Controllable Human Trajectory Generation Using Profile-Guided Latent Diffusion

YIWEN SONG, Department of Electronic Engineering, Shenzhen International Graduate School, Tsinghua University, Beijing, China
JINGTAO DING and JIAN YUAN, Department of Electronic Engineering, Beijing National Research Center for Information Science and Technology (BNRist), Tsinghua University, Beijing, China
QINGMIN LIAO, Department of Electronic Engineering, Shenzhen International Graduate School, Tsinghua University, Beijing, China
YONG LI, Department of Electronic Engineering, Beijing National Research Center for Information Science and Technology (BNRist), Tsinghua University, Beijing, China

Trajectory generation is a vital element in AI applications. Firstly, it enables simulation such as traffic simulation and epidemic spreading modeling. Secondly, it can provide synthetic privacy-preserving data for training AI models. Notably, trajectory generation featuring controllable user profiles holds substantial value in generating customized mobility trajectories tailored to diverse requirements. However, relevant work is still lacking. On the one hand, traditional deep generative models fall short in guiding controllable trajectory generation due to the statistical nature of human mobility patterns and the corresponding insufficient control mechanisms. On the other hand, though the diffusion model has demonstrated strong generative capabilities in many fields, to achieve controllable generation on discrete trajectory data, we still need to redesign the structure of the continuous diffusion model. In this article, we introduce a controllable trajectory generation framework that leverages a continuous diffusion model and classifier guidance for more robust condition control. Our proposed framework comprises two modules: a latent trajectory diffusion model and a trajectory classifier for profile guidance. Experiments on two real-world mobility datasets consistently demonstrate its capability of generating trajectories matching given user profiles and conforming to human mobility patterns. Our source code and trained models are released at https://github.com/tsinghua-fib-lab/User-Profile-Guided-Latent-Diffusion.

CCS Concepts: • **Computing methodologies** → **Knowledge representation and reasoning**; • **Information systems** → **Data mining**;

Additional Key Words and Phrases: controllable generation, diffusion model, human mobility, user profile.

Associate Editor: Yanjie Fu

This work is supported by a grant from the National Natural Science Foundation of China under Grants 62272260 and U20B2060.

Authors' Contact Information: Yiwen Song, Department of Electronic Engineering, Shenzhen International Graduate School, Tsinghua University, Beijing, China; e-mail: songyw23@mails.tsinghua.edu.cn; Jingtao Ding (corresponding author), Department of Electronic Engineering, Beijing National Research Center for Information Science and Technology (BNRist), Tsinghua University, Beijing, China; e-mail: dingjt15@tsinghua.org.cn; Jian Yuan, Department of Electronic Engineering, Beijing National Research Center for Information Science and Technology (BNRist), Tsinghua University, Beijing, China; e-mail: jyuan@tsinghua.edu.cn; Qingmin Liao, Department of Electronic Engineering, Shenzhen International Graduate School, Tsinghua University, Beijing, China; e-mail: liaoqm@tsinghua.edu.cn; Yong Li, Department of Electronic Engineering, Beijing National Research Center for Information Science and Technology (BNRist), Tsinghua University, Beijing, China; e-mail: liyong07@tsinghua.edu.cn.

ACM 1556-472X/2024/12-ART22
https://doi.org/10.1145/3701736

ACM Reference format:
Yiwen Song, Jingtao Ding, Jian Yuan, Qingmin Liao, and Yong Li. 2024. Controllable Human Trajectory Generation Using Profile-Guided Latent Diffusion. *ACM Trans. Knowl. Discov. Data.* 19, 1, Article 22 (December 2024), 25 pages.
https://doi.org/10.1145/3701736

1 Introduction

Trajectory generation, which entails the generation of synthetic trajectories capable of faithfully reproducing the individual statistical patterns of human mobility [27], can replace real data collection and thus is crucial to downstream AI applications such as traffic simulation and personalized location recommendations. Trajectory generation has been widely used in various areas, including epidemic spreading [10], privacy protection of data collection [31, 46], origin-destination demand estimation [5, 33, 34], and so on. More importantly, given a specific user profile, controllable generation is of higher application value, as it can generate mobility trajectories tailored to diverse needs. In this scenario, the user profile refers to discrete portraits such as age, educational background, income level, and gender, and continuous ones such as home location. For example, in a travel app, controllable generation tailors itineraries based on user profiles, such as age and income, offering personalized experiences such as luxury tours, suiting individual preferences and lifestyles. Notably, physical attributes and environmental conditions affect fine-grained human motion [52, 57], but have minimal impact on macro-level human mobility. In contrast, user profiles significantly influence the human mobility patterns we are studying. Nevertheless, developing a practical framework for user profile controllable trajectory generation is still an open challenge.

Existing trajectory generation methods include mechanistic generative models [15, 16] and deep generative models [27]. The former fits simple probability distributions through data and characterizes different people's mobility patterns in a universal mechanism. For example, Timegeo [16] fits a probabilistic model, which is based on a rank-based **Exploration and Preferential Return (EPR)** mechanism [40], to real-world human mobility trajectories and generates synthetic trajectories using the model. However, since mechanistic generative models assume that all users conform to a common mobility mechanism, they are not suitable for generating trajectories for users with different profile attributes. Thus, we mainly focus on the latter, which can be categorized into three groups based on their model architectures. The first group of models relies on **Variational Autoencoder (VAE)** [19, 26, 60], featuring both an encoder and a decoder. The encoder learns the distribution of latent variables from real trajectory data, while the decoder generates synthetic trajectories. For example, VOLUNTEER [26] and EETG [60] are both robust VAE-based trajectory generators. VOLUNTEER uses dual VAEs to capture group-based user distribution, while EETG learns global and local semantics, such as random routing change, to generate trajectories with road-level precision. The second group of models is based on **Generative Adversarial Network (GAN)** [5, 10, 11, 31, 50], which trains a generator using adversarial training, which can generate synthetic data that are indistinguishable from real data. For example, MoveSim [10] employs a generator based on a self-attention-driven sequential modeling network, accompanied by a discriminator that incorporates mobility regularity-aware loss. The last category of models is grounded in **Generative Adversarial Imitation Learning (GAIL)** [12, 46], which models human movement as a decision-making process and trains it in a manner similar to GAN. For example, PateGail [46] is a privacy-preserving imitation learning model trained on mobility data stored on distributed user devices, which models human movement using the rank-based EPR mechanism [40], similar to Timegeo [16]. Additionally, SAND [58] is a GAIL-based knowledge-driven activity simulation framework that models the progression of human requirements as the

fundamental mechanism motivating activity generation. Unfortunately, these groups of models are not ideal for a controllable generation due to their weak condition mechanism of attribute concatenation [37]. Consequently, controllable trajectory generation is very challenging given its high application value.

We aim to bridge the gap between advanced deep generative models and the challenge of controllable trajectory generation by designing a model that can generate synthetic trajectories given user profiles such as age, gender, home location, and so on. Moreover, the model should generate trajectories flexibly according to different needs. For example, it should be able to achieve compositional control by dealing with profile combinations that never occur in training datasets. However, it is challenging for existing learning-based methods to generate realistic trajectories for users with a specific profile. Firstly, existing work has proved the strong correlation between the trajectories of users and their profiles [25], which is usually reflected statistically in metrics such as travel distance, gyration radius, and so on. Nevertheless, the widely used controllable generation method of attribute concatenation [37] cannot explicitly supervise the statistical features of generated trajectories. Secondly, existing models aim to learn maximum-likelihood objectives [10, 37, 46] and thus must be ineffective in tailoring new objectives to different tasks, such as dealing with combinations of controls.

In pursuit of achieving flexible generation, we capitalize on recent advances in diffusion models, which have made remarkable breakthroughs in many areas, including images [8, 28, 32], texts [20], and protein structures [1]. Importantly, diffusion models can incorporate conditions at each diffusion step and achieve effective, controllable generation through such guidance, which has benefited control tasks including text generation [20], image generation [8], network generation [24], traffic generation [62], and motion generation [43]. Notably, the object of trajectory generation tends to be a specific place or a region, which corresponds to the discrete space. Yet, existing discrete diffusion models [2] cannot capitalize on the above advantages of the diffusion model.

To achieve controllable human trajectory generation, we propose a profile-guided latent diffusion modeling framework. Motivated by latent diffusion model [32], our trajectory generation method performs diffusion and denoising processes in latent space. Concretely, we add an extra embedding and mapping process between continuous latent embedding and discrete location indexes based on DDPM [13]. Our denoising model is based on a transformer with a spatio-temporal context embedding process that adapts to the characteristics of the trajectory data. To facilitate profile-guided trajectory generation, we integrate user profile information into every diffusion step. This is accomplished by adding the gradient of the user profile classifier to that of the update of the noise sample in the denoising process of the diffusion model. Hence, this successfully injects implicit knowledge about the statistical properties of human mobility from the user profile classifier into the entire generated trajectory, which is rather difficult for existing conditional generation method [37]. Moreover, given classifiers for multiple independent control tasks, this gradient-based controllable generation framework makes it practical to generate from the intersection of multiple controls by replacing the gradient of a single classifier with the sum of gradients from multiple classifiers.

In summary, our main contributions are as follows.

- —We take the initiative in tackling the problem of generating human trajectories conditioned on specific user profiles, which is of high application value.
- —We present a novel framework for controllable trajectory generation that combines a diffusion model with classifier guidance, leveraging the strong generative ability of the diffusion model and the exquisite control ability of classifier guidance. Furthermore, we convert the discrete trajectory generation task into continuous space with a latent diffusion model.
- —Comprehensive experiments on two real-world datasets demonstrate our framework's superiority over several state-of-the-art methods in various aspects. Our framework maintains

a good balance between controllability and realism and can achieve compositional control. Moreover, our framework performs best in different application experiments, including generating home-based travels and testing the effectiveness of the synthetic trajectories. We also perform an ablation study to validate the rationality and correctness of our model design.

2 Preliminaries

2.1 Problem Definition

We denote real human mobility data as $[D_1, D_2, \ldots, D_n]$, where N is the number of users, $D_i = [d_1, d_2, \ldots, d_m]$ represents the trajectory data of the ith user, and M is the length of the trajectory sequence. The jth data point $d_j = (g_j, w_j, h_j)$ includes geographical location g_j, date w_j, and hour h_j. Besides, we denote the tuples of trajectory sequence and its corresponding user profile as $[(P_1, D_1), (P_2, D_2), \ldots, (P_n, D_n)]$. The user profile P_i for the ith user includes attributes such as age, gender, education level, and home location. As shown in Table 1, a table of the primary notations is provided to make this article more understandable. With these notations, we can formulate the problems of mobility trajectory generation and user profile-controllable trajectory generation as follows:

Definition 2.1 (Mobility Trajectory Generation). Given a real-word mobility trajectory dataset $[D_1, D_2, \ldots, D_n]$, the objective is to generate a realistic mobility trajectory $[\hat{D}_1, \hat{D}_2, \ldots, \hat{D}_r]$.

Definition 2.2 (User Profile Controllable Trajectory Generation). Given a real-word mobility trajectory dataset with the corresponding user profiles $[(P_1, D_1), (P_2, D_2), \ldots, (P_n, D_n)]$ and specific user profiles $[\hat{P}_1, \hat{P}_2, \ldots, \hat{P}_r]$, where r is the number of given user profiles, the objective is to generate a realistic mobility trajectory $[(\hat{P}_1, \hat{D}_1), (\hat{P}_2, \hat{D}_2), \ldots, (\hat{P}_r, \hat{D}_r)]$.

Take a toy example to elucidate the problem under study. The input is provided as labeled data (P, D). For instance, P may represent attributes like "gender: male," "education level: high school," "daily income: 76.0 (yuan)," and "age: 28." Meanwhile, D could consist of data points such as $((116.45079, 39.89415), Monday, 3)$, indicating the user's location at this latitude and longitude at 3 am on Monday. Given "gender: female," the objective is for the model to generate trajectories that conform to female travel patterns.

This article aims to build a synthetic data generation framework capable of generating trajectories close to real-world data in terms of statistical metrics based on the given user profiles. The model-generated trajectories do not belong to any specific real user, hence eliminating privacy issues.

2.2 Data Analysis

To further clarify the motivation of our design, we conduct data analysis on two real-world datasets to validate the strong correlation between human mobility and user profiles. Our data analysis is primarily based on the Mobile Dataset and the Tencent Dataset, as elaborated later in Section 4.1.1. Notably, we choose a trajectory length of 7 days in our study for two reasons. Firstly, we have conducted a thorough review of existing works and find that almost all are centered on trajectories spanning from 1 to 7 days [3, 5, 9, 10, 16, 21, 25, 31, 53]. Secondly, our data analysis have illustrated that trajectories within this period can display periodicity, reflecting enduring patterns of human activity.

As evidenced by previous studies [25, 49, 51, 54] and our data analysis, users with varying profile attributes indeed exhibit markedly distinct mobility patterns. In particular, Figure 1 illustrates the distributions of gyration radius and travel distance across different user profiles within the two datasets. Gyration radius and travel distance, to be further elucidated in Section 4.1.3, serve as indicators of human mobility, with larger values indicating greater mobility. These two indicators

Table 1. Notations

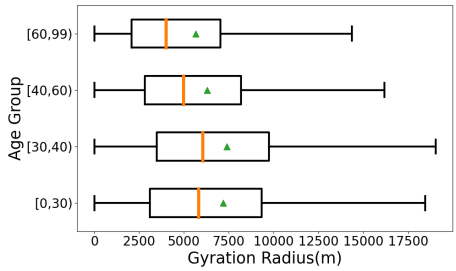
Symbols	Description
(P_i, D_i)	Tuple of trajectory sequence D and its corresponding user profile P of the ith user
$d = (g, w, h)$	Trajectory data point, where g is geographical location, w is date, and h is hour
K	Total diffusion steps
$\boldsymbol{x}_k$	Latent vector at diffusion step k
$\boldsymbol{g}$	Discrete location ID sequence
$p_\theta(\boldsymbol{x}_{0:K})$	Reversed denoising process
$q(\boldsymbol{x}_{1:K}\|x_0)$	Diffusion process
$\mathcal{G}$	Space of all location IDs
$\beta_1, ..., \beta_K$	Variance schedule of the diffusion process
α_k	Notation of $1 - \beta_k$ at diffusion step k
$\bar{\alpha}_k$	Notation of $\prod_{s=0}^{k} \alpha_s$ at diffusion step k
pos	Positional information in a sequence
c	Classification label
y	Sign function

can be used to explore differences in mobility patterns among different groups. For instance, in terms of different user profiles, people's mobility patterns may vary. In Figure 1(a), it is apparent that younger individuals generally have higher mobility, a notion consistent with common sense. Notably, users in the age group of 30–40 demonstrate the broadest range of activities probably due to commuting and business, while older groups have fewer activities potentially because of health issues. Additionally, Figure 1(b) depicts the distribution of gyration radius across different income levels, revealing that users with higher income tend to have a larger gyration radius. The possible reason is that people with higher incomes tend to own private transportation, thus further increasing the range of movement. Similarly, Figure 1(c) and (d) also highlight a significant difference in the distributions of travel distance among different age groups and genders. Supplementarily, the 7-day length of the trajectory can reflect enduring patterns of human activity and is consistent with common sense, and is therefore a reasonable trajectory length.

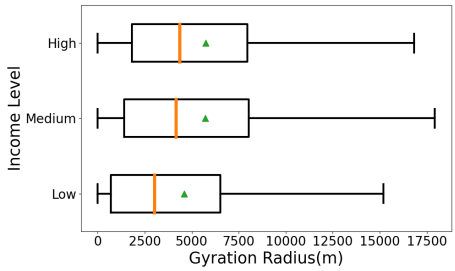
Consequently, based on the above data analysis, it is evident that user profiles significantly influence human mobility, thereby providing a strong foundation and premise for our method.

2.3 Diffusion Model

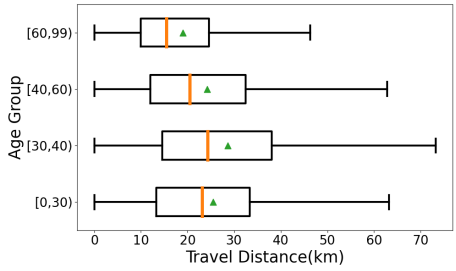
Before introducing our method in this article, we provide a concise overview of fundamental concepts essential for comprehending the **Denoising Diffusion Probabilistic Model (DDPM)** [13]. The DDPM establishes a forward diffusion process, which involves adding noise to original data samples to transform them into samples following a simple distribution. Then, the denoising model develops the ability to gradually reverse the diffusion process and restore the original samples. The actual generation process involves sampling from the simple distribution and then reversely denoising.



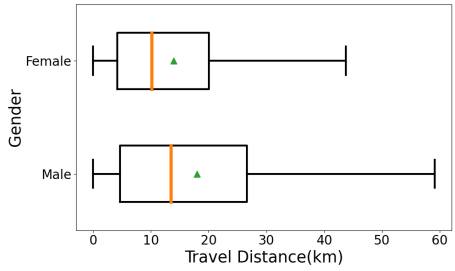
(a) Gyration radius distribution of different age groups, Mobile.



(b) Gyration radius distribution of different income levels, Tencent.



(c) Travel distance distribution of different age groups, Mobile.



(d) Travel distance distribution of different genders, Tencent.

Fig. 1. The distribution of gyration radius ((a) and (b)) and travel distance ((c) and (d)) of different user profiles on two real-world datasets.

Specifically, the DDPM defines a forward noise-adding process, gradually adding noise to the real data sample $\boldsymbol{x}_0$: $q(\boldsymbol{x}_t|\boldsymbol{x}_0) = N(\boldsymbol{x}_t; \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0, (1-\bar{\alpha}_t)\mathbf{I})$, where the constant $\bar{\alpha}_t$ is a hyperparameter. Based on the above Markov chain, we can sample $\boldsymbol{x}_t = \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_t$, where $\boldsymbol{\epsilon}_t \sim \mathcal{N}(0, \mathbf{I})$.

The diffusion model acquires the ability to reconstruct data from noise through a reversed denoising process:

$$p_\theta\left(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t\right) = \mathcal{N}\left(\boldsymbol{x}_{t-1}; \boldsymbol{\mu}_\theta\left(\boldsymbol{x}_t\right), \Sigma_\theta\left(\boldsymbol{x}_t\right)\right), \tag{1}$$

where neural networks $\boldsymbol{\mu}_\theta$ and Σ_θ are used to predict the conditional distribution of $\boldsymbol{x}_{t-1}$ at diffusion step t. The reverse denoising model is trained using the log-likelihood of the variational lower bound of $\boldsymbol{x}_0$ [18], formulated as

$$\begin{aligned}\mathcal{L} = -log p_\theta\left(\boldsymbol{x}_0 \mid \boldsymbol{x}_1\right) + \sum_{t=2}^{T} \mathcal{D}_{KL}\left(q\left(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}_0\right) || \, p_\theta\left(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t\right)\right) \\ + \mathcal{D}_{KL}(q\left(\boldsymbol{x}_T \mid \boldsymbol{x}_0\right) || \, p(\boldsymbol{x}_T)),\end{aligned} \tag{2}$$

where the last term is irrelevant to training and thus can been omitted. Since both q and p_θ are Gaussian distributions, we can evaluate $\mathcal{D}_{KL}$ through the means and covariances of them. By re-parameterizing $\boldsymbol{\mu}_\theta$ as a noise prediction network $\boldsymbol{\epsilon}_\theta$, it can be trained through the mean square error between the predicted noise $\boldsymbol{\epsilon}_\theta\left(\boldsymbol{x}_t\right)$ and the real sampled Gaussian noise $\boldsymbol{\epsilon}_t$:

$$\mathcal{L}_{simple}(\theta) = \left|\boldsymbol{\epsilon}_\theta\left(\boldsymbol{x}_t\right) - \boldsymbol{\epsilon}_t\right|^2. \tag{3}$$

Furthermore, as the full $\mathcal{D}_{KL}$ term needs to be optimized to train the diffusion model using the learned reverse process covariance Σ_θ, we adopt the method proposed by Nichol et al. [29]. We train $\boldsymbol{\epsilon}_\theta$ using $\mathcal{L}_{\text{simple}}$, and training Σ_θ using the full $\mathcal{L}$. After training p_θ is completed, new data samples can be generated by initializing $\boldsymbol{x}_{t_{\max}} \sim \mathcal{N}(0, \mathbf{I})$ and sampling $\boldsymbol{x}_{t-1} \sim p_\theta\left(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t\right)$.

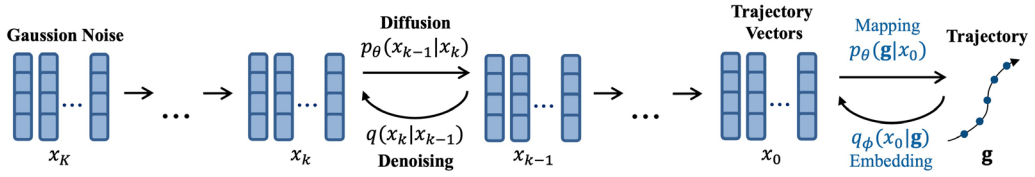


Fig. 2. Diffusion–denoising process.

3 Method

Our proposed framework comprises two modules: latent trajectory diffusion model and trajectory classifier for profile guidance. We first introduce the design of the two modules in Sections 3.1 and 3.2, then elaborate on the controllable generation process in Section 3.3.

3.1 Latent Trajectory Diffusion Model

We propose a latent trajectory diffusion model, trained on a large dataset of unlabeled trajectory data, to accurately capture the complicated patterns in mobility trajectories. This model generates trajectories that adhere to human mobility patterns. Assuming a total of K diffusion steps, as depicted in Figure 2, to generate discrete trajectory sequences using continuous diffusion models, we incorporate a diffusion–denoising process in the latent vector space, where $\{\boldsymbol{x}_k\}_{k=0}^{K}$ represents latent vectors of dimension $L \times D \in \mathbb{R}^d$. Additionally, this process involves mapping the continuous latent vector $\boldsymbol{x}_0$ back to a discrete location ID sequence $\boldsymbol{g}$, where $\boldsymbol{x}_0$ corresponds to the latent vector of the location ID sequence $\boldsymbol{g}$.

Latent Diffusion and Denoising Process. The probability transitions between the latent vector $\boldsymbol{x}_0$ and the location ID sequence $\boldsymbol{g}$ are parameterized as follows:

$$q_\phi(\boldsymbol{x}_0|\boldsymbol{g}) = \mathcal{N}(EMB(\boldsymbol{g}), {\sigma_0}^2\mathbf{I}), \tag{4}$$

$$p_\theta(\boldsymbol{g}|\boldsymbol{x}_0) = \prod_{i=1}^{n} p_\theta(\boldsymbol{g}_i|\boldsymbol{x}_i), \tag{5}$$

where $EMB(\boldsymbol{g}) \in \mathbb{R}^d$ represents the embedding of the location ID sequence $\boldsymbol{g}$. The transition from $EMB(\boldsymbol{g})$ to $\boldsymbol{x}_0$, parameterized by $q_\phi(\boldsymbol{x}_0|\boldsymbol{g})$, also conforms to a Gaussian distribution, with mean $EMB(\boldsymbol{g})$ and covariance matrix ${\sigma_0}^2\mathbf{I}$. ϕ represents the learnable embedding parameters. The denoising process also includes a learnable parameterized model $p_\theta(\boldsymbol{g}_i|\boldsymbol{x}_i)$, where the output is a softmax distribution over the space of all location IDs $\mathcal{G}$. Drawing from previous research [20], it is established that predicting $\hat{\boldsymbol{x}}_0$ and $\boldsymbol{x}_{k-1}$ are mathematically equivalent. Furthermore, predicting $\hat{\boldsymbol{x}}_0$ can make the estimated $\hat{\boldsymbol{x}}_0$ closer to the embedding of a specific location ID. Consequently, the denoising model directly predicts $\hat{\boldsymbol{x}}_0$ and then performs posterior sampling according to $p(\boldsymbol{x}_{k-1}|\hat{\boldsymbol{x}}_0, \boldsymbol{x}_k)$ to obtain denoised sample $\boldsymbol{x}_{k-1}$:

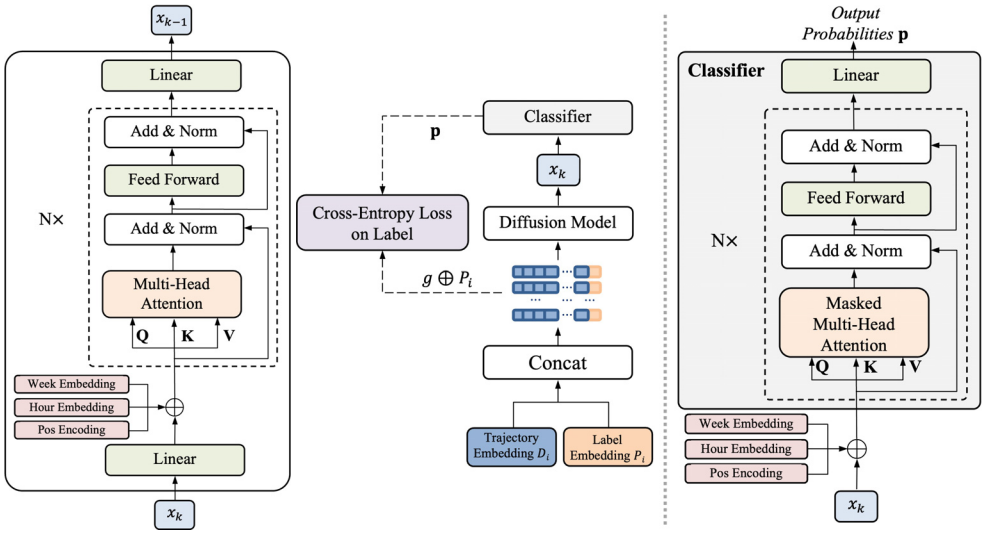
$$\boldsymbol{x}_{k-1} = \boldsymbol{\mu}(\hat{\boldsymbol{x}}_0, \boldsymbol{x}_k) + \sigma_{k-1}\boldsymbol{\varepsilon}, \boldsymbol{\varepsilon} \sim \mathcal{N}(0, \mathbf{I}), \tag{6}$$

where the mean $\boldsymbol{\mu}(\hat{\boldsymbol{x}}_0, \boldsymbol{x}_k)$ is given by

$$\boldsymbol{\mu}(\hat{\boldsymbol{x}}_0, \boldsymbol{x}_k) = \frac{\sqrt{\bar{\alpha}_{k-1}}\beta_k}{1-\bar{\alpha}_k}\hat{\boldsymbol{x}}_0 + \frac{\sqrt{\alpha_k}(1-\bar{\alpha}_{k-1})}{1-\bar{\alpha}_k}\boldsymbol{x}_k \tag{7}$$

and the variance $\Sigma(\hat{\boldsymbol{x}}_0, \boldsymbol{x}_k)$ is given by

$$\sigma_{k-1} = \frac{1-\bar{\alpha}_{k-1}}{1-\bar{\alpha}_k}\beta_k, \tag{8}$$



(a) denoising model (b) profile classifier

Fig. 3. (a) Structure of the transformer-based denoising model. (b) Structure of the user profile classifier.

where

$$\bar{\alpha}_k = \prod_{s=0}^{k} \alpha_s, \ \alpha_k = 1 - \beta_k. \tag{9}$$

The diffusion model aims to minimize the following loss function:

$$\begin{aligned} \mathcal{L}^{diff}(\boldsymbol{g}) = & \mathbb{E}q_\phi(\boldsymbol{x}_{0:K}|\boldsymbol{g}) \Big[\mathcal{L}(\boldsymbol{x}_0) + \left|EMB(\boldsymbol{g}) - \boldsymbol{\mu}_\theta(x_1, 1)\right|^2 \\ & - \log p_\theta(\boldsymbol{g}|\boldsymbol{x}_0) \Big]. \end{aligned} \tag{10}$$

The right side of the equation consists of three terms, which are respectively: $\mathcal{L}(\boldsymbol{x}_0)$, $|EMB(\boldsymbol{g}) - \boldsymbol{\mu}_\theta(x_1, 1)|^2$, and $\log p_\theta(\boldsymbol{g}|\boldsymbol{x}_0)$. The first term $\mathcal{L}(\boldsymbol{x}_0)$ denotes the optimization objective of all the diffusion–denoising processes in the latent space from the first step to the Kth step:

$$\begin{aligned} \mathcal{L}(\boldsymbol{x}_0) &= \sum_{k=1}^{K} \mathbb{E}_{q(\boldsymbol{x}_k|\boldsymbol{x}_0)} ||\boldsymbol{\mu}_\theta(\boldsymbol{x}_k, k) - \hat{\boldsymbol{\mu}}(\boldsymbol{x}_k, \boldsymbol{x}_0)||^2 \\ &= \sum_{k=1}^{K} \mathbb{E}_{q(\boldsymbol{x}_k|\boldsymbol{x}_0)} \frac{\sqrt{\bar{\alpha}_{k-1}}\beta_k}{1 - \bar{\alpha}_k} ||\boldsymbol{x}_0 - f_\theta(\boldsymbol{x}_k, k)||^2, \end{aligned} \tag{11}$$

where $\hat{\boldsymbol{\mu}}(\boldsymbol{x}_k, \boldsymbol{x}_0)$ refers to the mean of $q(\boldsymbol{x}_{k-1}|\boldsymbol{x}_0, \boldsymbol{x}_k)$ and $\boldsymbol{\mu}_\theta(\boldsymbol{x}_k, k)$ refers to the mean of $p_\theta(\boldsymbol{x}_{k-1}|\boldsymbol{x}_k)$. The second term stands for the optimization objective of the diffusion-denoising process in the continuous latent space at step $k = 0 \rightarrow 1$, and the third term represents the optimization objective of the mapping from embedding $\boldsymbol{x}_0$ to discrete location IDs. Moreover, the model adopts an end-to-end training approach, enabling it to learn both the denoising process and trajectory embedding simultaneously.

Transformer-Based Denoising Model. Transformer has proved to be highly suitable for handling sequential data, and trajectories also fall under this category. Consequently, we design a denoising model based on feature encoding and a transformer, as illustrated in Figure 3(a). Different from

positional encoding [44], in diffusion step k of the denoising process, we perform feature encoding to combine temporal context embedding, positional encoding, and diffusion step embedding. Specifically, the feature encoding is formulated as follows:

$$\mathrm{E}(w, h, pos, k) = E_t(w, h) + E_p(pos) + E_k(k), \tag{12}$$

where w, h, and pos represent the date, hour, and positional information in the sequence, respectively. Notably, after the attention calculation, we have

$$\boldsymbol{a} = \mathrm{MA}(\mathrm{Linear}(\boldsymbol{x}_k) + \mathrm{PE}(w, h, pos, k)), \tag{13}$$

where Linear refers to linear layers and MA refers to Multi-Head Attention. The denoising model can effectively capture temporal and positional information within the sequence while executing the denoising process.

3.2 Trajectory Classifier for Profile Guidance

We have designed a transformer-based classifier to classify $\boldsymbol{x}_k$ and derive user profiles as classification labels, as illustrated in Figure 3(b). During the training process, after sampling diffusion steps k, we add noise to the latent vector $\boldsymbol{x}_0$ for k steps to give $\boldsymbol{x}_k$. In order to facilitate the subsequent implementation of gradient guidance, we train the model to classify x_k. It calculates the parameterized probability distribution of the classification labels $p(c|\boldsymbol{x}_k)$ and then computes the cross-entropy loss function between the predicted labels and the input labels.

Given the trajectory data with label information $[(P_1, D_1), (P_2, D_2), \ldots, (P_n, D_n)]$, the trajectory sequence g is concatenated with the label information P_i to give $g \bigoplus P_i$. To input both trajectories and labels while updating the classifier's gradients based on its label results, we feed $g \bigoplus P_i$ into the classifier. The context embedding and attention calculation processes are similar to those in the transformer structure [44], except that the classifier calculates a Masked Multi-head Attention mechanism.

For a specific sampled k, we denote the output probability that $\boldsymbol{x}_k$ belongs to the jth label category c_j as $p_{j,k} = p(c_j|\boldsymbol{x}_k)$. The proposed classifier aims to minimize the cross-entropy loss between the given label and the predicted label:

$$\mathcal{L}_{classifier} = -\sum_{j=1}^{M} y_j \log p_{j,k}, \tag{14}$$

where M is the number of label categories. y_j represents a sign function. It takes the value of 1 if the prediction is correct, and 0 if the prediction is incorrect. This function is used to distinguish between accurate and erroneous predictions within the model.

3.3 Guided Generation with User Profile

In broad terms, the gradient of controllable generation can be viewed as the sum of the gradient of unconditional generation and that of the classifier. Specifically, as is shown in Figure 4, given the targeted user profile c for the trajectory, we further design a classifier-guided generation framework to generate trajectories. Notably, the gradient update diminishes in strength as it approaches the trajectory generation, which is because in practical experiments, as the control information (i.e., gradient) is introduced during the denoising process, the trajectory data sample increasingly aligns with the characteristics of user trajectories that share the given attribute. Firstly, the pre-trained denoising model estimates the probability of the transition from $\boldsymbol{x}_k$ to $\boldsymbol{x}_{k-1}$ and sample $\boldsymbol{x}_{k-1}$. Then, the classifier applies gradient-based corrections to $\boldsymbol{x}_{k-1}$, and the updated $\hat{\boldsymbol{x}}_{k-1}$ is the input for the denoising model in the next diffusion step $k-1$. In the diffusion step k, the latent vector $\boldsymbol{x}_{k-1}$ is

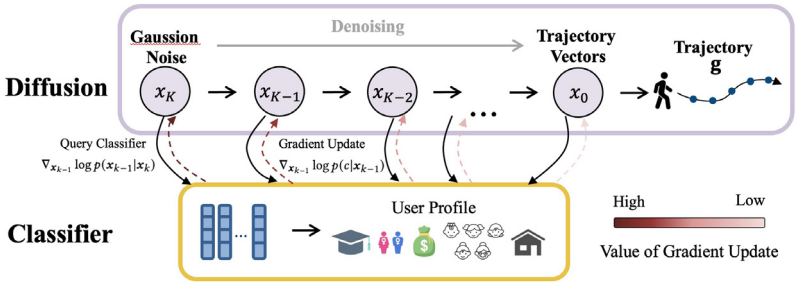


Fig. 4. Process of classifier-guided trajectory generation.

computed by the denoising model using the following equation:

$$\boldsymbol{x}_{k-1} = \boldsymbol{\mu}(\hat{\boldsymbol{x}}_0, \boldsymbol{x}_k) + \Sigma\,(\hat{\boldsymbol{x}}_0, \boldsymbol{x}_k)^{\frac{1}{2}} \cdot \boldsymbol{\varepsilon}, \boldsymbol{\varepsilon} \sim \mathcal{N}(0, 1). \tag{15}$$

The vector obtained by concatenating $\boldsymbol{x}_{k-1}$ with the encoded label, $\boldsymbol{x}_{k-1} \bigoplus E_c(c)$, serves as the input to the classifier. The probability of the classifier guiding the computation of the latent vectors $\{\boldsymbol{x}_k\}_{k=0}^K$ can be expressed as $p(\{\boldsymbol{x}_k\}_{k=0}^K|c) = \prod_{k=1}^K p(\boldsymbol{x}_{k-1}|\boldsymbol{x}_k, c)$. It can be decomposed into probabilities within diffusion step k as follows:

$$\begin{aligned} \log p(\boldsymbol{x}_{k-1}|\boldsymbol{x}_k, c) &= \log\left(\frac{p(\boldsymbol{x}_{k-1}, c|\boldsymbol{x}_k)}{p(c|\boldsymbol{x}_k)}\right) \\ &\propto \log p(\boldsymbol{x}_{k-1}|\boldsymbol{x}_k) \cdot p(c|\boldsymbol{x}_{k-1}, \boldsymbol{x}_k) \\ &= \log p(\boldsymbol{x}_{k-1}|\boldsymbol{x}_k) \cdot p(c|\boldsymbol{x}_{k-1}). \end{aligned} \tag{16}$$

To enhance control quality, we introduce a weight coefficient λ, and the gradient update process for $\boldsymbol{x}_{k-1}$ in diffusion step k can be expressed as:

$$\begin{aligned} &\nabla_{\boldsymbol{x}_{k-1}} \log p(\boldsymbol{x}_{k-1}|\boldsymbol{x}_k, c) \\ &= \lambda \cdot \nabla_{\boldsymbol{x}_{k-1}} \log p(\boldsymbol{x}_{k-1}|\boldsymbol{x}_k) + \nabla_{\boldsymbol{x}_{k-1}} \log p(c|\boldsymbol{x}_{k-1}). \end{aligned} \tag{17}$$

To strengthen attribute control, the latent vector is updated H times at each diffusion step k.

In the implementation of the framework, the term $\nabla_{\boldsymbol{x}_{k-1}} \log p(\boldsymbol{x}_{k-1}|\boldsymbol{x}_k, c)$ refers to the gradient of conditional generation, the term $\cdot\nabla_{\boldsymbol{x}_{k-1}} \log p(\boldsymbol{x}_{k-1}|\boldsymbol{x}_k)$ refers to the gradient of unconditional generation and the last term $\nabla_{\boldsymbol{x}_{k-1}} \log p(c|\boldsymbol{x}_{k-1})$ refers to the gradient of the classifier. Consequently, we can achieve guided generation by summing the gradients of the unconditional generation and the classifier.

In essence, through the incorporation of the above diffusion model and trajectory classifier, and by adhering to the guided generation process, our proposed framework demonstrates the capability to consistently generate controllable trajectories based on given user profiles.

4 Experiments

4.1 Experiment Settings

4.1.1 Dataset. Our main experiments are conducted on two real-world mobility datasets collected from two data sources, covering the geographical area of Beijing, China. These two datasets are labeled with user profiles, such as income, gender, education, and age. To enrich our study and rigorously test the model's adaptability and generalizability, we generate home-based travels on two other datasets, covering the geographical area of Shenzhen, China, and Senegal, the results of which will be described later in Section 4.5.1. Table 2 presents the basic information of these four datasets, and further details are described below. For each dataset, we perform a split into training,

Table 2. Basic Information of the Original Real-World Mobility Datasets

Dataset	Region	Time Period	# of Users	# of Trajectories	# of Trajectory Points
Mobile	Beijing, China	1 July 2017–31 August 2017	7,264	7,264	1,220,352
Tencent	Beijing, China	1 October 2019–31 December 2019	9,561	9,561	1,606,248
Shenzhen	Shenzhen, China	1 January 2018–31 December 2018	10,000	10,000	1,680,000
Senegal	Senegal	1 January 2013–31 December 2013	10,000	10,000	1,680,000

testing, and validation datasets with a ratio of 0.7:0.2:0.1. It should be noted that, throughout the data processing for both model training and evaluation, these datasets preserve only key profiles such as age, gender, income level, and education level, meaning that we cannot obtain real user identity.

Mobile Dataset. This dataset, obtained from a local mobile operator, contains records of users' connection times and locations to cellular base stations. Additionally, users' profiles, such as income level, gender, education level and age, are collected through digital surveys. For example, the users are divided into four groups based on their different income levels. Table 3(a) illustrates the distribution of user profiles.

Tencent Dataset. This dataset, sourced from a social network platform, documents users' mobility behavior, while user profiles are gathered through surveys. Table 3(b) illustrates the distribution of user profiles.

Shenzhen Dataset. This dataset, sourced from a social network platform, documents users' mobility behavior.

Senegal Dataset. This dataset, obtained from a local mobile operator in Senegal, contains records of users' connection times and locations to cellular base stations.

4.1.2 Baseline. Generally, our baselines include two categories of models, mechanistic generative models and deep generative models. For mechanistic generative models, we need to filter out data that meet the criteria for each condition as sub-datasets to simulate controllable generation. For deep generative models, we consider mainstream models such as GAN [11], VAE [19], and GAIL [12]. We generally concatenate the latent variables with the embeddings of given conditions in the latent space.

Timegeo [16]. This method is a mechanistic modeling framework that models temporal human mobility patterns based on several designed features such as dwell rate, and spatial patterns based on a rank-based EPR mechanism [40].

MoveSim [10]. This method introduces an innovative approach that combines model-free generative adversarial techniques with domain knowledge. It utilizes a self-attention-based sequential modeling network to capture intricate temporal dynamics and incorporates urban structural information to produce coherent and meaningful trajectories. MoveSim also leverages pre-trained mobility patterns for accelerated learning.

CVAE [38]. This approach, grounded in controllable VAEs and transformers, presents a trajectory generation model that enhances the generative modeling capabilities of VAEs by introducing an automatic controller.

PateGail [46]. This method is a privacy-preserving imitation learning model for generating human mobility trajectories. It achieves high-quality trajectory generation and supports practical applications through decentralized data training and aggregation mechanisms.

4.1.3 Evaluation Metrics. We consider evaluating the synthetic trajectories from two perspectives, namely controllable generation success rate and trajectory quality. We generate synthetic trajectories conditioned on the corresponding user profile of each trajectory in the testing or

Table 3. Distribution of User Profiles on Mobile and Tencent datasets

(a) Mobile Dataset

User Profile	Category
Income	Low (15.80%), Lower Medium (26.46%), Upper Medium (20.43%), High (37.31%)
Gender	Male (35.13%), Female (64.87%)
Education	Junior High School (9.03%), Senior High School (20.98%), Undergraduate (58.59%), Postgraduate (11.40%)
Age	0–30 (18.72%), 30–40 (30.78%), 40–60 (41.13%), 60–99 (9.36%)

(b) Tencent Dataset

User Profile	Category
Income	Low (17.98%), Medium (57.76%), High (24.27%)
Gender	Male (54.50%), Female (45.50%)
Education	Senior High School (13.28%), Undergraduate (56.11%), Postgraduate (30.60%)
Age	0–30 (29.51%), 30–40 (22.48%), 40–60 (37.35%), 60–99 (10.67%)

validation datasets. Then, we calculate the two kinds of metrics for the overall collection of the synthetic trajectories.

Controllable Generation Success Rate. We design our evaluation metrics based on weighted F1-score, which is widely used in user profiling studies [23, 25]. Specifically, we feed attributes including stay points, gyration radius, and region entropy [25] into the random forest classifier. To provide a benchmark for the accuracy evaluation of different labels, for label category i, we normalize the F1-score F_1^i of the generated trajectories based on $\hat{F}_1^i$, which is obtained by feeding the real-world testing dataset into the classifier. We define this result as the controllable generation success rate, which is formulated as follows, given a profile with M categories:

$$score^i = \frac{F_1^i}{\hat{F}_1^i}, F_1 = \frac{1}{M}\sum_{m=1}^{M}\frac{2 \cdot TP_m}{2 \cdot TP_m + FP_m + FN_m}, \tag{18}$$

where TP_m, FP_m, and FN_m represent the true positive rate (TP), false positive rate (FP), and false negative rate (FN) of the m-classification task, respectively. Based on the above definition, the control effect of different label categories can be compared. The higher the controllable generation success rate, the better the control effect of the generation model.

Trajectory Quality. We calculate the **Jensen–Shannon (JS)** divergence for a quantitative assessment of the generated trajectory quality. This divergence metric gauges the dissimilarity between two probability distributions, serving as a measure of similarity between generated trajectories and

real-world counterparts with respect to mobility patterns. The lower the value of JS divergence, the closer the distribution of generated trajectories and real-world counterparts.

Specifically, we will quantitatively calculate the JS divergence of generated trajectories and real-world trajectories in the following five mobility patterns [10, 27]:

Travel distance: the accumulated travel distance of each user per day.
Gyration radius: the spatial range of user activities per day.
Duration: the stay time visiting each location.
DailyLoc: the daily count of visited locations.
I-rank: the frequency of individual visits to each location, where only the 100 most frequently visited locations are considered.

In our experiment, a low JS divergence value indicates a strong similarity between the generated trajectory and the real-world trajectories in terms of mobility patterns.

4.1.4 Implementation Detail. We implement all of the algorithms of our method in Python. We conduct the experiments on a workstation that has a 2.60 GHz Platinum 8358 CPU (equipped with 32 cores and 64 threads) from Intel, 128 G memory, four 4,090 GPU cards, and two 3,090 GPU cards from Nvidia.

Our diffusion model utilizes a Transformer architecture with 3 million parameters, with the fixed trajectory sequence length $n = 168$, the number of diffusion steps $K = 2{,}000$, and a square-root noise schedule [20] with the starting noise level $s = 0.0001$. The embedding dimension d is also a hyperparameter. We select from $d \in \{312, 396, 516, 768\}$ and select $d = 396$ for both Mobile and Tencent datasets. For Transformer layers, we select from $N \in \{4, 8, 12\}$ and select $N = 8$ for the Mobile dataset and $N = 12$ for the Tencent dataset. For attention heads, we select from $h \in \{4, 8\}$ and select $h = 4$ for both datasets. Our attribute classifier is also based on a Transformer architecture. We select the embedding dimension $d' = 396$, Transformer layers $N' = 12$ and attention heads $h' = 12$ for both Mobile and Tencent datasets.

We train our diffusion model using the Adam optimizer with an initial learning rate of 1e-4, linear decay, a dropout rate of 0.1, and a batch size of 128. We train the classifiers using the Adam optimizer with an initial learning rate, chosen from 0.001, 0.0001, and 0.00001, with linear decay, a dropout rate of 0.1, and a batch size of 128. Training is performed on a single GPU, either NVIDIA GeForce RTX 3090 or NVIDIA GeForce RTX 4090.

4.2 Overall Performance

In this part, we perform model comparison in terms of controllable generation success rate and trajectory quality.

4.2.1 Controllable Generation Success Rate. Table 4 respectively shows the controllable generation success rates of various methods on income, gender, education, and age conditions on Mobile and Tencent datasets. Based on these results, we can derive the following conclusions:

— *Our method delivers the best results.* Our method significantly improves the controllable generation success rate by over 10% on most conditions on the Mobile dataset. Notably, our method improves the metrics by over 40% on education and age conditions. For the Tencent dataset, our method also improves all metrics by over 10%.
— *Autoregressive deep generative models perform poorly in capturing statistical features.* The relationship between user profiles and trajectories is not reflected in any specific location but in many statistical metrics, such as accumulated travel distance. Autoregressive deep generative models such as VAE, GAN, and GAIL perform poorly on all metrics. However,

Table 4. Overall Performance of Conditional Control Success Rate

Dataset	Mobile				Tencent			
Model	Income	Gender	Education	Age	Income	Gender	Education	Age
Timegeo	0.70	<u>0.84</u>	<u>0.68</u>	<u>0.61</u>	<u>0.69</u>	<u>0.68</u>	<u>0.66</u>	<u>0.38</u>
CVAE	<u>0.72</u>	0.79	0.52	0.35	0.42	0.56	0.40	0.35
MoveSim	0.36	0.75	0.20	0.24	0.16	0.48	0.15	0.18
PateGail	0.50	0.81	0.49	0.55	0.50	0.48	0.42	0.18
Ours	**0.77**	**0.94**	**0.96**	**0.94**	**0.77**	**0.79**	**0.77**	**0.48**
Improvement (%)	6.9	11.9	41.2	54.1	11.6	16.1	16.7	26.3

Bold text denotes the optimal result, while underlined text indicates the second best.

our method generates trajectories in a non-autoregressive way, which is more suitable for capturing statistical relationships.

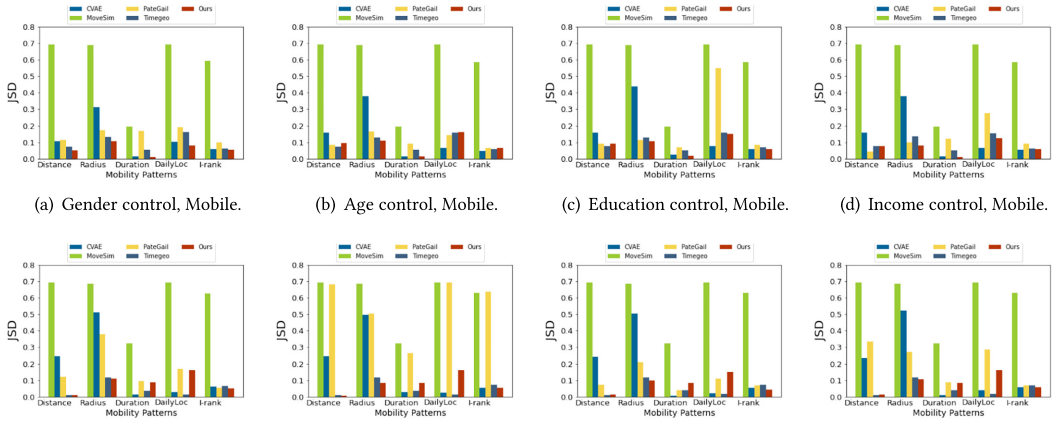
— *Classifier guidance is more sufficient in controllable generation.* Our method outperforms other deep generative models on all metrics. Deep generative models such as VAE, GAN, and GAIL achieve controllable generation by concatenating the latent variables with the embeddings of given conditions in the latent space, which is not powerful enough for a difficult task like controlling user profiles. Meanwhile, the diffusion model can introduce conditions in each diffusion step and use the implicit knowledge learned by the classifier to generate samples through gradient guidance in multiple iterations forcibly. We can observe that classifier guidance is a sufficient mechanism for user profile controllable trajectory generation.

— *User profile controllable trajectory generation is challenging for deep generative models.* The Timegeo model, a mechanistic modeling framework, ranks second on most metrics. It indicates that user profile controllable trajectory generation is still a very difficult task for deep generative models. However, our method is the best-performed deep generative model and beats the Timegeo model on all metrics.

4.2.2 Quality of Generated Trajectories. We previously demonstrated that our method performs well in controllable generation success rate. In this part, we will compare the performance of trajectory quality by computing the JS divergence of different models on related spatiotemporal mobility patterns.

Figure 5(a)–(d) presents the control experiments of gender, age, education, and income profiles on the Mobile dataset, respectively. Similarly, in Figure 5(e)–(h), we illustrate the outcomes for the Tencent dataset. Generally, our method surpasses the baselines on mobility patterns in most or all metrics while maintaining a high success rate in controllable generation. For some metrics where it ranks second, our method still achieves comparable performance to the best baseline. Among the baselines, CVAE tends to capture temporal information but overlooks spatial information; contrarily, PateGail captures spatial information but overlooks time information. MoveSim performs poorly on almost all metrics due to the instability of GAN's training. Timegeo is not a deep generative model, so to compare experimental results, we input user trajectories that meet the given profile conditions into the Timegeo model to generate synthetic trajectories. This approach should only be considered as a reference for experimental results and does not have practical application value.

By observing the results of spatial metrics such as travel distance and gyration radius, we can conclude that the performance of our method is the best, suggesting a highly accurate capture of spatial features. PateGail ranks second in performance, likely due to its EPR mechanism [16], which allows it to effectively capture spatial information. MoveSim performs worst on most metrics,



(a) Gender control, Mobile. (b) Age control, Mobile. (c) Education control, Mobile. (d) Income control, Mobile.

(e) Gender control, Tencent. (f) Age control, Tencent. (g) Education control, Tencent. (h) Income control, Tencent.

Fig. 5. Trajectory quality performance on two datasets.

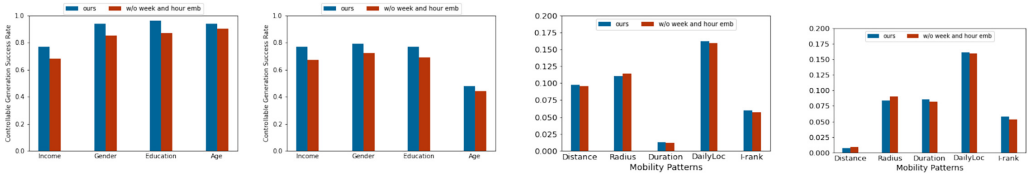
which could be attributed to the unstable training process, complex tuning, and demanding pre-training of GAN, making it challenging to achieve good training results. CVAE overlooks the spatial significance of trajectory points' geographic locations, and thus, its performance on spatial metrics is poor.

Observing the temporal metrics, such as duration and daily locations, we can conclude that our method consistently ranks first or second, demonstrating its ability to accurately capture temporal relations. CVAE also performs well on these metrics, nearly achieving top results on the Tencent dataset. Furthermore, our method consistently outperforms others on the I-rank metric, with CVAE following closely behind. The I-rank reflects users' preferences and habits, aligning with our earlier observation that CVAE tends to excel in capturing temporal relations.

4.3 Ablation Study

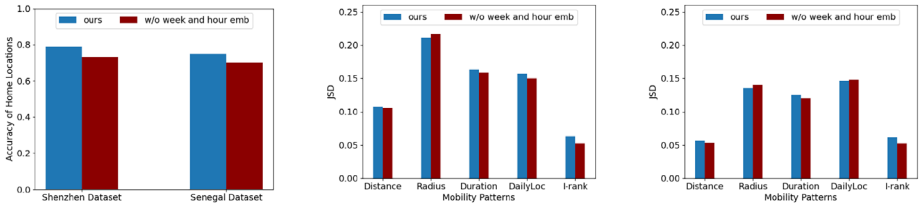
Context Embedding. To enhance the quality of generated trajectories, we employ context embedding instead of positional encoding in the structure of the user profile classifier, which combines the embedding of hour and week information. We test the effect of this modification on age label on Tencent and Mobile datasets, and home label on Shenzhen and Senegal datasets. The outcomes are presented in Figures 6 and 7. Figure 6(a) and (b) clearly demonstrate that the performance of controllable generation success rate increases by over 10% with the application of context embedding. Meanwhile, Figure 7(a) demonstrates that the performance of accuracy of home locations also increases with the application of context embedding. We can conclude that context embedding is of great use for learning human mobility patterns, and thus, contributing to classifying user profiles. Furthermore, as depicted in Figure 6(c) and (d), the trajectory quality isn't influenced much, since it's primarily determined by the diffusion model rather than the classifier. Notably, in order to eliminate the randomness introduced by different experiments, we conduct the experiments five times and take the average of the results.

Uncontrollable Generation. We calculate the relative increase in spatial temporal metrics in the controllable generation setting compared to uncontrollable generation. Notably, these metrics are negatively correlated with the generative ability of the models. Table 5 reveals that our method's performance remains largely unaffected by the control mechanism.



(a) Success rate, Mobile. (b) Success rate, Tencent. (c) Trajectory quality, Mobile. (d) Trajectory quality, Tencent.

Fig. 6. Ablation study on context embedding in terms of controllable generation success rate ((a)–(b)) and trajectory quality ((c)–(d)) on two datasets.



(a) Accuracy of home locations on Shenzhen and Senegal datasets. (b) Trajectory quality, Shenzhen. (c) Trajectory quality, Senegal.

Fig. 7. Ablation study on context embedding in terms of accuracy of home locations (a) and trajectory quality ((b)–(c)) on Shenzhen and Senegal dataset.

Table 5. Performance Difference between Controllable Generation Setting and Uncontrollable Counterpart, Where a Zero Value Indicates Equal Performance and a Negative Value Indicates Better Performance

Dataset	Mobile					Tencent				
Model	Distance	Radius	Duration	DailyLoc	I-rank	Distance	Radius	Duration	DailyLoc	I-rank
CVAE	0.0316	0.0131	**0.0002**	0.0921	0.0042	0.0003	0.0575	0.1619	**−0.0609**	0.0224
MoveSim	0.1176	0.0111	0.2898	0.381	0.526	0.0005	−0.0138	0.0789	0.0036	0.0338
PateGail	0.0078	0.1480	0.0157	0.3522	0.0212	0.0883	0.1419	0.1222	0.3087	0.0331
Ours	**−0.0309**	**−0.2048**	0.0088	**0.0411**	**0.0039**	**−0.0188**	**−0.0314**	**0.0776**	0.1405	**−0.0087**

Bold text denotes the optimal result, while underlined text indicates the second best.

4.4 Compositional Control

Due to the modularity of our controllable generation framework, given classifiers for multiple independent attribute control tasks, we can achieve multiple controls by adding gradients of different classifiers together. For example, to achieve control on both gender and age attributes, the gradient of controllable generation would be the sum of the gradients of three factors, which are unconditional generation, the trained gender classifier, and the age classifier.

Tables 6(a) and (b) demonstrates the complete results of different control combinations of our model. For every single row, compared to the data on the diagonal, the data not on the diagonal does not degrade significantly, and the decrease is mostly within 50%. We can conclude that compositional control is feasible while the control effectiveness is maintained. To compare the ability of compositional control of different models, we demonstrate an example of controlling gender and education conditions on two datasets in Table 7. Our model shows its ability to achieve compositional control. We can observe that our model achieves a high success rate for both control settings, and our method exhibits a performance improvement of at least 20% compared to the

Table 6. Compositional Control Experiment on Two Datasets

(a) Mobile Dataset

Main Attribute \ Auxiliary Attribute	Income	Gender	Education	Age
Income	0.77	0.51	0.45	0.42
Gender	0.64	0.94	0.58	0.61
Education	0.58	0.53	0.96	0.61
Age	0.51	0.55	0.57	0.94

(b) Tencent Dataset

Main Attribute \ Auxiliary Attribute	Income	Gender	Education	Age
Income	0.77	0.45	0.47	0.51
Gender	0.51	0.79	0.49	0.43
Education	0.48	0.45	0.77	0.47
Age	0.31	0.34	0.35	0.48

Each data represents the conditional control success rate of the main attribute when both the main and auxiliary attributes are controlled simultaneously. Notably, the data on the diagonal indicate the result of single attribute control.

Table 7. Performance Comparison with Baseline Methods in Compositional Control Experiment on the Mobile Dataset (Gender and Education Attributes Are Controlled)

Dataset	Mobile		Tencent	
Model	Gender	Education	Gender	Education
CVAE	0.31	<u>0.33</u>	<u>0.24</u>	0.25
MoveSim	0.32	0.10	0.19	0.11
PateGail	<u>0.47</u>	0.31	0.18	<u>0.28</u>
Ours	**0.58**	**0.53**	**0.49**	**0.45**

Bold text denotes the optimal result, while underlined text indicates the second best result.

baseline. Thus, our model effectively accomplishes compositional control due to its modular model structure. In contrast, the baselines fail to do so, possibly due to the inadequacy of adding conditions in the latent space to simultaneously capture two statistical features.

4.5 Applications Experiment

To further prove the high application value of our framework in various downstream needs, we design two application experiments, generative home-based travels and augmenting imbalanced datasets.

Table 8. Trajectory Quality and Accuracy Results of Generating Home-Based Travels on Four Real-World Datasets

(a) Mobile Dataset

Model	Distance	Gyration	Duration	DailyLoc	I-rank	Accuracy
CVAE	0.1970	0.5046	0.0138	0.1692	0.0693	0.54
MoveSim	0.6932	0.6932	0.2155	0.6932	0.6116	0.11
PateGail	0.1909	0.2021	0.1845	0.5428	0.0791	0.38
Ours	0.1288	0.1966	0.0128	0.2233	0.0501	0.82

(b) Tencent Dataset

Model	Distance	Gyration	Duration	DailyLoc	I-rank	Accuracy
CVAE	0.1400	0.4300	0.1766	0.1692	0.0693	0.62
MoveSim	0.6932	0.6932	0.2953	0.6932	0.6095	0.15
PateGail	0.2107	0.2378	0.2186	0.4790	0.0865	0.42
Ours	0.0310	0.1012	0.1191	0.1356	0.0633	0.86

(c) Shenzhen Dataset

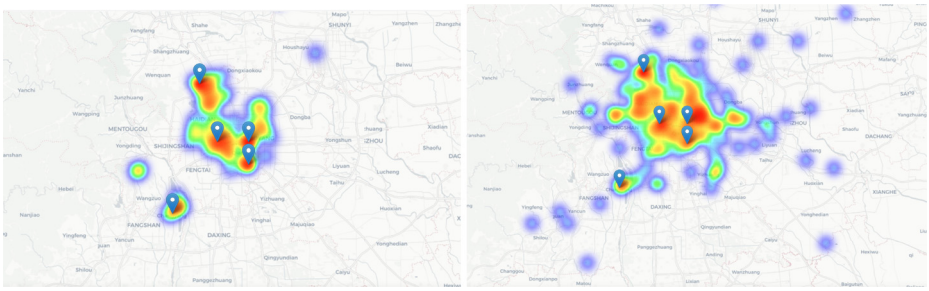
Model	Distance	Gyration	Duration	DailyLoc	I-rank	Accuracy
CVAE	0.2103	0.4744	0.2076	0.1721	0.0754	0.49
MoveSim	0.6932	0.6931	0.2727	0.6932	0.5975	0.11
PateGail	0.2051	0.2256	0.1852	0.5086	0.0812	0.35
Ours	0.1074	0.2106	0.1634	0.1576	0.0624	0.79

(d) Senegal Dataset

Model	Distance	Gyration	Duration	DailyLoc	I-rank	Accuracy
CVAE	0.2063	0.5362	0.1589	0.1610	0.0752	0.47
MoveSim	0.6852	0.6901	0.2853	0.6725	0.5927	0.10
PateGail	0.1957	0.2643	0.2025	0.5373	0.0961	0.33
Ours	0.0566	0.1356	0.1251	0.1465	0.0612	0.75

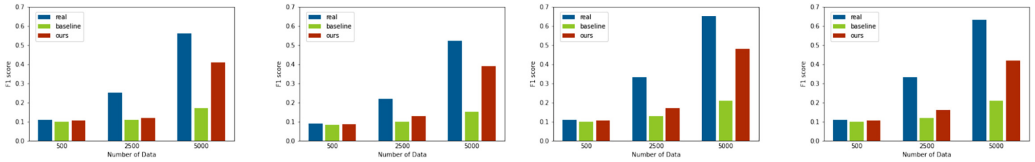
4.5.1 Generating Home-Based Travels. Home is a crucial dwelling location in human mobility, and controlling its location poses a more challenging and intricate task. On one hand, the location of homes can be on any spatial grid point, leading to a vast candidate space of possible classification results, making the task difficult. On the other hand, our main controllable generation tasks involve exploring statistical features, whereas determining the location of a home involves mining spatio-temporal semantics, making the task even more intricate.

In this experiment, we select the top 10 most frequently occurring residential areas in the datasets as the designated home locations. For each home location, we generate 100 trajectories of seven days based on our method and the baselines. When evaluating the results, we use the algorithm of Timegeo [16] to refer homes of generated trajectories. The ground truth data for evaluation is a set of trajectories from the training dataset, with these 10 designated locations as home positions. Table 8(a) demonstrates that our model attains the highest performance in both trajectory quality and home control accuracy on all four datasets. Meanwhile, our model performs exceptionally well on datasets from different cities and countries, indicating that our approach has strong adaptability



(a) Ours (b) CVAE

Fig. 8. The spatial distribution of trajectory points of (a) our model and (b) the best baseline model (CVAE) in Mobile dataset.



(a) Age, Mobile Dataset. (b) Income, Mobile Dataset. (c) Age, Tencent Dataset. (d) Income, Tencent Dataset.

Fig. 9. Classification performance in the synthetic scenario on (a) and (b) Mobile and (c) and (d) Tencent dataset, taking age and income labels as an example.

and generalizability. In particular, Figure 8 visualizes the spatial distribution of trajectory points of our model and the top-performing baseline model on the Mobile dataset, where the blue markers mark the five given locations of homes. It can be observed that our model can not only achieve higher control accuracy, but also generate more reasonable and realistic trajectories. On one hand, trajectory points generated by our model are more obviously centered on the location of home compared to the baseline. On the other hand, trajectory points generated by our model are more concentrated, thus complying with human mobility patterns in the real world, while those generated by the baseline model are too scattered to be realistic. This is also shown in Table 8(a), where our model performs better than CVAE in the metric of travel distance and gyration radius.

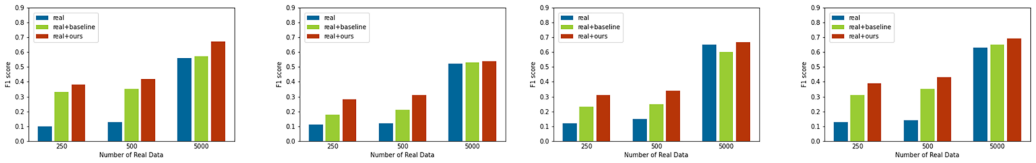
4.5.2 The Effectiveness of the Synthetic Trajectories. Due to privacy concerns, real-world trajectories and user profiles cannot be directly shared with downstream applications. Instead, we can generate synthetic trajectories to protect sensitive information while retaining access to real-world data. To assess the effectiveness of our synthetic trajectories in preserving correct user profile information, we design an application experiment in which we train a random forest-based profile classifier using the synthetic trajectories generated by our proposed approach and test its performance on the original testing dataset.

Notably, our experiments involve synthetic data across two categories:

Synthetic Scenario. Exclusively utilizing synthetic trajectories for enhanced privacy protection.

Hybrid Scenario. It combines both real and synthetic trajectories, aligning with common data augmentation practices.

The synthetic scenario is used to assess the retained utility of the generated data by assessing whether its performance closely aligns with that of real-world data. Additionally, the hybrid scenario is used to test whether the dataset enhanced through data augmentation can train more robust classifiers.



(a) Age, Mobile Dataset. (b) Income, Mobile Dataset. (c) Age, Tencent Dataset. (d) Income, Tencent Dataset.

Fig. 10. Classification performance in the hybrid scenario on (a) Mobile and (b) Tencent dataset, using age and income labels as examples. We consider different quantities of real-world trajectories, namely 250, 500, 5,000, to which we augment with an additional 5,000 generated sequences.

Figure 9 illustrates that, in comparison to the best baseline, the classification performance on the dataset generated by our model closely approaches that of the real-world dataset. Furthermore, Figure 10 highlights the superiority of the random forest classifier trained on the augmented dataset, outperforming the one trained solely on the real-world dataset. Notably, the data augmented by our model surpasses that of the best baseline. It's worth noting that data augmentation proves particularly beneficial when dealing with small-scale real-world datasets, such as those with only 250 trajectories. All the results above demonstrate the practicality of synthetic data.

5 Related Work

5.1 Trajectory Generation Method

With the growing prominence of computer science and AI, recent years have witnessed the emergence of various approaches in trajectory generation. These approaches can be broadly categorized into two groups: probability-based mathematical modeling methods and deep generative model-based techniques. Mathematical modeling methods [15, 16] operate on the premise that human mobility can be effectively represented through a finite set of parameters with clear physical significance. Though widely used in the real world, they also have many limitations. Firstly, they may overly simplify human mobility patterns and ignore complex patterns such as higher-order features and so on, thus lacking flexibility. Additionally, these methods cannot achieve controllable generation from the perspective of mathematical principles.

In terms of deep generative model-based methods, most deep learning methods combined with trajectory generation adopt the structure of GAN [5, 10, 31, 50], a few others adopt VAE [14, 60] and GAIL [46]. Moreover, there are methods [17, 64, 65] used for generating continuous trajectories that comply with real-world road network at a finer granularity, but these tasks are different from the problem studied in this article. Notably, there is currently no relevant work on trajectory generation that can control user profiles, so our work had made groundbreaking progress in addressing this issue.

5.2 Diffusion Model

The concept and foundational framework of diffusion model are initially introduced by Jascha et al. [39], and is perfected by the DDPM method [13], which is also the most mainstream framework currently. High-quality samples are generated by simulating the forward diffusion and backward denoising process of data samples. For the slow sampling speed of DDPM, Song et al. [41] propose the DDIM framework, extending DDPM through a non-Markov diffusion process which can correspond to a deterministic generation process. Additionally, in response to the poor maximum likelihood estimation of diffusion models, Nichol et al. [29] propose a new noise mechanism and introduce an alternative of learning variance in the denoising process.

Over the past few years, diffusion models have demonstrated exceptional performance across diverse fields. Diffusion models can handle continuous or other high-dimensional data such as images [28, 30, 35], spatio-temporal flows [61, 63] and neural network parameters [56] and discrete data such as natural language sequences [20]. In the latent domain, Rombach et al. [32] propose latent diffusion model, which employs the diffusion model within the latent space of highly capable pre-trained autoencoders. Consequently, there is relatively few diffusion model framework that can handle discrete spatial-temporal trajectories in the continuous domain [7, 55, 65]. Different from these works that directly design the diffusion process on the spatiotemporal data space, to bridge this gap, our proposed framework redesigns the structure of diffusion model by adding the process of mapping and embedding, enabling the framework to tackle discrete spatial-temporal data in the continuous domain.

5.3 Controllable Generation Framework of Diffusion Models

The existing controllable generation frameworks can generally be divided into three categories. The first commonly used method is to concatenate the embeddings of given conditions to the latent variables [6, 37, 66]. Inspired by Bayes' formula, the second approach, referred to as classifier-guidance [8, 20, 24, 62], leverages classifier gradients to condition the pre-trained diffusion model and guide the sampling process. The third approach, classifier-free guidance, involves training both a conditional diffusion model and an unconditional one simultaneously. Notably, some new frameworks have been proposed to handle different modalities of conditions [45, 59]. However, they are not directly relevant to the problem in our study and thus falls outside the scope of consideration.

6 Conclusion and Discussion

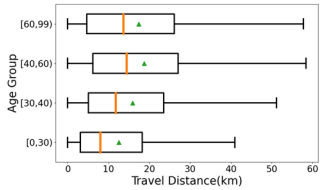
6.1 Conclusion

In this article, we address the challenge of trajectory generation with controllable user profiles by proposing a controllable generation framework that leverages a continuous diffusion model and classifier guidance. The proposed non-autoregressive framework can well capture statistical features and thus can model complex relationships between trajectories and user profiles. Our extensive experiments on two real-world datasets underscore the robustness and versatility of our proposed framework. It successfully strikes a balance between generating high-quality trajectories that adhere to human mobility patterns and providing precise control over user profiles. Notably, our framework excels in various applications, including the generation of home-based travels and dataset augmentation, surpassing the baselines.

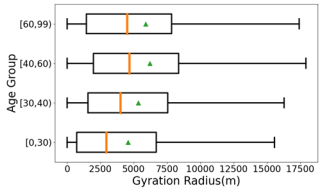
6.2 Discussion

6.2.1 Implicit Attributes. The data may implicitly contain relationships between other hidden attributes and trajectories. For example, if users' age and gender are considered features, but the trajectories are primarily related to their race, which is not included in the user profiles in the data. The modeling process will incorporate the trajectory characteristics of different races. Consequently, when generating samples, the "race" feature will also be randomly sampled. However, this implicit relationship cannot be explicitly controlled in the implementation.

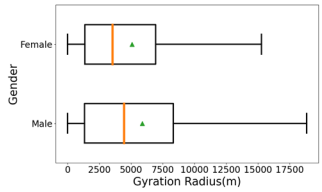
6.2.2 Unbalanced Profiles. If the dataset contains highly imbalanced profile features, training the classifier becomes a significant challenge, as the results of minority class may be poor. Consequently, controllable generation also encounters difficulties, necessitating the incorporation of unbalanced learning techniques such as KNN-based methods [47], synthetic minority over-sampling techniques



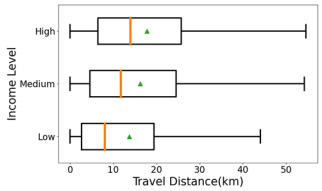
(a) Travel distance distribution of different age groups, Tencent.



(b) Gyration radius distribution of different age groups, Tencent.



(c) Gyration radius distribution of different genders, Tencent.



(d) Travel distance distribution of different income levels, Tencent.

Fig. 11. The distribution of travel distance ((a) and (d)) and gyration radius ((b) and (c)) on Tencent datasets.

[4, 48], other under-sampling methods [22, 42], and so on to enhance the performance of guided generation.

6.2.3 Scalability and Potential Optimizations. Our model exhibits stronger control over gender and education attributes, but weaker control over age. This may be due to a less pronounced connection between age groups and trajectory features compared to other profiles. Effective controllable trajectory generation depends on the strength of the relationship between attributes and trajectories, with stronger connections yielding better control. As is shown in Figure 11(a)–(c), the influence of different age groups on user trajectories is not as significant as that of income levels, especially since the trajectory characteristics of users aged 40–60 and those aged 60–90 are very similar, which may pose greater challenges for classification and guided generation tasks. Furthermore, it could be due to gender classification involving fewer categories, making the task relatively simpler. As is shown in Figure 11(c), the influence of different genders on user trajectory characteristics is quite significant as well.

In terms of limitations, our model's performance on sparse categories needs improvement. Additionally, we intend to incorporate semantic information, such as the functions of locations, into our framework to better capture the relationship between trajectories and user profiles. A recent work proposed to leverage the powerfulness of **Large Language Models (LLMs)** to generate human mobility trajectories in a controllable manner [36], which can be potentially helpful as the world knowledge stored in LLMs can alleviate the data scarcity issue in this problem.

References

[1] N. Anand and Tudor Achim. 2022. Protein structure and sequence generation with equivariant denoising diffusion probabilistic models. arXiv:2205.15019. Retrieved from https://arxiv.org/abs/2205.15019

[2] Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. 2021. Structured denoising diffusion models in discrete state-spaces. arXiv:2107.03006. Retrieved from https://arxiv.org/abs/2107.03006

[3] Maya Benarous, Eran Toch, and Irad Ben-Gal. 2022. Synthesis of longitudinal human location sequences: Balancing utility and privacy. *ACM Transactions on Knowledge Discovery from Data* 16 (2022), 1–27.

[4] Chumphol Bunkhumpornpat, Krung Sinapiromsaran, and Chidchanok Lursinsap. 2012. DBSMOTE: Density-based synthetic minority over-sampling technique. *Applied Intelligence* 36 (2012), 664–684.

[5] Chu Cao and Mo Li. 2021. Generating mobility trajectories with retained data utility. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*.

[6] Haoye Chai, Tong Li, Fenyu Jiang, Shiyuan Zhang, and Yong Li. 2024. Knowledge guided conditional diffusion model for controllable mobile traffic generation. In *Proceedings of the Companion Proceedings of the ACM on Web Conference 2024*, 851–854.

[7] Hongyi Chen, Jingtao Ding, Yong Li, Yue Wang, and Xiao-Ping Zhang. 2024. Social physics informed diffusion model for crowd simulation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38, 474–482.

[8] Prafulla Dhariwal and Alex Nichol. 2021. Diffusion models beat GANs on image synthesis. arXiv:2105.05233. Retrieved from https://arxiv.org./abs/2105.05233

[9] Jie Feng, Yong Li, Chao Zhang, Funing Sun, Fanchao Meng, Ang Guo, and Depeng Jin. 2018. Deepmove: Predicting human mobility with attentional recurrent networks. In *Proceedings of the 2018 World Wide Web Conference*, 1459–1468.

[10] J. Feng, Zeyu Yang, Fengli Xu, Haisu Yu, Mudan Wang, and Yong Li. 2020. Learning to simulate human mobility. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*.

[11] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Proceedings of the Advances in Neural Information Processing Systems,* Vol. 27.

[12] Jonathan Ho and Stefano Ermon. 2016. Generative adversarial imitation learning. In *Proceedings of the Advances in Neural Information Processing Systems,* Vol. 29.

[13] Jonathan Ho, Ajay Jain, and P. Abbeel. 2020. Denoising diffusion probabilistic models. arXiv:2006.11239.

[14] Dou Huang, Xuan Song, Zipei Fan, Renhe Jiang, Ryosuke Shibasaki, Yu Zhang, Haizhong Wang, and Yugo Kato. 2019. A variational autoencoder based generative model of urban human mobility. In *Proceedings of the IEEE Conference on Multimedia Information Processing and Retrieval (MIPR '19)*, 425–430.

[15] Sibren Isaacman, Richard A. Becker, Ramón Cáceres, Margaret Martonosi, James R. Rowland, Alexander E. Varshavsky, and Walter Willinger. 2012. Human mobility modeling at metropolitan scales. In *Proceedings of the ACM SIGMOBILE International Conference on Mobile Systems, Applications, and Services*.

[16] Shan Jiang, Yingxiang Yang, Siddharth Gupta, Daniele Veneziano, Shounak Athavale, and Marta C. González. 2016. The TimeGeo modeling framework for urban mobility without travel surveys. *Proceedings of the National Academy of Sciences* 113 (2016), E5370–E5378.

[17] Wenjun Jiang, Wayne Xin Zhao, Jingyuan Wang, and Jiawei Jiang. 2023. Continuous trajectory generation based on two-stage GAN. arXiv:2301.07103. Retrieved from https://arxiv.org/abs/2301.07103

[18] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. arXiv:abs/1412.6980. Retrieved from https://arxiv.org/abs/1412.6980

[19] Diederik P. Kingma and Max Welling. 2013. Auto-encoding variational bayes. arXiv:1312.6114. Retrieved from https://arxiv.org/abs/1312.6114

[20] Xiang Li, John Thickstun, Ishaan Gulrajani, Percy S. Liang, and Tatsunori B. Hashimoto. 2022. Diffusion-lm improves controllable text generation. In *Proceedings of the Advances in Neural Information Processing Systems,* Vol. 35, 4328–4343.

[21] Yuxuan Liang, Kun Ouyang, Yiwei Wang, Xu Liu, Hongyang Chen, Junbo Zhang, Yu Zheng, and Roger Zimmermann. 2022. TrajFormer: Efficient trajectory classification with transformers. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*.

[22] Wei-Chao Lin, Chih-Fong Tsai, Ya-Han Hu, and Jing-Shang Jhang. 2017. Clustering-based undersampling in class-imbalanced data. *Information Sciences* 409 (2017), 17–26.

[23] Zongyu Lin, Shiqing Lyu, Hancheng Cao, Fengli Xu, Yuqiong Wei, Hanan Samet, and Yong Li. 2020. HealthWalks: Sensing fine-grained individual health condition via mobility data. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4 (2020), Article 138, 1–26. Retrieved from https://api.semanticscholar.org/CorpusID:229320914

[24] Chang Liu, Jingtao Ding, Yiwen Song, and Yong Li. 2024. TDNetGen: Empowering complex network resilience prediction with generative augmentation of topology and dynamics. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 1875–1886.

[25] Yu Liu, Zhilun Zhou, Yong Li, and Depeng Jin. 2024. Urban knowledge graph aided mobile user profiling. *ACM Transactions on Knowledge Discovery from Data* 18, 1 (2024), 1–30.

[26] Qingyue Long, Huandong Wang, Tong Li, Lisi Huang, Kun Wang, Qiong Wu, Guangyu Li, Yanping Liang, Li Yu, and Yong Li. 2023. Practical synthetic human trajectories generation based on variational point processes. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. Retrieved from https://api.semanticscholar.org/CorpusID:260499642

[27] Massimiliano Luca, Gianni Barlacchi, Bruno Lepri, and Luca Pappalardo. 2020. A survey on deep learning for human mobility. *ACM Computing Surveys (CSUR)* 55 (2020), 1–44.

[28] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. 2021. GLIDE: Towards photorealistic image generation and editing with text-guided diffusion models. In *Proceedings of the International Conference on Machine Learning*.

[29] Alexander Quinn Nichol and Prafulla Dhariwal. 2021. Improved denoising diffusion probabilistic models. In *Proceedings of the International Conference on Machine Learning*. PMLR, 8162–8171.

[30] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. 2022. Hierarchical text-conditional image generation with CLIP latents. arXiv:2204.06125. Retrieved from https://arxiv.org/abs/2204.06125

[31] Jinmeng Rao, Song Gao, Yuhao Kang, and Qunying Huang. 2020. LSTM-TrajGAN: A deep learning approach to trajectory privacy protection. arXiv:2006.10521. Retrieved from https://arxiv.org/abs/2006.10521

[32] Robin Rombach, A. Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2021. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR '22)*, 10674–10685.

[33] Can Rong, Jingtao Ding, and Yong Li. 2023. An interdisciplinary survey on origin-destination flows modeling: Theory and techniques. arXiv:2306.10048. Retrieved from https://arxiv.org/abs/2306.10048

[34] Can Rong, Zhicheng Liu, Jingtao Ding, and Yong Li. 2024. Learning to generate temporal origin-destination flow based-on urban regional features and traffic information. *ACM Transactions on Knowledge Discovery from Data* 18, 6 (2024), 1–17.

[35] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L. Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, Seyedeh Sara Mahdavi, Raphael Gontijo Lopes, Tim Salimans, Jonathan Ho, David J. Fleet, and Mohammad Norouzi. 2022. Photorealistic text-to-image diffusion models with deep language understanding. arXiv:2205.11487. Retrieved from https://arxiv.org/abs/2205.11487

[36] Chenyang Shao, Fengli Xu, Bingbing Fan, Jingtao Ding, Yuan Yuan, Meng Wang, and Yong Li. 2024. Beyond imitation: Generating human mobility from context-aware reasoning with large language models. arXiv:2402.09836. Retrieved from https://arxiv.org/abs/2402.09836

[37] Huajie Shao, Jun Wang, Haohong Lin, Xuezhou Zhang, Aston Zhang, Heng Ji, and Tarek F. Abdelzaher. 2021. Controllable and diverse text generation in e-commerce. In *Proceedings of the Web Conference 2021*.

[38] Huajie Shao, Shuochao Yao, Dachun Sun, Aston Zhang, Shengzhong Liu, Dongxin Liu, Jun Wang, and Tarek F. Abdelzaher. 2020. ControlVAE: Controllable variational autoencoder. In *Proceedings of the International Conference on Machine Learning*.

[39] Jascha Narain Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. 2015. Deep unsupervised learning using nonequilibrium thermodynamics. arXiv:1503.03585.Retrieved from https://arxiv.org/abs/1503.03585

[40] Chaoming Song, Tal Koren, Pu Wang, and Albert-László Barabási. 2010. Modelling the scaling properties of human mobility. *Nature Physics* 6, 10 (2010), 818–823.

[41] Jiaming Song, Chenlin Meng, and Stefano Ermon. 2020. Denoising diffusion implicit models. arXiv:2010.02502. Retrieved from https://arxiv.org/abs/2010.02502

[42] Xinmin Tao, Qing Li, Wenjie Guo, Chao Ren, Qing He, Rui Liu, and JunRong Zou. 2020. Adaptive weighted over-sampling for imbalanced datasets based on density peaks clustering with heuristic filtering. *Information Sciences* 519 (2020), 43–73.

[43] Guy Tevet, Sigal Raab, Brian Gordon, Yonatan Shafir, Daniel Cohen-Or, and Amit H Bermano. 2022. Human motion diffusion model. arXiv:2209.14916. Retrieved from https://arxiv.org/abs/2209.14916

[44] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the Advances in Neural Information Processing Systems*, Vol. 30.

[45] Andrey Voynov, Kfir Aberman, and Daniel Cohen-Or. 2022. Sketch-guided text-to-image diffusion models. arXiv:2211.13752. Retrieved from https://arxiv.org/abs/2211.13752

[46] Huandong Wang, Changzheng Gao, Yuchen Wu, Depeng Jin, Lina Yao, and Yong Li. 2023. PateGail: A privacy-preserving mobility trajectory generator with imitation learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37, 14539–14547.

[47] Le Wang, Meng Han, Xiaojuan Li, Ni Zhang, and Haodong Cheng. 2021. Review of classification methods on unbalanced data sets. *IEEE Access* 9 (2021), 64606–64628.

[48] Shuo Wang and Xin Yao. 2009. Diversity analysis on imbalanced data sets by using ensemble models. In *Proceedings of the IEEE Symposium on Computational Intelligence and Data Mining*. IEEE, 324–331.

[49] Lun Wu, Liu Yang, Zhou Huang, Yaoli Wang, Yanwei Chai, Xia Peng, and Yu Liu. 2019. Inferring demographics from human trajectories and geographical context. *Computers, Environment and Urban Systems* 77 (2019), 101368.

[50] Gang Xiong, Zhishuai Li, Meihua Zhao, Yu Zhang, Qinghai Miao, Yisheng Lv, and Fei Wang. 2023. TrajSGAN: A semantic-guiding adversarial network for urban trajectory generation. *IEEE Transactions on Computational Social Systems* 11 (2023), 1733–1743.

[51] Fengli Xu, Zongyu Lin, Tong Xia, Diansheng Guo, and Yong Li. 2020. SUME: Semantic-enhanced urban mobility network embedding for user demographic inference. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4 (2020), Article 98, 1–25.

[52] Sirui Xu, Zhengyuan Li, Yu-Xiong Wang, and Liang-Yan Gui. 2023. Interdiff: Generating 3D human-object interactions with physics-informed diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 14928–14940.

[53] Hao Xue, Flora Salim, Yongli Ren, and Nuria Oliver. 2021. MobTCast: Leveraging auxiliary trajectory forecasting for human mobility prediction. In *Proceedings of the Advances in Neural Information Processing Systems,* Vol. 34, 30380–30391.

[54] Yanwei Yu, Hongjian Wang, and Zhenhui Jessie Li. 2018. Inferring mobility relationship via graph embedding. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2 (2018), 1–21.

[55] Yuan Yuan, Jingtao Ding, Chenyang Shao, Depeng Jin, and Yong Li. 2023a. Spatio-temporal diffusion point processes. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 3173–3184.

[56] Yuan Yuan, Chenyang Shao, Jingtao Ding, Depeng Jin, and Yong Li. 2024. Spatio-temporal few-shot learning via diffusive neural network generation. In *Proceedings of the 12th International Conference on Learning Representations*.

[57] Ye Yuan, Jiaming Song, Umar Iqbal, Arash Vahdat, and Jan Kautz. 2023. Physdiff: Physics-guided human motion diffusion model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 16010–16021.

[58] Yuan Yuan, Huandong Wang, Jingtao Ding, Depeng Jin, and Yong Li. 2023. Learning to simulate daily activities via modeling dynamic human needs. In *Proceedings of the ACM Web Conference 2023*, 906–916.

[59] Lvmin Zhang and Maneesh Agrawala. 2023. Adding conditional control to text-to-image diffusion models. arXiv:2302.05543.

[60] Liming Zhang, Liang Zhao, and Dieter Pfoser. 2022. Factorized deep generative models for end-to-end trajectory generation with spatiotemporal validity constraints. In *Proceedings of the 30th International Conference on Advances in Geographic Information Systems*.

[61] Shiyuan Zhang, Tong Li, Depeng Jin, and Yong Li. 2024. NetDiff: A service-guided hierarchical diffusion model for network flow trace generation. *Proceedings of the ACM on Networking* 2, CoNEXT3 (2024), 1–21.

[62] Ziyuan Zhong, Davis Rempe, Danfei Xu, Yuxiao Chen, Sushant Veer, Tong Che, Baishakhi Ray, and Marco Pavone. 2023. Guided conditional diffusion for controllable traffic simulation. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '23)*. IEEE, 3560–3566.

[63] Zhilun Zhou, Jingtao Ding, Yu Liu, Depeng Jin, and Yong Li. 2023. Towards generative modeling of urban flow through knowledge-enhanced denoising diffusion. In *Proceedings of the 31st ACM International Conference on Advances in Geographic Information Systems*, 1–12.

[64] Qingyan Zhu, Yize Chen, Hao Wang, Zhenyu Zeng, and Hao Liu. 2022. A knowledge-enhanced framework for imitative transportation trajectory generation. In *Proceedings of the IEEE International Conference on Data Mining (ICDM '22)*, 823–832.

[65] Yuanshao Zhu, Yongchao Ye, Shiyao Zhang, Xiangyu Zhao, and James Yu. 2023. DiffTraj: Generating GPS trajectory with diffusion probabilistic model. In *Proceedings of the 37th Conference on Neural Information Processing Systems*.

[66] Yuanshao Zhu, James Jianqiao Yu, Xiangyu Zhao, Qidong Liu, Yongchao Ye, Wei Chen, Zijian Zhang, Xuetao Wei, and Yuxuan Liang. 2024. ControlTraj: Controllable trajectory generation with topology-constrained diffusion model. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 4676–4687.

Received 18 December 2023; revised 28 August 2024; accepted 9 October 2024