0153	0.0773	0.4149	0.1101	0.4585	0.1114	0.5530	0.0197	0.1486	0.2286
	TrajGDM	0.0289	0.3339	0.0286	0.2001	0.0424	0.0622	0.3817	0.0184	0.2475	0.0262	0.0814	0.0396	0.0377	0.1262
	DiffTraj	0.0102	0.0363	0.0271	0.0976	0.0002	0.0164	0.2817	0.0278	0.0317	0.0936	0.1143	0.0003	0.0151	0.0629
	COLA	0.0164	0.2162	0.0362	0.3246	0.0003	0.0662	0.8813	0.0158	0.3140	0.0606	0.3429	0.0002	0.0658	0.8974
	Traveller	**0.0063**	**0.0214**	**0.0012**	**0.0036**	0.0007	**0.0005**	**0.0037**	**0.0074**	**0.0278**	**0.0016**	**0.0004**	0.0003	**0.0001**	**0.0560**

Table 4
Performance comparison of Traveller against ablation variants on the Beijing and Shenzhen datasets (Short-term). **Bold** denotes best results.

Method	Beijing (Short-term)						Shenzhen (Short-term)					
	Displacement	*Radius*	*Duration*	*LocNum*	*G-rank*	*I-rank*	*Displacement*	*Radius*	*Duration*	*LocNum*	*G-rank*	*I-rank*
w/o Spatial Anchor	0.0584	0.1601	0.0752	0.0337	0.0651	0.0057	0.0194	0.0750	0.0341	0.0027	0.0180	0.0006
w/o Temporal Modes	0.0358	0.0332	0.1987	0.1404	0.0009	0.0275	0.0323	0.0436	0.2982	0.1612	0.0014	0.0460
w/o MeanClamp	0.0336	0.0301	0.0851	0.0729	0.0009	0.0128	0.0272	0.0331	0.0934	0.1139	0.0013	0.0482
w/o Compression	0.0034	0.0237	0.0018	0.0010	0.0015	0.0002	0.0049	0.0274	0.0030	0.0042	0.0038	0.0005
w/o TravCond-Diff	0.0068	0.1384	0.0147	0.0094	0.0005	0.0021	0.0159	0.1279	0.0176	0.0079	0.0002	0.0022
w/ Random Embedding	0.0203	0.0887	0.0045	0.0052	0.0217	0.0007	0.0143	0.0854	0.0028	0.0013	0.0581	0.0004
w/ E2E Embedding	0.0067	0.0249	0.0060	0.0086	0.0517	0.0015	0.0059	0.0200	0.0049	0.0044	0.0018	0.1027
Traveller	**0.0023**	**0.0040**	**0.0006**	**0.0002**	**0.0004**	**0.0001**	**0.0017**	**0.0030**	**0.0014**	**0.0002**	**0.0001**	**0.0003**

ance, significantly outperforms in reconstructing both spatial and temporal human movement patterns.

Previous discrete diffusion studies, such as those in text generation [44,45], typically adopt an end-to-end training strategy to learn the transformation from discrete tokens to continuous latent spaces. However, we find that a pre-trained embedding matrix yields better performance for trajectory generation. Following the BERT pretraining framework [60], we mask 15 % of the trajectory tokens and use a 2-layer transformer to predict the masked tokens. Through this process, the embedding matrix maps discrete locations into a continuous latent space, effectively capturing trajectory semantics. To validate the effectiveness of our pretraining method, we perform an ablation study comparing it against the end-to-end framework [44,45] and Gaussian random embeddings. As depicted in Table 4, Traveller with the pre-trained embedding matrix has significantly better performance than the end-to-end method and random embeddings. This is because the pre-training process enables the model to learn robust location representations from masked trajectory data, capturing spatial and temporal dependencies more effectively.

5.4. Visualization analysis of collective patterns

To validate the alignment of our generated individual trajectories with collective human travel behaviors, we compared the major origin-destination (OD) flows at the subdistrict level across real data, our model, and baseline models. The Common Part of Commuters (CPC) metric [12] is used to measure the similarity of OD flows between real data and generated data. As shown in Fig. 4, the visualization reveals that our model closely replicates real-world OD flows, demonstrating significant spatial similarity to real trajectory data. Compared to other baselines, our model achieves a higher CPC score of 0.6996, outperforming SeqGAN (0.5857). Specifically, the core economic zones of Shenzhen are concentrated in the central and western regions, characterized by high employment and residential densities that drive major interregional flows. The trajectories generated by our model successfully capture these dominant flows, accurately reflecting the concentration of travel behaviors in these areas.

The spatial-temporal distribution of population is also an important collective pattern of human mobility. We visualize and compare the aggregated population distributions from both the real-world dataset and the trajectories generated by our model on the Beijing dataset. Specifically, we create heatmaps representing the population distributions at four representative time points: 00:00 AM, 9:00 AM, 6:00 PM, and 11:00 PM. These time points are chosen to reflect different mobility patterns, including nighttime stability, morning commute, evening peak, and late-night activity. As shown in Fig. 5, the heatmaps of our generated data exhibit overall similarity to those of the real-world dataset and reflect general patterns of activity throughout the day. At 00:00 AM, the population is concentrated in residential areas, which is accurately captured by our model. At 9:00 AM and 6:00 PM, both the real and generated heatmaps exhibit similar distributions for most of the spatial units, reflecting high population density in major business districts and transportation hubs during peak commuting hours. By nighttime, as shown in the 11:00 PM heatmaps, populations in both the real and generated data return predominantly to residential areas, demonstrating the model's ability to replicate the typical daily mobility cycle. These results demonstrate that our model effectively captures the spatiotemporal dynamics of human mobility, producing realistic population distributions at various times of

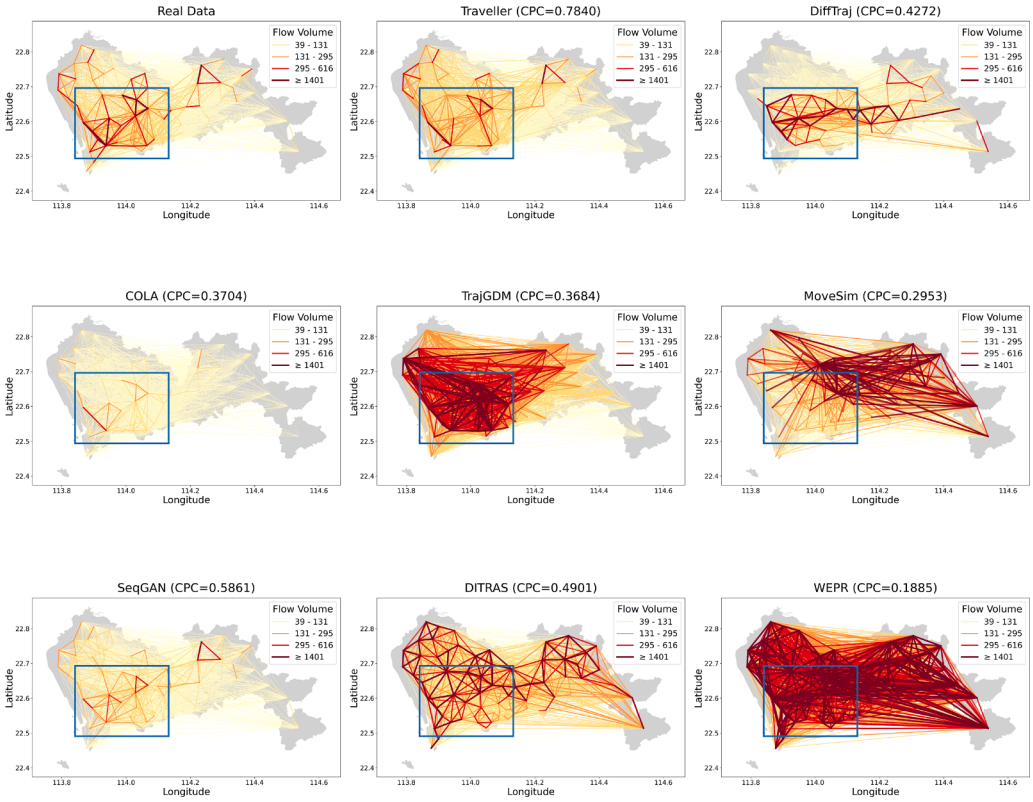

Fig. 4. Visualization of street-level origin-destination flows in Shenzhen, comparing real data with trajectories generated by Traveller and other baselines.

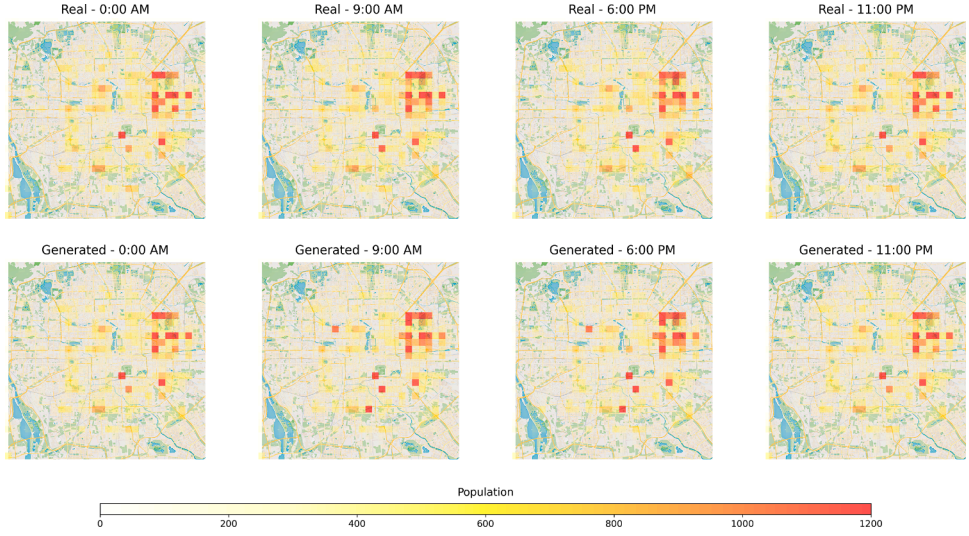

Fig. 5. Heatmaps of aggregated population distributions at four representative time points (12:00 AM, 9:00 AM, 6:00 PM, and 11:00 PM) on the Beijing dataset. The top row shows the distributions from the real dataset, while the bottom row depicts those generated by our model.

the day. The close alignment with real-world data highlights the ability of Traveller to generate high-fidelity trajectories that accurately reflect collective human movement patterns.

5.5. Visualization of individual trajectories

To effectively demonstrate the fidelity of our generated trajectories, we present four generated individual trajectories for one week of the Shenzhen dataset in Fig. 6. These examples represent three distinct movement categories: Home Stayer, Routine Commuter, and Explorer. As shown in Fig. 6, the trajectories cover one week in Shenzhen. Cases 1 and 2 fall under the Home Stayer category, as their movements are entirely concentrated around their home locations, a pattern that dominates the dataset. Case 3 represents Routine Commuters, such as students or employees, who exhibit regular travel patterns on weekdays. Lastly, Case 4 exemplifies the Explorer category, characterized by individuals with a broader and more variable range of movements. With a spatial resolution of 1 km, undetected movement does not imply individuals stayed home but rather that their activities occurred within the same spatial unit. These results highlight the ability of Traveller to effectively capture diverse travel patterns, accurately reflecting the complexity of real-world human mobility behaviors.

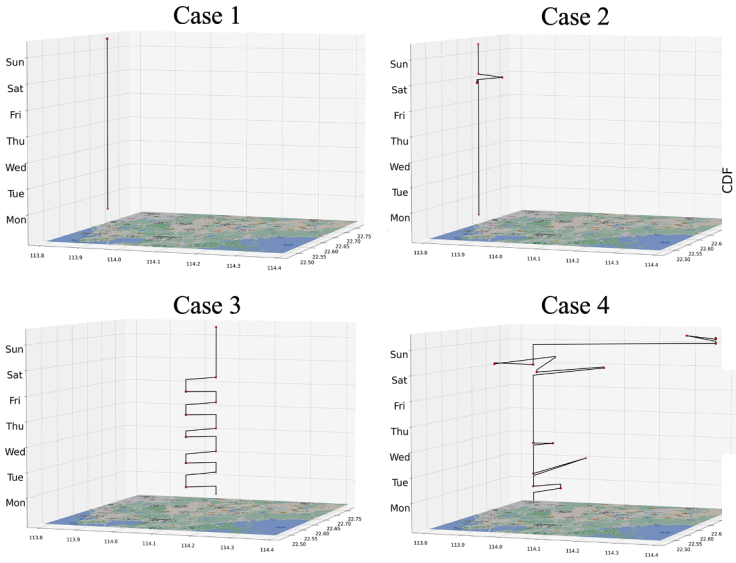

Fig. 6. Visualization of one-week individual trajectory samples generated by Traveller, showcasing different travel patterns: Home Stayer (Cases 1 and 2), Routine Commuter (Case 3), and Explorer (Case 4).

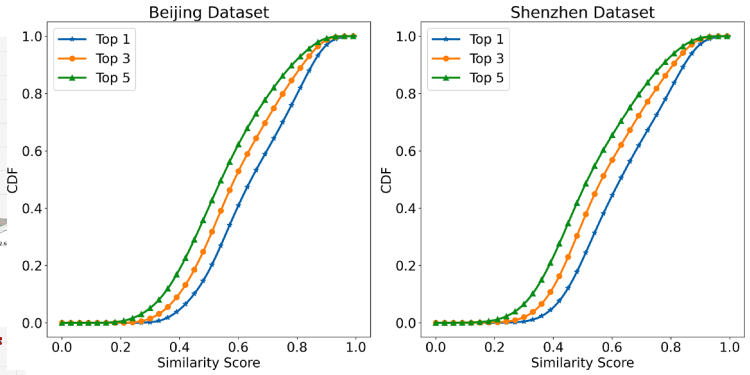

Fig. 7. Results of uniqueness testing.

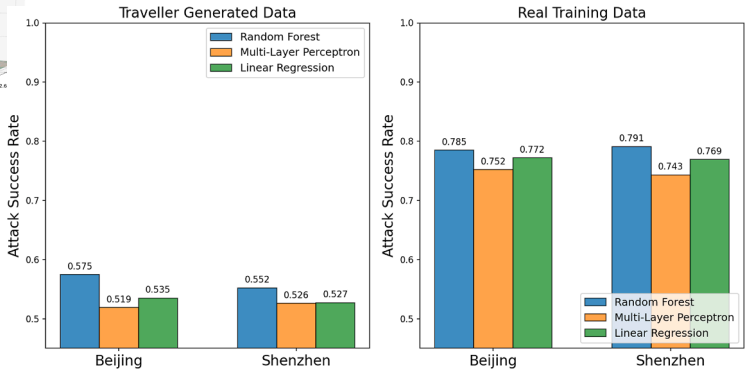

Fig. 8. Success rates of membership inference attacks using different classification algorithms. Comparison between Traveller-generated trajectories and a non-synthesis baseline using sampled real training data.

5.6. User privacy protection

The use of real trajectory data is often restricted due to privacy concerns. In many real-world scenarios, such as with telecom operators or mobility service providers, individual-level trajectory data cannot be shared externally due to regulatory and ethical constraints. Motivated by this challenge, our goal is to enable the generation of high-quality synthetic trajectories that preserve the statistical characteristics of human mobility while minimizing privacy risks to individuals, enabling two practical privacy-preserving deployment scenarios: (1) internal generation of synthetic data that can be safely shared for research or application use, and (2) training the model on sensitive data in a secure environment, after which the trained model can be exported and used for downstream simulation tasks without further access to real data. In this section, we evaluate whether synthetic trajectories could pose a risk of privacy leakage. Specifically, we focus on assessing whether individual trajectories from the training data can be identified or distinguished based on the generated outputs. Our goal is to examine whether the model memorizes and reproduces specific training samples, which would indicate a potential privacy risk at the user level. In this case, we adopt two commonly used evaluation methods in trajectory generation.

5.6.1. Uniqueness testing

Previous research commonly uses uniqueness testing to ensure that generated trajectories are distinct and do not pose privacy risks [19,69]. In this study, we randomly select generated trajectories and compare them with real trajectories from the training dataset using similarity matching. Specifically, we calculate similarity based on Levenshtein distance [70]. We then analyze the similarity distribution and display the cumulative distribution of the top 1, 3, and 5 most similar real trajectories for each generated trajectory.

If the model memorized and replicated real trajectories from the training set, the Levenshtein distance between some generated and real trajectories would be zero (i.e., an exact match), resulting in a Top-1 similarity score of 1.0. Consequently, the cumulative distribution function (CDF) of the top-k similarity scores would show a pronounced spike near 1.0, indicating memorization at the upper end of the distribution. However, as shown in Fig. 7, most generated trajectories fall within a similarity range of 0.3 to 0.8, and no generated sample achieves a perfect match (similarity $= 1$) with any real trajectory. This evaluation is effective because a perfect or near-perfect match would indicate that the model may have memorized and reproduced specific training instances, which is a major privacy concern. In contrast, the observed results suggest that while the generated data preserves movement characteristics, it does not replicate individual trajectories exactly. This implies that the model generalizes patterns from the training data rather than memorizing them, thereby reducing the risk of direct re-identification.

5.6.2. Analysis of membership inference attack

Algorithm 1: The main processes of AR-Temp.

```
/* Training Process:                                          */
```
Raw trajectories **s**; Initialize Transformer decoder with parameters ϕ;
foreach *each individual trajectory* $\mathbf{s}_i$ **do**
  Identify l_h via longest nighttime stay ;
  Convert $\mathbf{s}_i$ to mask-location sequence $\mathbf{Z}_i$;
  Compute $P(z_{1:k} \mid z_{1:k-1}, l_h; \phi)$ with Transformer decoder via Eq. (6)-(7) ;
  Update ϕ by maximizing $\mathcal{L}(\mathbf{Z} \mid l_h)$ via Eq. (5) ;

```
/* Generation Process:                                        */
```
Input: Sampled home location l_h
Output: Mask-location sequence **Z**
Initialize **Z** as empty sequence ;
Set z_0 as `cls_token` (home location l_h) ;
foreach *each timestep* $k \in [1, K]$ **do**
  Compute $P(z_k \mid z_{1:k-1}, l_h; \phi)$ via Transformer decoder ;
  Sample z_k from $P(z_k \mid z_{1:k-1}, l_h; \phi)$;
  Append z_k to **Z** ;

Table 5
The performance of mobility prediction based on generated trajectories and real data, measured by Accuracy, Exact Match Ratio (EMR), and Average Prediction Length (APL).

Method	Beijing (Short-term)						Shenzhen (Short-term)					
	ACC	ACC_{t+1}	ACC_{t+2}	ACC_{t+3}	EMR	APL	ACC	ACC_{t+1}	ACC_{t+2}	ACC_{t+3}	EMR	APL
Real	0.7168	0.8196	0.7065	0.6245	0.5906	2.1014	0.7504	0.8508	0.7388	0.6616	0.6404	2.2223
DiffTraj	0.7013	0.8099	0.6902	0.6036	0.5713	2.0566	0.5668	0.6601	0.5551	0.4851	0.4661	1.674
Traveller	0.7174	0.8125	0.7164	0.6091	0.5764	2.0913	0.7410	0.8442	0.7315	0.6513	0.6297	2.1964

The objective of this attack is to determine whether a given sample was part of the model's training dataset. Following prior studies [19,69], we adopt three commonly used classification algorithms, Random Forest (RF), Multi-Layer Perceptron (MLP), and Logistic Regression (LR), to perform membership inference attacks (MIA) [71]. This ensemble approach helps mitigate classifier-specific biases and provides a more robust evaluation of membership leakage risk Algorithms 1 and 2.

Algorithm 2: The main processes of TravCond-Diff.

```
/* Training Process:                                              */
Raw trajectories s, home locations l_h;
Get mask-location sequence Z by AR-Temp;
Pretrain embedding matrix W_s using BERT-style framework;
Transform s to latent X via X = W_s · s;
foreach each timestep t ∈ [1, T] do
    Add noise to X based on t ;
    Linear project X_t ∈ R^{K×C} to X_t^{low} ∈ R^{K×C} ;
    Compute X̂_0^{low} = f_θ(X_t^{low}, t, l_h, Z) via Eq. (8)-(11) ;
    Linear project X̂_0^{low} ∈ R^{K×C'} to X̂_0 ∈ R^{K×C} ;
    Apply MeanClamp to X̂_0 ;
    Optimize L(θ) via Eq. (12) ;
/* Generation Process:                                            */
    Input: Sampled home location distribution
    Output: Synthetic trajectory ŝ
Sample l_h from real-world distribution ;
Generate Z via AR-Temp autoregressively ;
Initialize X_T ~ N(0, I) ;
foreach t = T to 1 do
    Denoise X_t using TravDiT Block with l_h, Z ;
    Apply MeanClamp to get X_{t-1} ;
Decode X̂_0 to ŝ via cosine similarity with W_s
```

Since we do not know whether a generated trajectory originates from the member set or the non-member set, we adopt a comparative approach to construct features and labels for the classifiers. Specifically, for each sampled generated trajectory, we identify its most similar real trajectory (Top-1) in both the member set (training data) and the non-member set (unseen test data) based on Levenshtein similarity. We then extract a set of trajectory-level features between each generated trajectory and its Top-1 matches in the member set and the non-member set, including Dynamic Time Warping (DTW) distance, cosine similarity, Jaccard similarity, and differences in start and end locations. These features are used as classifier inputs, with the label assigned as positive if the Top-1 match used for feature computation comes from the member set, and negative if it comes from the non-member set. In this case, if the generated data comes from the member set, it tends to be easier for the classifier to identify correctly.

As an additional analysis, we also evaluate the attack performance on a non-synthesis baseline, in which the classifiers are trained and tested using only real data samples from the training sets, without any synthetic data involved. This allows us to assess the difficulty of membership inference in the absence of generation. Fig. 8 presents the attack success rates (AUC) across datasets and classifiers. All results remain consistently below 0.6 and close to random guessing (0.5), while the non-synthesis baseline exhibits significantly higher success rates (more than 0.75). This demonstrates that the model does not regenerate or expose training samples in a way that enables reliable identification by adversaries. Combined with the results from uniqueness testing, these findings provide empirical support that Traveller preserves a practical degree of user privacy in synthetic trajectory generation.

Table 6
Efficiency comparison of different models on the Shenzhen dataset in terms of model size, FLOPs, and inference time. (Batch Size = 512).

Method	Params (M)	Short-term		Long-term	
		FLOPs (G)	Time (s)	FLOPs (G)	Time (s)
TrajGDM	9.11	11,813.92	619	417,720.70	21,240
DiffTraj	14.44	97.82	350	666.99	900
Traveller	3.19	1,644.64	209	11,509.29	706

5.7. Utility of the generated trajectories

To assess whether Traveller's generated trajectories can substitute real trajectories in practical applications, we conduct a task of next-location prediction. Using both the Beijing and Shenzhen datasets, we train an LSTM-based trajectory prediction model on three datasets: real trajectories, those generated by Traveller, and those from DiffTraj. The trained models are then evaluated on a test dataset sampled from the real trajectories, where the model takes a historical trajectory as input and predicts the next three locations. Table 5 presents the results, measured by Accuracy, Exact Match Ratio (EMR), and Average Prediction Length (APL).

The results indicate that while the model trained on Traveller's generated data performs slightly lower than the one trained on real trajectories, the difference is minimal, suggesting that Traveller-generated data retains the essential characteristics of real-world trajectories. In contrast, the model trained on DiffTraj's generated data shows a more significant performance drop, indicating that DiffTraj-generated trajectories do not match the real data as closely in terms of utility for next-location prediction. This comparison highlights the stronger alignment of Traveller-generated trajectories with real-world movement patterns, underlining the practical value of Traveller for applications that rely on trajectory data.

5.8. Efficiency of generation analysis

Generation efficiency is critical for large-scale urban simulations, where thousands or millions of trajectories may need to be synthesized. To address this challenge, Traveller employs two key strategies to enhance generation efficiency without sacrificing output quality. First, we adopt the Denoising Diffusion Implicit Model (DDIM) [27] for trajectory sampling instead of the standard diffusion process. DDIM enables faster generation by skipping a large portion of the diffusion steps through a non-Markovian sampling mechanism, significantly reducing computation time. Second, within our TravCond-Diff module, we compress the original high-dimensional trajectory embeddings from dimension $\mathbb{R}^{K\times C}$ to a lower-dimensional space $\mathbb{R}^{K\times C'}$ before denoising. This compact representation allows the denoising network to operate in a smaller latent

space, thereby reducing computational overhead while preserving essential spatial-temporal information.

In this section, we compare the generation efficiency of Traveller with two diffusion-based baselines, TrajGDM and DiffTraj, on the Shenzhen dataset, using a batch size of 512. As shown in Table 6, Traveller exhibits a smaller model size than both baselines, reflecting a more lightweight architecture. Although DiffTraj has fewer FLOPs in theory, its reliance on a U-Net architecture limits parallelism and runtime efficiency. In contrast, Traveller leverages the highly parallelizable Transformer architecture, enabling significantly faster inference despite higher FLOPs. Specifically, Traveller completes short-term trajectory generation in 209 seconds and long-term generation in 706 seconds, while TrajGDM requires 619 and 21,240 seconds, and DiffTraj takes 350 and 900 seconds, respectively. These results demonstrate that Traveller achieves higher efficiency compared to other diffusion-based baselines.

6. Conclusion and future work

This study presents Traveller, an innovative framework for travel-pattern-aware trajectory generation that addresses the growing need for realistic and privacy-preserving human mobility data, one of the essential data sources in cross-domain fusion within urban computing. Traveller integrates two core modules: AR-TempPlan for capturing temporal regularities and TravCond-Diff for modeling spatial transitions, guided by spatial anchors and temporal modes derived from individual travel patterns. This dual-guidance mechanism enables Traveller to generate synthetic trajectories with high spatiotemporal fidelity and behavioral diversity. Comprehensive experiments on real-world datasets demonstrate that Traveller significantly outperforms state-of-the-art baselines in reconstructing fine-grained human mobility patterns, while maintaining strong utility in downstream tasks such as next-location prediction. These results highlight Traveller's potential as a reliable synthetic data generator in urban computing contexts where real mobility data are limited by privacy regulations and acquisition challenges.

However, Traveller has several limitations. First, the sampling efficiency of the diffusion-based spatial generation module remains relatively low, posing a challenge for large-scale applications. Second, like most data-driven trajectory generation models, Traveller requires city-specific training data to perform well and currently lacks generalization capability across different urban contexts. This restricts its use in cities where no real mobility data is available for training, limiting its deployment in low-data or zero-data settings. Third, the use of home location as a spatial anchor, while effective for most users, may not suit individuals with irregular mobility patterns (e.g., delivery workers).

To address these limitations, future work will focus on enhancing the efficiency of the diffusion-based component, possibly through architectural optimization or fast sampling techniques. For example, recently, some autoregressive models like the next-scale prediction framework with faster generation speed compared with diffusion models can be used in spatiotemporal modeling. We also plan to explore cross-city transfer learning and domain adaptation to improve the generalizability of Traveller across heterogeneous urban environments. One potential solution is to pretrain the temporal planning module on source cities to learn generalizable mobility patterns, then adapt it to a target city via fine-tuning or by introducing auxiliary conditions (e.g., age distribution or activity level) to generate city-specific mask-location sequences. For spatial modeling, a shared embedding space (based on location IDs or coordinates) can be constructed across cities, with OD flows serving as external guidance to adapt spatial selection to the new city without full retraining. Furthermore, to better support individuals with non-standard mobility patterns, future work will consider multi-scale spatial anchoring strategies that combine fine-grained anchors (e.g., home, frequent locations) with coarse-grained references (e.g., administrative zones or service areas), thereby improving adaptability across diverse user profiles. Finally, to further strengthen privacy protection, we aim to incorporate formal privacy-preserving mechanisms, such as differential privacy or adversarial regularization, into the training phase, enabling Traveller to provide both empirical and formal privacy guarantees when deployed in sensitive real-world scenarios.

Despite these limitations, Traveller also provides a practical solution that balances realism, privacy, and utility. It contributes to critical urban applications such as digital twin simulations, crime risk analysis, transportation optimization, and epidemic control, thereby advancing the broader goal of data-driven, sustainable urban management.

CRediT authorship contribution statement

Yuxiao Luo: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Resources, Methodology, Investigation, Formal analysis, Data curation, Conceptualization; **Songming Zhang:** Writing – review & editing, Writing – original draft, Methodology; **Kang Liu:** Supervision, Data curation, Funding acquisition; **Yang Xu:** Writing – review & editing, Supervision, Methodology, Conceptualization; **Ling Yin:** Writing – review & editing, Writing – original draft, Supervision, Methodology, Funding acquisition, Conceptualization.

Data availability

We have shared my data and code via the link in the paper.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

Ling Yin acknowledges supports from the National Natural Science Foundation of China (No. 42271475) and Shenzhen Key Basic Research Project (No. JCYJ20241202125008011). Yang Xu acknowledges supports from the National Natural Science Foundation of China (No. 42171454). Kang Liu acknowledges supports from the National Natural Science Foundation of China (No. 42571532, 42271474).

References

[1] W. Chen, G. Wang, J. Zeng, Impact of urbanization on ecosystem health in Chinese urban agglomerations, Environ. Impact Assess. Rev. 98 (2023) 106964.
[2] B. Guo, Y. Wang, H. Zhang, C. Liang, Y. Feng, F. Hu, Impact of the digital economy on high-quality urban economic development: evidence from Chinese cities, Econ. Model. 120 (2023) 106194.
[3] Y. Liu, F. Dong, How technological innovation impacts urban green economy efficiency in emerging economies: a case study of 278 Chinese cities, Resour. Conserv. Recycl. 169 (2021) 105534.
[4] J. Uitermark, C. Hochstenbach, J. Groot, Neoliberalization and urban redevelopment: the impact of public policy on multiple dimensions of spatial inequality, Urban Geogr. 45 (4) (2024) 541–564.
[5] W. Wu, Y. Lin, The impact of rapid urbanization on residential energy consumption in China, PLoS ONE 17 (7) (2022) e0270226.
[6] A. Bekkar, B. Hssina, S. Douzi, K. Douzi, Air-pollution prediction in smart city, deep learning approach, J. Big Data 8 (2021) 1–21.
[7] Y. Liang, Z. Zhao, F. Ding, Y. Tang, Z. He, Time-dependent trip generation for bike sharing planning: a multi-task memory-augmented graph neural network, Inf. Fusion 106 (2024) 102294.
[8] Y. Zheng, L. Capra, O. Wolfson, H. Yang, Urban computing: concepts, methodologies, and applications, ACM Trans. Intell. Syst. Technol. 5 (3) (2014) 1–55.
[9] X. Zou, Y. Yan, X. Hao, Y. Hu, H. Wen, E. Liu, J. Zhang, Y. Li, T. Li, Y. Zheng, et al., Deep learning for cross-domain data fusion in urban computing: taxonomy, advances, and outlook, Inf. Fus. 113 (2025) 102606.
[10] L.-M. Ang, S.K. Phooi, Big sensor data applications in urban environments, Big Data Res. 4 (2016) 1–12.
[11] D. Kasraian, K. Maat, D. Stead, B. Van Wee, Long-term impacts of transport infrastructure networks on land-use change: an international review of empirical studies, Transp. Rev. 36 (6) (2016) 772–792.
[12] M. Luca, G. Barlacchi, B. Lepri, L. Pappalardo, A survey on deep learning for human mobility, ACM Comput. Surv. 55 (1) (2021) 1–44.
[13] J. Rao, S. Gao, Y. Kang, Q. Huang, LSTM-TrajGAN: a deep learning approach to trajectory privacy protection,(2020) arXiv preprint arXiv:2006.10521.

[14] J. Zhang, Q. Huang, Y. Huang, Q. Ding, P.-W. Tsai, DP-TrajG AN: a privacy-aware trajectory generation model with differential privacy, Fut. Gen. Comput. Syst. 142 (2023) 25–40.

[15] O. Zheng, M. Abdel-Aty, L. Yue, A. Abdelraouf, Z. Wang, N. Mahmoud, City sim: a drone-based vehicle trajectory dataset for safety-oriented research and digital twins, Transp. Res. Rec. 2678 (4) (2024) 606–621.

[16] H. Zhu, F. Wang, An agent-based model for simulating urban crime with improved daily routines, Comput. Environ. Urban Syst. 89 (2021) 101680.

[17] Y. Wang, J. Cao, W. Huang, Z. Liu, T. Zheng, M. Song, Spatiotemporal gated traffic trajectory simulation with semantic-aware graph learning, Inf. Fusion 108 (2024) 102404.

[18] Y. Luo, L. Yin, K. Zhu, K. Liu, Architecting urban epidemic defense: A hierarchical region-individual control framework for optimizing large-scale individual mobility interventions, Comput. Environ. Urban Syst. 121 (2025) 102312.

[19] Z. Cao, K. Liu, X. Jin, L. Ning, L. Yin, F. Lu, STAGE: A spatiotemporal-knowledge enhanced multi-task generative adversarial network (GAN) for trajectory generation, Int. J. Geograph. Inf. Sci. (2024) 1–28.

[20] N. Savage, Synthetic data could be better than real data, Nature (2023).

[21] Y. Ma, Y. He, X. Cun, X. Wang, S. Chen, X. Li, Q. Chen, Follow your pose: pose-guided text-to-video generation using pose-free videos, in: Proceedings of the AAAI Conference on Artificial Intelligence, 38, 2024, pp. 4117–4125.

[22] S. Tulyakov, M.-Y. Liu, X. Yang, J. Kautz, Mocogan: Decomposing motion and content for video generation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 1526–1535.

[23] G. Zhu, E. Deng, Z. Qin, F. Khan, W. Wei, G. Srivastava, H. Xiong, S. Kumari, Cross-modal interaction and multi-source visual fusion for video generation in fetal cardiac screening, Inf. Fusion 111 (2024) 102510.

[24] L. Yu, W. Zhang, J. Wang, Y. Yu, Seqgan: sequence generative adversarial nets with policy gradient, in: Proceedings of the AAAI Conference on Artificial Intelligence, 31, 2017.

[25] J. Feng, Y. Li, C. Zhang, F. Sun, F. Meng, A. Guo, D. Jin, Deepmove: predicting human mobility with attentional recurrent networks, in: Proceedings of the 2018 World Wide Web Conference, 2018, pp. 1459–1468.

[26] J. Ho, A. Jain, P. Abbeel, Denoising diffusion probabilistic models, Adv. Neural Inf. Process. Syst. 33 (2020) 6840–6851.

[27] J. Song, C. Meng, S. Ermon, Denoising diffusion implicit models,(2020) arXiv preprint arXiv:2010.02502.

[28] C. Chu, H. Zhang, P. Wang, F. Lu, Simulating human mobility with a trajectory generation framework based on diffusion model, Int. J. Geograph. Inf. Sci. 38 (5) (2024) 847–878.

[29] Y. Zhu, Y. Ye, S. Zhang, X. Zhao, J. Yu, Difftraj: generating gps trajectory with diffusion probabilistic model, Adv. Neural Inf. Process. Syst. 36 (2023) 65168–65188.

[30] Y. Zhu, J.J. Yu, X. Zhao, Q. Liu, Y. Ye, W. Chen, Z. Zhang, X. Wei, Y. Liang, Controltraj: controllable trajectory generation with topology-constrained diffusion model, in: Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2024, pp. 4676–4687.

[31] Y. Yuan, J. Ding, H. Wang, D. Jin, Y. Li, Activity trajectory generation via modeling spatiotemporal dynamics, in: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2022, pp. 4752–4762.

[32] K. Liu, X. Jin, S. Cheng, S. Gao, L. Yin, F. Lu, Act2 Loc: a synthetic trajectory generation method by combining machine learning and mechanistic models, Int. J. Geograph. Inf. Sci. 38 (3) (2024) 407–431.

[33] H. Barbosa, F.B. de Lima-Neto, A. Evsukoff, R. Menezes, The effect of recency to human mobility, EPJ Data Sci. 4 (2015) 1–14.

[34] S. Jiang, Y. Yang, S. Gupta, D. Veneziano, S. Athavale, M.C. González, The time geo modeling framework for urban mobility without travel surveys, Proc. Natl. Acad. Sci. 113 (37) (2016) E5370–E5378.

[35] C. Song, T. Koren, P. Wang, A.-L. Barabási, Modelling the scaling properties of human mobility, Nat. Phys. 6 (10) (2010) 818–823.

[36] L. Alessandretti, P. Sapiezynski, V. Sekara, S. Lehmann, A. Baronchelli, Evidence for a conserved quantity in human mobility, Nat. Hum. Behav. 2 (7) (2018) 485–491.

[37] L. Pappalardo, F. Simini, Data-driven generation of spatio-temporal routines in human mobility, Data Min. Knowl. Discov. 32 (3) (2018) 787–829.

[38] J. Feng, Z. Yang, F. Xu, H. Yu, M. Wang, Y. Li, Learning to simulate human mobility, in: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2020, pp. 3426–3433.

[39] D. Huang, X. Song, Z. Fan, R. Jiang, R. Shibasaki, Y. Zhang, H. Wang, Y. Kato, A variational autoencoder based generative model of urban human mobility, in: 2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR), IEEE, 2019, pp. 425–430.

[40] Q. Long, H. Wang, T. Li, L. Huang, K. Wang, Q. Wu, G. Li, Y. Liang, L. Yu, Y. Li, Practical synthetic human trajectories generation based on variational point processes, in: Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2023, pp. 4561–4571.

[41] W. Peebles, S. Xie, Scalable diffusion models with transformers, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 4195–4205.

[42] T. Wei, Y. Lin, S. Guo, Y. Lin, Y. Huang, C. Xiang, Y. Bai, H. Wan, Diff-rntraj: a structure-aware diffusion model for road network-constrained trajectory generation, IEEE Trans. Knowl. Data Eng. (2024).

[43] Y. Song, J. Ding, J. Yuan, Q. Liao, Y. Li, Controllable human trajectory generation using profile-guided latent diffusion, ACM Trans. Knowl. Discov. Data 19 (1) (2024) 1–25.

[44] S. Gong, M. Li, J. Feng, Z. Wu, L. Kong, Diffuseq: sequence to sequence text generation with diffusion models, (2022) arXiv preprint arXiv:2210.08933.

[45] X. Li, J. Thickstun, I. Gulrajani, P.S. Liang, T.B. Hashimoto, Diffusion-lm improves controllable text generation, Adv. Neural Inf. Process. Syst. 35 (2022) 4328–4343.

[46] J. Lovelace, V. Kishore, C. Wan, E. Shekhtman, K.Q. Weinberger, Latent diffusion for language generation, Adv. Neural Inf. Process. Syst. 36 (2024).

[47] S. Feng, C. Miao, Z. Zhang, P. Zhao, Latent diffusion transformer for probabilistic time series forecasting, in: Proceedings of the AAAI Conference on Artificial Intelligence, 38, 2024, pp. 11979–11987.

[48] Y. Li, X. Lu, Y. Wang, D. Dou, Generative time series forecasting with diffusion, denoise, and disentanglement, Adv. Neural Inf. Process. Syst. 35 (2022) 23009–23022.

[49] H. Liu, S. Liu, Z. Zhou, M. Xu, Y. Xie, X. Han, J.C. Pérez, D. Liu, K. Kahatapitiya, M. Jia, et al., Mardini: masked autoregressive diffusion for video generation at scale, arXiv preprint arXiv:2410.20280 (2024).

[50] B. Chen, D.M. Monso, Y. Du, M. Simchowitz, R. Tedrake, V. Sitzmann, Diffusion forcing: next-token prediction meets full-sequence diffusion, (2024) arXiv preprint arXiv:2407.01392.

[51] T. Wu, Z. Fan, X. Liu, H.-T. Zheng, Y. Gong, J. Jiao, J. Li, J. Guo, N. Duan, W. Chen, et al., Ar-diffusion: auto-regressive diffusion model for text generation, Adv. Neural Inf. Process. Syst. 36 (2023) 39957–39974.

[52] M. Arriola, A. Gokaslan, J.T. Chiu, Z. Yang, Z. Qi, J. Han, S.S. Sahoo, V. Kuleshov, Block diffusion: interpolating between autoregressive and diffusion language models, (2025) arXiv preprint arXiv:2503.09573.

[53] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al., Language models are unsupervised multitask learners, OpenAI Blog 1 (8) (2019) 9.

[54] M. Lewis, Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension, (2019) arXiv preprint arXiv:1910.13461.

[55] L. Yu, J. Lezama, N.B. Gundavarapu, L. Versari, K. Sohn, D. Minnen, Y. Cheng, A. Gupta, X. Gu, A.G. Hauptmann, B. Gong, M.-H. Yang, I. Essa, D.A. Ross, L. Jiang, Language model beats diffusion - tokenizer is key to visual generation, in: The Twelfth International Conference on Learning Representations, 2024.

[56] P.J. Liu, M. Saleh, E. Pot, B. Goodrich, R. Sepassi, L. Kaiser, N. Shazeer, Generating wikipedia by summarizing long sequences, arXiv preprint arXiv:1801.10198 (2018).

[57] H. He, C. Bai, K. Xu, Z. Yang, W. Zhang, D. Wang, B. Zhao, X. Li, Diffusion model is an effective planner and data synthesizer for multi-task reinforcement learning, Adv. Neural Inf. Process. Syst. 36 (2023) 64896–64917.

[58] A. Kazerouni, E.K. Aghdam, M. Heidari, R. Azad, M. Fayyaz, I. Hacihaliloglu, D. Merhof, Diffusion models in medical imaging: a comprehensive survey, Med. Image Anal. 88 (2023) 102846.

[59] J. Ho, A. Jain, P. Abbeel, Denoising diffusion probabilistic models, Neural Information Processing Systems (2020).

[60] J. Devlin, Bert: Pre-training of deep bidirectional transformers for language understanding, (2018) arXiv preprint arXiv:1810.04805.

[61] E. Perez, F. Strub, H. De Vries, V. Dumoulin, A. Courville, Film: visual reasoning with a general conditioning layer, in: Proceedings of the AAAI Conference on Artificial Intelligence, 32, 2018.

[62] X. Tan, B. Huang, M. Batty, W. Li, Q.R. Wang, Y. Zhou, P. Gong, The spatiotemporal scaling laws of urban population dynamics, Nat. Commun. 16 (1) (2025) 2881.

[63] C. Shao, F. Xu, B. Fan, J. Ding, Y. Yuan, M. Wang, Y. Li, Beyond imitation: generating human mobility from context-aware reasoning with large language models, (2024) arXiv preprint arXiv:2402.09836.

[64] J. Wang, L. Dong, X. Cheng, W. Yang, Y. Liu, An extended exploration and preferential return model for human mobility simulation at individual and collective levels, Physica A 534 (2019) 121921.

[65] I. Sutskever, Sequence to Sequence Learning with Neural Networks, (2014) arXiv preprint arXiv:2409.3215.

[66] Y. Wang, T. Zheng, Y. Liang, S. Liu, M. Song, Cola: cross-city mobility transformer for human trajectory simulation, in: Proceedings of the ACM Web Conference 2024, 2024, pp. 3509–3520.

[67] D.P. Kingma, Adam: a method for stochastic optimization, (2014) arXiv preprint arXiv:1412.6980.

[68] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al., Pytorch: an imperative style, high-performance deep learning library, Adv. Neural Inf. Process. Syst. 32 (2019).

[69] Y. Yuan, J. Ding, H. Wang, D. Jin, Generating daily activities with need dynamics, ACM Trans. Intell. Syst. Technol. 15 (2) (2024) 1–28.

[70] V.I. Lcvenshtcin, Binary coors capable or " correcting deletions, insertions, and reversals, in: Soviet Physics-Doklady, 10, 1966.

[71] R. Shokri, M. Stronati, C. Song, V. Shmatikov, Membership inference attacks against machine learning models, in: 2017 IEEE Symposium on Security and Privacy (SP), IEEE, 2017, pp. 3–18.

Yuxiao Luo is currently a Ph.D student at The Hong Kong Polytechnic University. His research encompasses spatiotemporal data mining and human mobility modeling. He contributed to the conception of the research idea, the implementation of the algorithm, the design of experiments, and the writing of the manuscript.

Songming Zhang is currently a Ph.D student at the Nanjing University. His research interests include trustworthy machine learning and AI for science. He contributed to the implementation of the algorithm, the design of experiments, and the writing of the manuscript.

Kang Liu is currently an Associate Professor at Shenzhen Institute of Advanced Technology (SIAT), Chinese Academy of Sciences (CAS). Her research interests mainly focus on human mobility modeling and geospatial artificial intelligence (GeoAI). She contributed to the supervision of the experiments.

Yang Xu is currently an Associate Professor in the Department of Land Surveying and Geo-Informatics, The Hong Kong Polytechnic University. His research interests include human mobility, GIScience, and urban informatics. He contributed to the conception of the research idea, the supervision of the experiments, the writing, and the revision of the manuscript.

Ling Yin is currently a Professor at Shenzhen Institute of Advanced Technology (SIAT), Chinese Academy of Sciences (CAS), and she is also an adjunct Professor at the Faculty of Computer Science and Control Engineering, Shenzhen University of Advanced Technology. Her research interests include spatiotemporal data intelligence, human mobility, and spatial epidemiology. She contributed to the conception of the research idea, the supervision of the experiments, the writing, and the revision of the manuscript.