

One Fits All: General Mobility Trajectory Modeling via Masked Conditional Diffusion

Qingyue Long, Can Rong, Huandong Wang, Yong Li

Abstract—Trajectory data play a crucial role in many applications, ranging from network optimization to urban planning. Existing studies on trajectory data are task-specific, and their applicability is limited to the specific tasks on which they have been trained, such as generation, recovery, or prediction. However, the potential of a unified model has not yet been fully explored in trajectory modeling. Although various trajectory tasks differ in inputs, outputs, objectives, and conditions, they share common mobility patterns. Based on these common patterns, we can construct a general framework that enables a single model to address different tasks. However, building a trajectory task-general framework faces two critical challenges: 1) the diversity in the formats of different tasks and 2) the complexity of the conditions imposed on different tasks. In this work, we propose a general trajectory modeling framework via masked conditional diffusion (named GenMove). Specifically, we utilize mask conditions to unify diverse formats. To adapt to complex conditions associated with different tasks, we utilize historical trajectory data to obtain contextual trajectory embeddings, which include rich contexts such as spatiotemporal characteristics and user preferences. Integrating the contextual trajectory embedding into diffusion models through a classifier-free guidance approach allows the model to flexibly adjust its outputs based on different conditions. Extensive experiments on mainstream tasks demonstrate that our model significantly outperforms state-of-the-art baselines, with the highest performance improvement exceeding 13% in generation tasks.

Index Terms—Mobility trajectory, generative models, diffusion models.

I. INTRODUCTION

Trajectory data record human movement and reflect the patterns of human activities, supporting various applications, such as urban planning, traffic control, and emergency response [1], [2], [3]. Given the rich information embedded in trajectory data, researchers have developed methods to solve tasks like trajectory generation, recovery, and prediction. Although numerous studies have addressed various trajectory tasks, most existing works still rely on models specifically designed for a single task. For example, in trajectory prediction, researchers typically develop specialized models that use historical trajectory data to predict future locations [4], [5]. In trajectory generation, deep generative models are used to generate synthetic trajectory data that conform to realistic mobility patterns [6], [7]. For trajectory recovery, the common

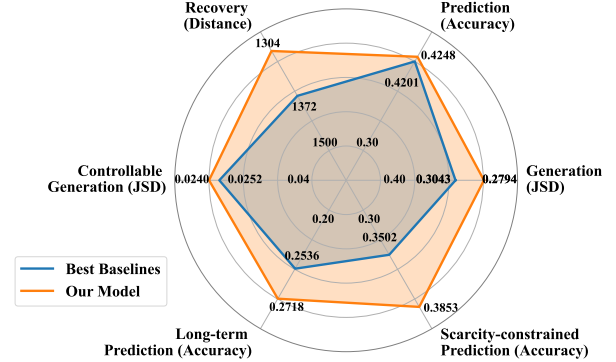


Fig. 1. Performance of GenMove on ISP dataset.

approach involves using statistical interpolation or deep learning techniques to address missing data in trajectories [8], [9]. While these methods excel in specific tasks, they are limited by a lack of flexibility and generality.

In fact, different trajectory tasks essentially rely on common human mobility patterns, such as periodicity, sequentiality, and circadian rhythms. Leveraging these shared patterns across various tasks within the same framework can significantly enhance the performance of each task. On the other hand, different application scenarios have diverse conditions for trajectory tasks. For example, in traffic management, short-term trajectory prediction is needed to optimize real-time traffic flow [10]. In contrast, urban planning relies on long-term trajectory prediction to design the layout of future infrastructure [11]. When limited data is available due to privacy protections, unconditional trajectory generation can provide a substantial amount of generated data [12], [13]. Moreover, controllable trajectory generation is necessary for personalized advertising or travel recommendations to meet user needs precisely [14], [15]. Therefore, it is necessary to develop a general framework that can flexibly adapt to various application scenarios.

There are remarkable achievements of general frameworks in computer vision and natural language processing [16], [17], such as GPT, which can perform various tasks, including text translation, question answering, and image generation from text [18]. Inspired by these developments, we naturally pose a new research question: *Can we address multiple tasks with a general framework for mobility modeling?* However, the task-general mobility modeling framework is still an open problem with the following challenges:

- **Diversity of different task formats.** Different tasks are natu-

Manuscript received January 7, 2025; revised xxxx, 2025.

The authors are with the Department of Electronic Engineering, Beijing National Research Center for Information Science and Technology (BNRist), Tsinghua University, Beijing 100084, China (e-mail: longqy21@mails.tsinghua.edu.cn; rc20@mails.tsinghua.edu.cn; wanghuandong@tsinghua.edu.cn; liyong07@tsinghua.edu.cn)

rally described in different forms, which differ in their inputs, outputs, and objectives. For example, trajectory prediction typically relies on past trajectory points, whereas trajectory recovery demands both past and future trajectory points. More importantly, these tasks focus on different dependencies between trajectory points. Therefore, it's challenging to model these diverse tasks within a general framework.

- **Complexity of the conditions imposed on different tasks.** Different trajectory tasks face varying conditions, which may be determined by external settings or inherent to the tasks. For example, unlike unconditional trajectory generation, controllable generation relies on setting conditions such as radius of gyration and speeds. Moreover, there are inherent differences in conditions between different tasks. For example, trajectory prediction requires modeling the historical patterns of specific users. In contrast, trajectory generation does not focus on specific users but rather captures the mobility patterns of the entire group. Thus, developing a general framework that can adapt to complex conditions is crucial but challenging.

To address the above challenges, we propose a general mobility modeling framework via masked conditional diffusion named **GenMove**. Firstly, we design a mask condition module to unify multiple tasks into a common format of task objectives and conditional observations. We utilize various masking strategies, including task-specific and pattern-general masks. Task-specific masks precisely shield different trajectory parts, effectively adapting to the specific objectives of different tasks. Additionally, pattern-general masks model common mobility patterns across various tasks. The mask is used as a condition for the diffusion model, enabling a single model to address multiple mobility modeling tasks. Secondly, we obtain contextual trajectory embedding from historical data to provide essential context for specific tasks. Then, we apply the contextual trajectory embedding to diffusion models using a classifier-free guidance approach, which enables the model to flexibly adjust its outputs based on different conditions. Experimentally, as shown in Figure 1, GenMove achieves state-of-the-art performance on multiple tasks. Overall, the contributions of our paper can be summarized as follows:

- We unify mobility trajectory modeling, aiming to explore the vast potential of a general model capable of handling various trajectory tasks.
- We propose a general trajectory modeling framework via masked conditional diffusion. Specifically, well-designed mask conditions unify diverse formats of different tasks. Then, it employs a classifier-free guidance approach to incorporate context information from the trajectory, addressing the complex conditions associated with different tasks.
- We conduct extensive experiments on six trajectory-related tasks using two real-world datasets. The results demonstrate that our method outperforms state-of-the-art models across six tasks, with performance enhancements exceeding 13% in trajectory generation.

II. RELATED WORK

a) Trajectory-based Tasks: In practical applications, there exist typical trajectory-based tasks such as trajectory generation,

recovery, and prediction [19], [20], [4]. The objectives and methods of different tasks vary significantly. Trajectory generation depends on learning the distribution of real trajectories to generate new trajectory data. For example, the EPR model uses the patterns of exploring new locations and returning familiar ones to generate trajectories [21]. TrajGen uses GAN to learn the spatial distribution of trajectory locations in the original data [7]. Trajectory prediction uses historical trajectory information to predict the future location. For example, TrImpute relies on crowd wisdom to guide trajectory recovery using the spatial relationships of neighboring GPS points [8]. AttnMove utilizes various intra-trajectory and inter-trajectory attention mechanisms to better recover the mobility regularity of users [9]. Trajectory recovery focuses on reconstructing missing parts of trajectory data. For example, MobTCast uses a transformer-based context-aware network for mobility prediction [22]. STAN uses non-contiguous but functionally similar visited points to predict the next location [23]. Each task requires different model architectures and training strategies because of the different objectives. In this work, we propose a general model framework to address multiple trajectory-related tasks through a single model.

b) Task-general Models: Task-general Models utilize a single model to handle multiple tasks, eliminating the need for separate algorithms for each task. They have rapidly advanced in various fields such as computer vision, natural language processing, and time series analysis [18], [24], [25]. Task-general models are typically divided into two categories. The first category involves building models from scratch specifically for a particular domain. For example, TimesNet is designed with specialized backbone networks that capture intra-periodic and inter-periodic temporal variations, addressing multiple time series analysis tasks [26]. TrajGDM uses a trajectory generation framework to capture common mobility patterns in trajectory datasets, accomplishing trajectory-related tasks [27]. The second category involves leveraging pre-trained large language models (LLMs) to adapt to specific domains. Time-LLM is a methodology that utilizes LLMs for broad time series forecasting while leaving the backbone language models intact [28]. UrbanGPT is a general spatiotemporal prediction framework that integrates spatiotemporal information into language models. It enables LLMs to deeply understand the complex connections between time and space [29]. In this work, we develop a task-general framework from scratch for various trajectory modeling tasks. It can handle more complex tasks, such as controllable generation and scarcity-constrained prediction, beyond simple trajectory tasks.

III. PRELIMINARIES

A. Problem Definition

a) Definition 1: (Mobility Trajectory): The mobility trajectory of user u is a sequence of locations sampled at equal time intervals, denoted by $s_u = \{l_1, l_2, \dots, l_n\}$, where each location l_i is represented as the form in latitude and longitude coordinates or a region ID.

b) Definition 2: (General Mobility Trajectory Modeling Problem): The objective of this problem is to utilize a single

model to accomplish various trajectory-related tasks, such as trajectory generation, recovery, and prediction.

- *(Trajectory Generation)*. Given a set of real-world collected trajectories, $\mathcal{S} = \{s_1, s_2, \dots, s_m\}$, where each $s_u = \{l_1, l_2, \dots, l_n\}$ is a sequence of locations, the trajectory generation problem aims to develop a generative model G capable of generating synthetic trajectories.
- *(Trajectory Recovery)*. Given the sparse trajectory of user u with some missing values $s_u = \{l_1, \emptyset, l_3, \dots, \emptyset, l_n\}$, where $\emptyset$ represents the unobserved location at the corresponding time slot, the objective of the trajectory recovery task is to recover the missing locations.
- *(Trajectory Prediction)*. Given user u 's past trajectory $s_u = \{l_1, l_2, \dots, l_n\}$, the goal of the trajectory prediction task is to estimate the probability of their next location l_{n+1} .

B. Conditional Denoising Diffusion Probabilistic Models

The expression $p_\theta(x_0) := \int p_\theta(x_{0:T}) dx_{1:T}$ describes diffusion models, a type of latent variable model where $x_1, \dots, x_T$ are latent vectors matching the dimensionality of the data $x_0 \sim q(x_0)$. In the diffusion probabilistic model, two Markov chains are utilized: one to incrementally introduce noise to the data, and the other to reverse this process and restore the original data. The following Markov chain defines the diffusion process:

$$q(\mathbf{x}_{1:T} | \mathbf{x}_0) := \prod_{t=1}^T q(\mathbf{x}_t | \mathbf{x}_{t-1}), \quad (1)$$

where $q(\mathbf{x}_t | \mathbf{x}_{t-1}) := \mathcal{N}(\sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I})$ and β_t are modest positive constants that reflect the noise intensity. x_t may be represented in closed form as $x_t = \sqrt{\alpha_t}x_0 + (1-\alpha_t)\epsilon$. for $\epsilon \sim \mathcal{N}(0, \mathbf{I})$, where $\alpha_t = \sum_{i=1}^t (1-\beta_i)$.

Next, we discuss the conditional extension [30], where the modeled x is conditioned on observations c . Estimating the actual conditional data distribution is the goal of the conditional denoising diffusion probabilistic model. With a model distribution, $q(x_0|c) p_\theta(x_0|c)$.

Conversely, the conditional reverse process, which is represented by the Markov chain below, denoises x_t to retrieve x_0 :

$$p_\theta(\mathbf{x}_{0:T}|c) := p(\mathbf{x}_T) \prod_{t=1}^T p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t, c), \quad (2)$$

where $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Again, $p_\theta(x_{t-1}|x_t, c)$ is assumed as normally distributed with learnable parameters as follows:

$$p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t, c) := \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t|c), \sigma_\theta(\mathbf{x}_t, t|c) \mathbf{I}). \quad (3)$$

According to Ho et al. [31], the unconditional denoising diffusion model takes into account the particular parameterization of $p_\theta(x_{t-1}|x_t)$ as follows:

$$\begin{cases} \boldsymbol{\mu}_\theta(\mathbf{x}_t, t) = \frac{1}{\alpha_t} \left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\alpha_t}} \epsilon_\theta(\mathbf{x}_t, t) \right), \\ \sigma_\theta(\mathbf{x}_t, t) = \tilde{\beta}_t^{1/2} \text{ where } \tilde{\beta}_t = \begin{cases} \frac{1-\alpha_{t-1}}{1-\alpha_t} \beta_t & t > 1 \\ \beta_1 & t = 1 \end{cases} \end{cases}, \quad (4)$$

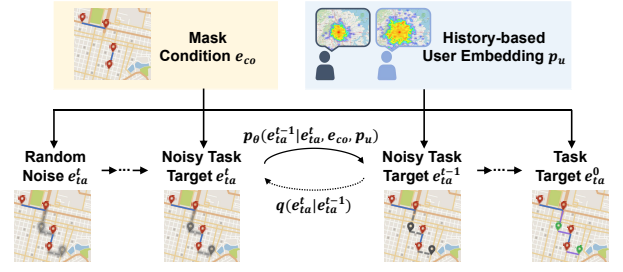


Fig. 2. The procedure of mobility modeling with GenMove. The reverse process p_θ gradually converts random noise into trajectory embedding, conditioned on mask condition e_{co} and history-based embedding p_u .

where ϵ_θ is a trainable denoising function. In Eq. 4, we represent $\boldsymbol{\mu}_\theta(\mathbf{x}_t, t)$ and $\sigma_\theta(\mathbf{x}_t, t)$ as $\mu^{\text{DDPM}}(\mathbf{x}_t, t, \epsilon_\theta(\mathbf{x}_t, t))$ and $\sigma^{\text{DDPM}}(\mathbf{x}_t, t)$.

Expanding upon the unconditional diffusion model, we derive a conditional denoising function ϵ_θ that accepts the condition c as input. Next, we examine the given parameterization with ϵ_θ :

$$\begin{cases} \boldsymbol{\mu}_\theta(\mathbf{x}_t, t | c) = \mu^{\text{DDPM}}(\mathbf{x}_t, t, \epsilon_\theta(\mathbf{x}_t, t | c)), \\ \sigma_\theta(\mathbf{x}_t, t | c) = \sigma^{\text{DDPM}}(\mathbf{x}_t, t). \end{cases} \quad (5)$$

The following loss function is minimized in order to train ϵ_θ :

$$\min_{\theta} \mathcal{L}(\theta) := \min_{\theta} \mathbb{E}_{\mathbf{x}_0 \sim q(\mathbf{x}_0), \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), t} \|\epsilon - \epsilon_\theta(\mathbf{x}_t, t | c)\|_2^2, \quad (6)$$

where $\mathbf{x}_t = \sqrt{\alpha_t}\mathbf{x}_0 + (1-\alpha_t)\epsilon$. The noise vector added to its noisy input x_t by the denoising function ϵ_θ is estimated, down-weighting the relevance of words at tiny t or low noise levels, and can be thought of as a negative log-likelihood weighted variational restriction.

IV. METHOD

Figure 2 shows the procedure of trajectory modeling with GenMove. We formulate the learning of the spatiotemporal distribution of trajectories as a denoising diffusion process. In the diffusion process, noise is progressively added to the trajectory until it becomes random noise. During the denoising process, we use e_{co} from the mask condition module and p_u from the contextual trajectory embedding module as conditions to guide the reconstruction of the trajectory from noise. As shown in Figure 3, the overview architecture of GenMove consists of three modules: mask condition, contextual trajectory embedding, and noise predictor. To handle diverse formats of different tasks, we utilize masking strategies to unify them with the standardized formats of task targets and conditional observations. To adapt to complex conditions associated with different tasks, we utilize contextual trajectory embedding from historical data to provide essential context for specific tasks. Subsequently, we apply this embedding to noise predictor through a classifier-free guidance approach, allowing the model to adapt its outputs flexibly based on the conditions encountered.

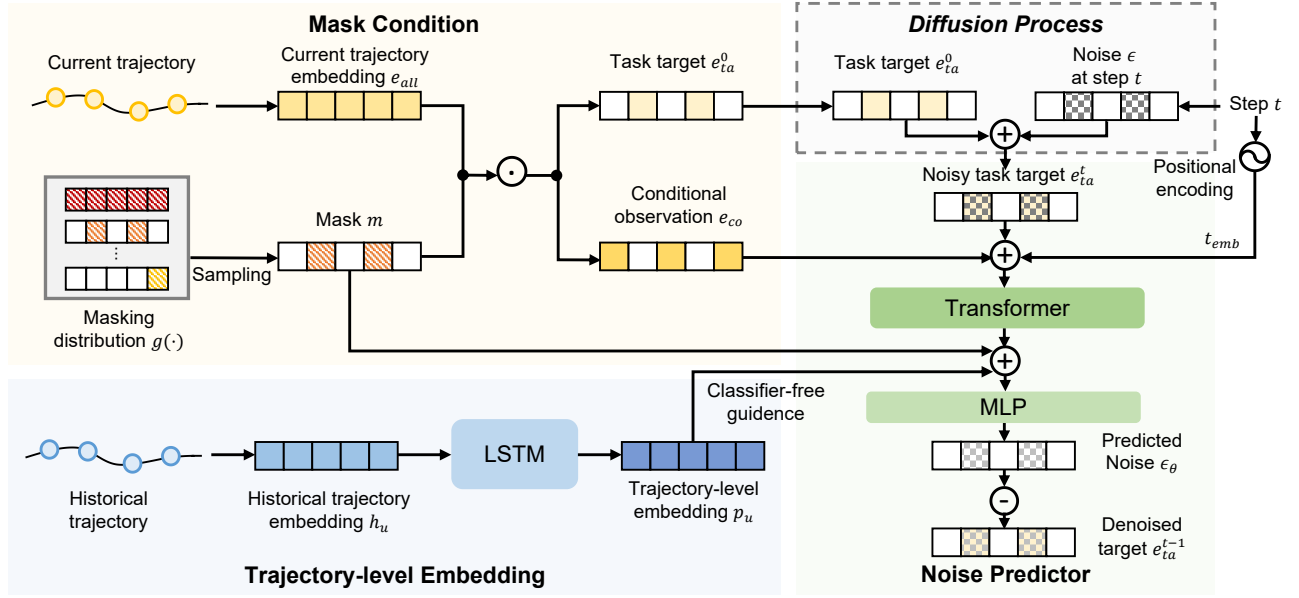


Fig. 3. The overview architecture of GenMove, which takes the noisy task target e_{ta}^t at the diffusion step t , as well as diffusion step embedding t_{emb} , the conditional observation e_{co} , mask m , and user embedding p_u as input.

A. Mask Condition

We unify diverse formats of different tasks through masking strategies, including task-specific masks and pattern-general masks. Specifically, we apply task-specific masks to mask different trajectory parts as task targets for various tasks, enabling a single model to solve multiple mobility modeling tasks: trajectory generation, prediction, and recovery. We utilize pattern-general masks for modeling common mobility patterns across various tasks.

Following the common strategies adopted in previous works [5], [32], we perform diffusion and denoising processes on the trajectory embedding. To represent the spatiotemporal dependency of the trajectory, we embed the trajectory into a dense representation as the input for other modules. In particular, to improve the geographical continuity of mobility, we build a spatial graph $G = (V, E)$ that includes a set of nodes V that contains all visited locations, and the weight of each edge represents the Euclidean distance between locations. We utilize the LINE method [33] to obtain the current trajectory embedding e_{all} .

Then, we construct the mask distribution $g(\cdot)$ by mixing five masking strategies based on adjustable ratios. Each time, we sample from the masking distribution to obtain a specific mask m . Therefore, we unify the input forms of all tasks as the current trajectory embedding e_{all} and mask m . We transform e_{all} into conditional observation e_{co} and task target e_{ta}^0 through the following operations:

$$\begin{cases} e_{co} = e_{all} \odot m, \\ e_{ta}^0 = e_{all} \odot (1 - m), \end{cases} \quad (7)$$

where e_{co} denotes conditional observation, e_{ta}^0 denotes task target, $m \in \{0, 1\}$ denotes mask, and $\odot$ represents element-wise products.

We provide five masking strategies as shown in Figure 4, including task-specific masks corresponding to the features of different tasks, and pattern-general masks for modeling common mobility patterns across various tasks.

(1) Random strategy: This strategy is for the recovery task. It is used when uncertain about missing patterns, selecting recovery targets arbitrarily as a percentage of observed values.

(2) Terminal strategy: This strategy masks future parts of a trajectory, such as the last point or a few points, and is particularly suited for prediction tasks.

(3) Complete strategy: This strategy is designed for the generation task. This strategy masks the entire trajectory, as no part is needed for observation in generation tasks.

(4) Sequential strategy: This strategy is inspired by the inherent pattern of people's mobility in the real world, where movement is the continuous location transition. It continuously masks a certain percentage of observations.

(5) Circadian rhythm strategy: This strategy aims to capture the circadian rhythm of human mobility, characterized by regular daytime work and nighttime rest patterns. Thus, it masks hours between 12 a.m. and 6 a.m., when people are not frequently moving.

B. Trajectory-level Embedding

We design the trajectory-level embedding to provide essential context for different tasks from historical trajectories, adapting to complex conditions associated with different tasks.

We obtain the historical trajectory embedding h_u using the trajectory embedding method. We then use LSTM to map users' historical trajectories to trajectory-level embedding p_u , $p_u = \text{LSTM}_\phi(h_u)$ enabling users with similar mobility patterns to have similar trajectory-level embedding. The obtained trajectory-level embedding p_u is then introduced into the noise predictor using classifier-free guidance, assisting the trajectory modeling process.

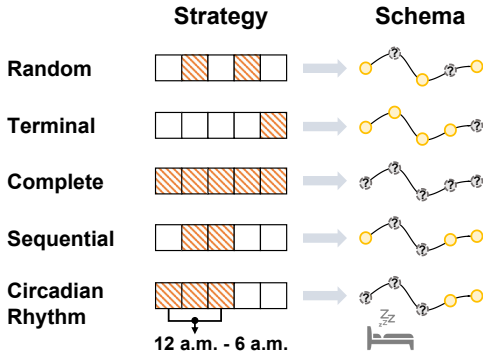


Fig. 4. Different masking strategies are divided into task-specific masks and mobility-common masks.

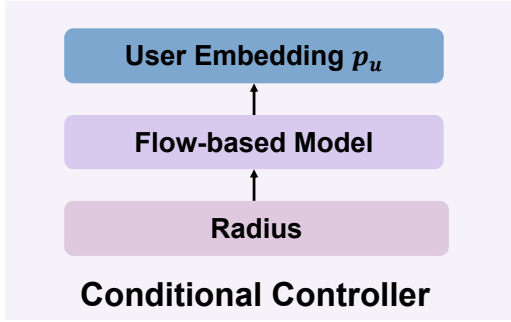


Fig. 5. Conditional controller replaces History-based user embedding.

Moreover, we propose *Conditional Controller* to replace *Trajectory-level Embedding*, achieving controllable trajectory generation. Drawing inspiration from GPT 4's ability to integrate plugins for various functions [34], we design a plugin-like *Conditional Controller* outside the main model, as shown in Figure 5.

Based on the trajectory-level embedding from the *Trajectory-level Embedding* and some features extracted from trajectories (such as radius), we train a conditional controller:

$$p_u = f_{\xi}(r_u). \quad (8)$$

We use the flow-based model [35] as a generative model $f_{\xi}(\cdot)$ to produce user embedding p_u based on conditions r_u . The p_u output by the *Conditional Controller* serves as a condition for the noise predictor, guiding generation to meet specific conditions, such as a restricted radius of gyration.

C. Noise Predictor

We utilize a transformer to capture the spatio-temporal dependencies of trajectories. Specifically, each step in the denoising process uses the previous prediction value e_{ta}^t , the denoising step t with positional encoding, conditional observation e_{co} , mask m , and user embedding p_u to achieve conditional denoising.

For the diffusion step t , we use positional encoding to obtain a 128-dimensional embedding t_{emb} following the previous

works [36], [37]:

$$t_{emb} = \left[\sin \left(10^{0 \times 4 / 63} t \right), \dots, \sin \left(10^{63 \times 4 / 63} t \right), \right. \\ \left. \cos \left(10^{0 \times 4 / 63} t \right), \dots, \cos \left(10^{63 \times 4 / 63} t \right) \right]. \quad (9)$$

Then, we use the conditional observation e_{co} and the mask m from *Mask Condition* as additional input to the *Noise Predictor*. We introduce the user embedding p_u from *History-based User Embedding* using classifier-free guidance to jointly train conditional and unconditional diffusion models. The *Noise Predictor* takes noisy task target e_{ta}^t , diffusion step t , conditional observation e_{co} , mask m , and user embedding p_u as input, outputting the predicted noise:

$$\hat{\epsilon}_{\theta} = \epsilon_{\theta}(e_{ta}^t, e_{co}, t, m, p_u), \quad (10)$$

The output $\hat{\epsilon}_{\theta}$ is of the same shape as e_{ta}^0 and ϵ_{θ} .

D. Training and Sampling

a) *Training*: We sample different masks from the mask distribution based on a blend of five masking strategies and use these masks to obtain different conditional observation e_{co} mentioned in Section IV-A. The conditional observation e_{co} serves as a condition for the conditional diffusion model. Then, we use classifier-free guidance to train our model. Specifically, we concurrently train an unconditional denoising diffusion model and a conditional denoising diffusion model. A singular neural network is used to parameterize both models. In the case of the unconditional model, a null token $\emptyset$ is used as input for the user embedding p_u during the noise prediction process. We jointly train the unconditional and conditional models by randomly setting p_u to the unconditional class identifier $\emptyset$ with some probability λ , which is regarded as a hyperparameter in our proposed algorithm.

Based on a similar derivation in Section III-B, we train the model using a simplified loss function:

$$\mathcal{L}(\theta) = \mathbb{E}_{e_{ta}^0, \epsilon, t} \|\epsilon - \epsilon_{\theta}(e_{ta}^t, t | e_{co}, p_u)\|_2^2. \quad (11)$$

b) *Sampling*: We sample using the following linear combination of the conditional and unconditional model [38]:

$$\tilde{\epsilon}_{\theta} = (1 + \omega)\epsilon_{\theta}(e_{ta}^t, t | e_{co}, p_u) - \omega\epsilon_{\theta}(e_{ta}^t, t | e_{co}, \emptyset), \quad (12)$$

where $\emptyset$ indicates a vector that effectively does not encode any information, with all conditional information set to 0. The model is a diffusion model with just conditional user embedding p_u when $\omega = 0$. The model will prioritize unconditionally predicting noise with a trade-off between variety and sample quality by raising ω . The model is an unconditional diffusion model for $\omega = 1$.

Figure 6 illustrates the iterative denoising steps during the sampling process. In this process, the noise predicted at each step is removed, progressively decreasing from T steps to 1 step, ultimately resulting in clean data, denoted as e_{ta}^0 . Below, we list the formula for noise removal at each step:

$$e_{ta}^{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(e_{ta}^t - \frac{1 - \alpha_t}{\sqrt{1 - \alpha_t}} \tilde{\epsilon}_{\theta} \right) + \sigma_t z \quad (13)$$

where $\tilde{\epsilon}_{\theta}$ is a function approximator intended to predict ϵ from e_{ta}^t and $z \sim N(0, I)$.

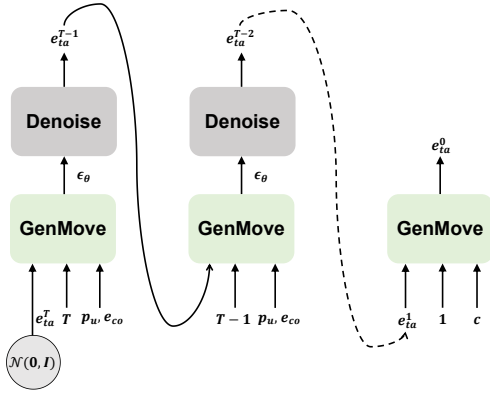


Fig. 6. The sampling process of GenMove.

TABLE I
BASIC STATISTICS OF MOBILITY DATASETS.

Dataset	City	Duration	#Users	#Loc	#Traj
ISP	Shanghai	7 days	90037	9158	261042
MME	Nanchang	7 days	6218	4096	43967

V. EXPERIMENTS

A. Experimental Settings

1) *Dataset*: We conducted broad experiments on two real-world datasets (**ISP** and **MME**), which were collected from two cities in China (Shanghai and Nanchang). The details of datasets are summarized in Table I.

- **ISP**: The ISP dataset is sourced from a prominent Internet service provider in China. The dataset contains trajectories of over 90000 users in Shanghai over a week, each containing an anonymous user ID, timestamp, and cellular base station.
- **MME**: The MME dataset is provided by an operator in China. The dataset contains the trajectories of more than 6000 users over a week in Nanchang, with each mobility record containing an anonymous user ID, time stamp, and region ID.

Moreover, we have taken proactive measures to protect users' privacy in our dataset. Firstly, all data collected in this study are stored on secure data warehouse servers behind the company's firewall. Secondly, staff from our partners, ISP and MME, have anonymized all user information, including personal and location identifiers.

2) *Tasks*: As shown in Figure 7, we evaluate our method across six typical trajectory-related tasks. However, our model is not limited to these tasks; it can adapt to a broader range of tasks. These six tasks are categorized into generation, recovery, and prediction. Each type is further divided into basic and extended tasks. For example, in the generation category, the basic task is the unconditional generation, while the extended task is the controllable generation (i.e., generating trajectories that meet specific conditions). In the prediction category, the basic task involves the next location prediction, while the extended task involves long-term prediction (predicting locations over multiple time steps). In the recovery category, the basic task is trajectory recovery, and the extended task is

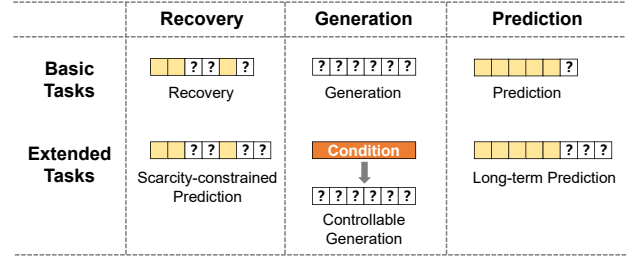


Fig. 7. Six trajectory-related tasks, divided into basic and extended tasks.

scarcity-constrained prediction (i.e., the next location prediction based on sparse trajectories).

3) *Metrics*: Following the previous related works [19], [13], [9], [20], [4], [23], we select common metrics for mobility modeling tasks to evaluate the effectiveness of our model.

a) *Trajectory Generation*: We evaluate the generated trajectory using the following six metrics:

- **Distance**: This is travel distance, which is determined by adding up each user's trip distance over a certain period.
- **Radius**: The spatial range of the user's daily movement is represented by the radius of gyration.
- **Duration**: This refers to the duration of stay, measured as the time spent at each visited destination.
- **Daily-loc**: This refers to the daily frequency of visited locations, quantified as the number of locations each user trips every day.
- **Density**: This metric evaluates the geographical distribution by comparing the density of the total and actual trajectories on a global scale.
- **Trip**: At the trajectory level, this statistic measures the relationship between the starting and ending points of a trip. It involves calculating the probability distributions for the initial and final locations in both the original and generated trajectories.

We employ the Jensen-Shannon divergence (JSD) to visually assess the similarity between the generated sequences and the real sequences. Specifically, for two distributions, $\mathbf{m}$ and $\mathbf{n}$, the Jensen-Shannon Divergence (JSD) between them is defined as:

$$\text{JSD}(\mathbf{m}, \mathbf{n}) = \frac{1}{2} \text{KL}(\mathbf{m} \| \frac{\mathbf{m} + \mathbf{n}}{2}) + \frac{1}{2} \text{KL}(\mathbf{n} \| \frac{\mathbf{m} + \mathbf{n}}{2}), \quad (14)$$

where $\text{KL}(\cdot \| \cdot)$ is the Kullback-Leibler divergence [39].

b) *Trajectory Recovery*: We evaluate the recovered trajectory using the following three metrics:

- **Recall**: It evaluates the model's ability to recover all ground-truth locations, averaged over all tested values. If the Recall is 1, it means that all ground-truth locations are fully recovered, as opposed to 0 if they are not recovered at all.
- **MAP**: It assesses the global ranking task to determine the quality of the entire ranking list, encompassing all locations. A higher MAP number indicates superior performance.
- **Distance**: It is defined as the geographical distance between the recovery location's center and the ground truth. The performance will improve as the distance decreases.

c) *Trajectory Prediction*: We established Accuracy@k as our assessment metric. To determine the top-k prediction accuracy, we initially employ the model to produce the k most probable predictions derived from the provided historical data, subsequently verifying if the actual subsequent location is among these top-k projections.

We repeated the experiments for each task five times, and the performance metrics presented are the average values of these five experiments.

4) *Baselines*: We compare the performance of our model with state-of-the-art baselines for different basic tasks. The baseline for the extended task is derived from a combination of one or more best baselines capable of completing the task.

a) *Trajectory Generation*: The baselines for trajectory generation are as follows:

- **TimeGEO** [6] is a model-based trajectory synthesis method that characterizes temporal choices through dwell rate, burst rate, and the weekly home-based tour frequency, while employing the explore and preferential return (EPR) model [21] to represent spatial choices.
- **MoveSim** [19]: This generative adversarial approach integrates the domain knowledge of human movement patterns.
- **VOLUNTEER** [40]: This model consists of a two-layer structure of user VAE and trajectory VAE to model the joint distribution of user attributes and mobility behavior to generate more practical trajectories.
- **PateGail** [13]: It employs the robust generative adversarial imitation learning model to replicate human decision-making processes.
- **DiffTraj** [12]: It utilizes diffusion models to generate trajectories by using the spatiotemporal features of real trajectories as conditional information

b) *Trajectory Recovery*: The baselines for trajectory recovery are as follows:

- **Linear** [41]: This approach presumes that a linear function can estimate the missing trajectory points. It is a technique based on rules with a significant assumption.
- **TrImpute** [8]: This approach depends on collective intelligence, using adjacent GPS coordinates to facilitate recovery.
- **RF** [42]: The random forest utilizes all elements of each trajectory for recovery. The random forest classifier is developed by training on features such as the entropy and radius of each trajectory, the gaps in time where data is missing, and the locations immediately before and after these gaps.
- **AttnMove** [9]: It employs diverse intra-trajectory and inter-trajectory attention processes to enhance the modeling of user movement patterns and capitalizes on periodic patterns in long-term history to reconstruct trajectories.
- **PeriodicMove** [20]: This is an innovative neural attention model based on a graph neural network that systematically captures intricate location transition patterns, multilevel periodicity, and the evolving periodicity of human movement.

c) *Trajectory Prediction*: The baselines for trajectory prediction are as follows:

- **Markov Model** [43]: The Markov model is a statistical framework used to characterize state transitions across time.

It employs previous trajectory data for position forecasting by computing the transition probabilities among different locations.

- **DeepMove** [4]: The method develops a multimodal embedding recurrent neural network to encapsulate intricate sequential transitions by concurrently embedding several elements that influence human mobility.
- **STAN** [23]: This approach correlates non-adjacent but functionally analogous visited locations to forecast the next location.
- **DSTPP** [5]: The approach employs diffusion models to comprehend intricate spatio-temporal joint distributions for trajectory prediction.

d) *Task-general Model*: The baselines are a general model capable of accomplishing three tasks: trajectory generation, recovery, and recovery.

- **LSTM** [44]: The LSTM network excels in processing sequential data and can encode long-term dependencies, which can naturally be applied to different tasks.
- **TrajGDM** [27]: This method is a trajectory generation framework using the diffusion model to encapsulate the universal mobility pattern inside a trajectory dataset.

B. Hyperparameters Settings

The dataset is divided by the user, with the first 70% as the training set, the second 10% as the validation set, and the remaining 20% as the test set. As for the hyperparameter settings of our model, we set the embedding size of the diffusion step and the size of the location embedding to 128, the number of channels of the convolutional network to 64, the number of transformer layers to 4, and the number of heads of the attention mechanism to 8. Then, we set the learning rate to 1e-3, the diffusion steps to 1000, and the batch size to 16 during the training process. The details of hyperparameters settings are shown in Table V. Finally, the proposed framework is implemented with Pytorch. We train it on a Linux server with eight GPUs (NVIDIA RTX 2080 Ti * 8).

C. Overall Performance

As shown in Figure 1 and 8, our method demonstrates similar or better performance than the state-of-the-art baselines for all tasks on two datasets. This reflects that our general framework is capable of performing various tasks with optimal performance.

1) *Trajectory Generation*: Table II shows the performance in the basic task of unconditional generation, where our method outperformed others across all metrics, with an average performance difference of about 13% compared to various baselines. We also visualized the generated results to better present the performance, as shown in Figure 9. Our model's generated trajectory distribution resembles the real trajectory distribution the most. The distribution generated by the baseline is more uniform and fails to replicate the trajectory distribution accurately. This reflects that the trajectories generated by our model can more accurately simulate real mobility patterns. Considering the demand for controllable generation in practical

TABLE II
PERFORMANCE COMPARISONS OF THE TRAJECTORY GENERATION TASK ON TWO MOBILITY DATASETS.

	ISP						MME					
	Distance	Radius	Duration	Daily-loc	Density	Trip	Distance	Radius	Duration	Daily-loc	Density	Trip
TimeGEO	0.0628	0.4703	0.0912	0.4257	0.0295	0.0726	0.0443	0.4126	0.0623	0.3868	0.0320	0.0619
LSTM	0.0712	0.0540	0.1185	0.4051	0.0258	0.0682	0.0362	0.3987	0.0601	0.3247	0.0309	0.5327
MoveSim	0.0254	0.4021	0.0635	0.2802	0.0247	0.0658	0.0217	0.3572	0.0561	0.2526	0.0294	0.0451
VOLUNTEER	0.0137	0.4316	0.0453	0.1584	0.0217	0.0506	0.0098	0.3725	0.0507	0.1435	0.0215	0.0403
PateGail	0.0102	0.2125	0.0384	0.0630	0.0203	0.0358	0.0004	0.1806	0.0783	0.0887	0.0168	0.0310
DiffTraj	<u>0.0085</u>	<u>0.1736</u>	<u>0.0307</u>	<u>0.0541</u>	0.0158	0.0280	<u>0.0002</u>	<u>0.1657</u>	<u>0.0498</u>	<u>0.0653</u>	0.0105	0.0241
TrajGDM	0.0089	0.1864	0.0326	0.0571	0.0139	<u>0.0224</u>	0.0003	0.1701	0.0515	0.0712	<u>0.0092</u>	<u>0.0254</u>
GenMove	0.0074	0.1431	0.0262	0.0488	0.0122	0.0249	0.0002	0.1512	0.0467	0.0507	0.0078	0.0181

TABLE III
PERFORMANCE COMPARISONS OF THE TRAJECTORY PREDICTION TASK ON TWO MOBILITY DATASETS.

	ISP			MME		
	Acc@1	@3	@5	Acc@1	@3	@5
Markov	0.3164	0.4215	0.5323	0.3832	0.4768	0.6027
LSTM	0.3772	0.4581	0.5907	0.4325	0.5152	0.6590
DeepMove	0.4013	0.4872	0.6391	0.4683	0.5324	0.6838
STAN	0.4175	0.4946	0.6503	0.4819	0.5506	0.7014
DSTPP	<u>0.4201</u>	<u>0.4968</u>	<u>0.6510</u>	<u>0.4838</u>	<u>0.5520</u>	<u>0.7022</u>
TrajGDM	0.4182	0.4950	0.6501	0.4768	0.5430	0.6917
GenMove	0.4248	0.5015	0.6596	0.4865	0.5552	0.7068

TABLE V
THE DETAILS OF HYPERPARAMETERS SETTINGS.

Hyper-parameters	Settings
The embedding size of diffusion step	128
The embedding size of location embedding	128
The channel number of the convolutional network	64
Transformer layer number	4
Transformer head number	8
Learning rate	1e-3
Diffusion steps	1000
Batch size	16

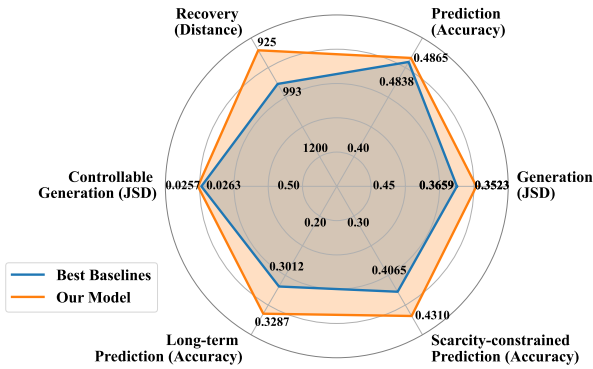


Fig. 8. Model performance comparison on MME dataset.

applications, we further conducted experiments on controllable generation. Figure 10 shows the effect of our controllable generation. Specifically, we use different radius constraints to control the generation. We find that setting different radius for controllable generation results in trajectories with distinctly different ranges.

TABLE IV
PERFORMANCE COMPARISONS OF THE TRAJECTORY RECOVERY TASK ON TWO MOBILITY DATASETS.

	ISP			MME		
	Recall	MAP	Dis.	Recall	MAP	Dis.
Linear	0.5024	0.5385	1709	0.5259	0.5649	1364
TrImpute	0.5406	0.5963	1602	0.5723	0.6216	1219
RF	0.4025	0.4397	2215	0.4371	0.4686	1752
LSTM	0.5210	0.6153	1583	0.6015	0.6741	1205
AttnMove	0.5893	0.6612	1436	0.6427	0.7153	1086
PeriodicMove	<u>0.5982</u>	<u>0.6748</u>	<u>1372</u>	<u>0.6548</u>	<u>0.7225</u>	<u>993</u>
TrajGDM	0.5715	0.6454	1597	0.6181	0.6678	1125
GenMove	0.6041	0.6856	1304	0.6674	0.7302	925

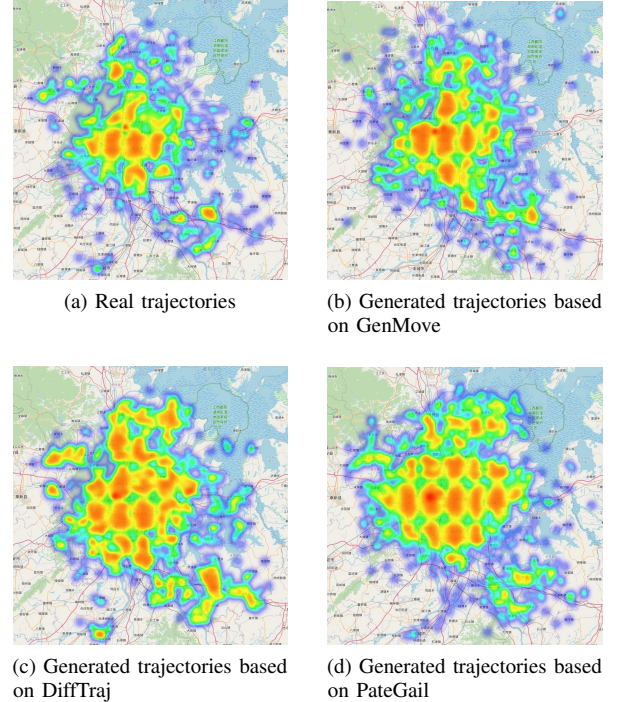


Fig. 9. Comparison of heatmaps between real and generated trajectories on the MME dataset.

2) *Trajectory Prediction*: The performance of our method on trajectory prediction is shown in Table III. For this task, our performance is only close to the best baseline, possibly because our model loses its specific design for single tasks to ensure generality for all tasks. However, our model significantly

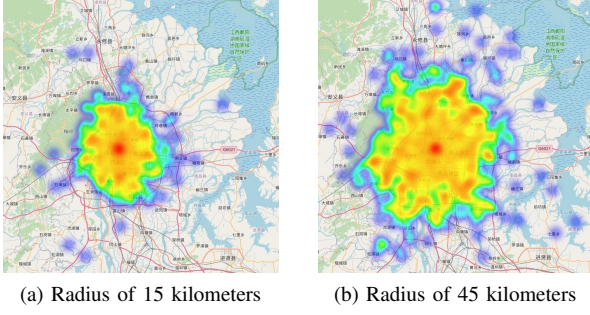


Fig. 10. Heatmap of controllable trajectory generation conditioned on the radius of gyration.

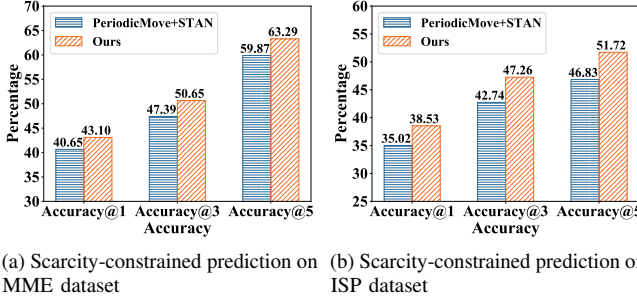


Fig. 11. Performance of scarcity-constrained trajectory prediction on two datasets.

outperforms other baseline models for extended task long-term trajectory prediction. Figure 12 shows the performance of our model in long-term trajectory prediction, predicting up to four hours (8 steps, with each step representing half an hour). We can observe that the performance of all models decreases as the prediction time step increases. However, our model consistently shows the best performance across different time steps. This indicates that our model is more suitable for long-term trajectory prediction tasks.

3) *Trajectory Recovery*: The performance of our method on trajectory recovery is shown in Table IV. Our model shows a significant performance improvement on the most representative metric Distance, achieving around 6% improvement. In the real world, the collected trajectory data is often sparse due to the low sampling frequency and limited coverage of the collection devices. Therefore, scarcity-constrained trajectory prediction is also a practical task. Since the current baseline cannot complete this task independently, we choose the best trajectory recovery baseline to convert sparse trajectories into dense ones and then use a trajectory prediction model for the next location prediction. As shown in Figure 11, our model exhibits an average performance improvement of 6.2% over the best baseline, because our random masking strategy allows the model to be trained with more trajectory samples with scarce history, enabling it to handle this task.

D. Zero-shot Performance

Zero-shot evaluation can demonstrate the adaptability of models. In datasets segmented by users, we train GenMove with data from some users and then evaluate it on a completely

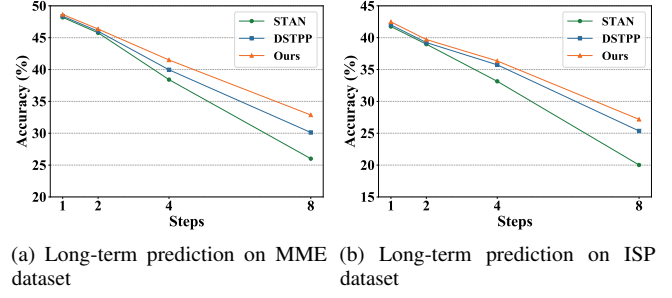


Fig. 12. Performance of long-term trajectory prediction on two datasets.

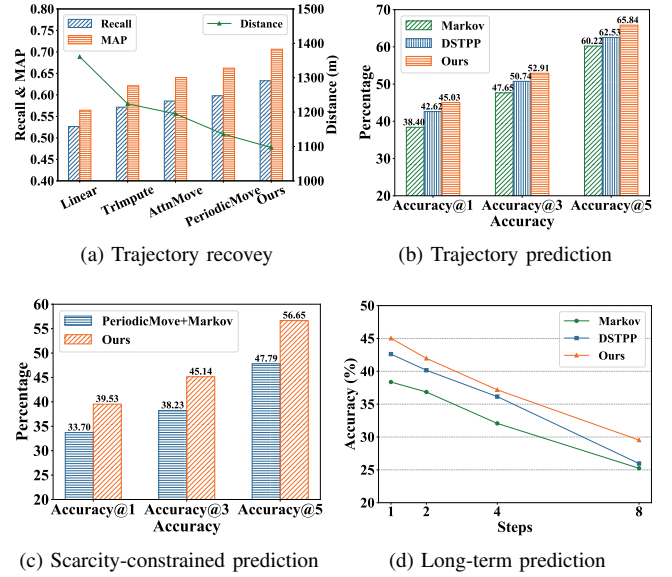


Fig. 13. Zero-shot performance in four tasks on the MME dataset.

unseen user dataset. Since the objective of the generation task is to model the trajectory distribution of all users without distinguishing between individuals, we do not create a new user-based dataset for generation tasks. We only performed zero-shot inference for tasks such as trajectory prediction and recovery.

As shown in Figure 13, GenMove exhibits exceptional zero-shot performance, significantly surpassing the best baseline. In scarcity-constrained prediction, it improves the Accuracy@5 metric by more than 18% compared to the best baseline; in long-term prediction tasks, the performance of predicting up to 8 steps ahead has improved by nearly 20%. These remarkable results demonstrate GenMove's ability to extract and learn similar mobility patterns from historical data. The zero-shot results further verify GenMove's strong generalization capabilities in entirely new scenarios.

E. Mutual Enhancement between Tasks

We analyze the relationship between mask ratio and task performance to explore whether frameworks that train multiple tasks together have advantages over traditional single task.

As shown in Figure 14, when the recovery or prediction task-specific mask (i.e., random masks) ratio is particularly

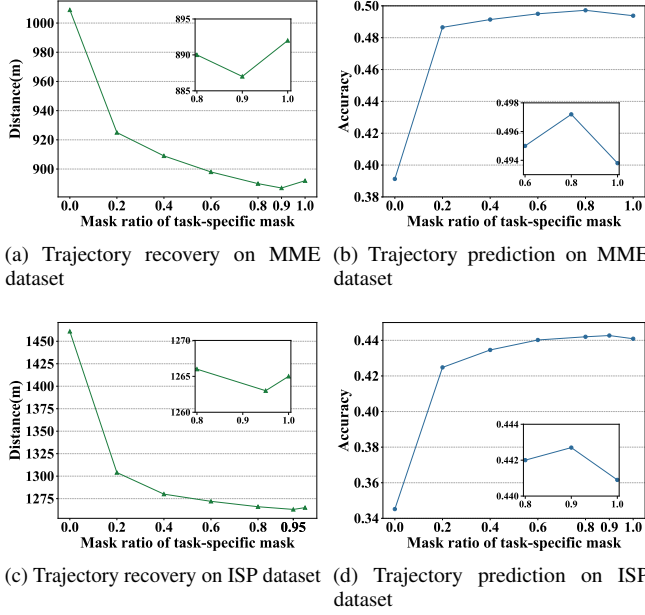


Fig. 14. Impact of the mask ratio on the performance of different tasks on two datasets.

small, the performance of the recovery or prediction task is poor. This proves the effectiveness of the task-specific masks. Moreover, we find an interesting phenomenon: the optimized performance is not at a mask ratio of 1, indicating that the general framework performs better when trained on multiple tasks simultaneously than when trained on single tasks separately. For example, Figure 14 (b) indicates optimal performance is at a mask ratio of 0.8 rather than 1, suggesting that the recovery or generation task enhances the prediction task. This supports our general framework's premise that different tasks can boost performance through shared patterns. This demonstrates that the general framework performs better when trained on multiple tasks simultaneously than when trained on single tasks separately. It further illustrates the motivation behind our general framework that different tasks can improve performance by sharing mobility patterns.

VI. CONCLUSION

In this paper, we address an important problem of building a general framework that accomplishes multiple tasks for mobility trajectory modeling. Diverse formats of different tasks are unified into a standard format of task objectives and conditional observations through masking strategies. By introducing contextual trajectory embedding through a classifier-free guidance approach, our model can flexibly adapt to different conditions. Extensive experiments conducted on diverse datasets with multiple evaluation metrics have demonstrated the superiority of our model. In the future, we plan to adopt transfer learning techniques to achieve mobility modeling across different cities, further enhancing the generality of our framework. Moreover, we can use different backbones to explore their impact on performance or generality.

REFERENCES

- [1] J. Wang, X. Kong, A. Rahim, F. Xia, A. Tolba, and Z. Al-Makhadmeh, "Is2fun: Identification of subway station functions using massive urban data," *IEEE Access*, vol. 5, pp. 27 103–27 113, 2017.
- [2] S. Wang, Y. Sun, C. Musco, and Z. Bao, "Public transport planning: When transit network connectivity meets commuting demand," in *Proceedings of the 2021 International Conference on Management of Data*, 2021, pp. 1906–1919.
- [3] T. Yabe, M. Luca, K. Tsubouchi, B. Lepri, M. C. Gonzalez, and E. Moro, "Enhancing human mobility research with open and standardized datasets," *Nature Computational Science*, pp. 1–4, 2024.
- [4] J. Feng, Y. Li, C. Zhang, F. Sun, F. Meng, A. Guo, and D. Jin, "Deepmove: Predicting human mobility with attentional recurrent networks," in *Proceedings of the 2018 world wide web conference*, 2018, pp. 1459–1468.
- [5] Y. Yuan, J. Ding, C. Shao, D. Jin, and Y. Li, "Spatio-temporal diffusion point processes," *arXiv preprint arXiv:2305.12403*, 2023.
- [6] S. Jiang, Y. Yang, S. Gupta, D. Veneziano, S. Athavale, and M. C. González, "The timegeo modeling framework for urban mobility without travel surveys," *PNAS*, vol. 113, no. 37, 2016.
- [7] C. Cao and M. Li, "Generating mobility trajectories with retained data utility," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021, pp. 2610–2620.
- [8] M. M. Elsharif, K. Isufaj, and M. F. Mokbel, "Network-less trajectory imputation," in *Proceedings of the 30th International Conference on Advances in Geographic Information Systems*, 2022, pp. 1–10.
- [9] T. Xia, Y. Qi, J. Feng, F. Xu, F. Sun, D. Guo, and Y. Li, "Attnmove: History enhanced trajectory recovery via attentional network," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 5, 2021, pp. 4494–4502.
- [10] X. Xu, C. Yang, and W. Wu, "Representation learning and graph convolutional networks for short-term vehicle trajectory prediction," *Physica A: Statistical Mechanics and its Applications*, vol. 637, p. 129560, 2024.
- [11] L. Rossi, A. Ajmar, M. Paolanti, and R. Pierdicca, "Vehicle trajectory prediction and generation using lstm models and gans," *Plos one*, vol. 16, no. 7, p. e0253868, 2021.
- [12] Y. Zhu, Y. Ye, X. Zhao, and J. J. Yu, "Diffusion model for gps trajectory generation," *arXiv preprint arXiv:2304.11582*, 2023.
- [13] H. Wang, C. Gao, Y. Wu, D. Jin, L. Yao, and Y. Li, "Pategail: A privacy-preserving mobility trajectory generator with imitation learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 12, 2023, pp. 14 539–14 547.
- [14] J. Dai, B. Yang, C. Guo, and Z. Ding, "Personalized route recommendation using big trajectory data," in *2015 IEEE 31st international conference on data engineering*. IEEE, 2015, pp. 543–554.
- [15] L. G. Xing, I. A. Abiodun, C. W. Khuen, and T. T. Boon, "A personalized recommendation framework with user trajectory analysis applied in location-based social network (lbsn)," in *2017 IEEE 3rd International Conference on Engineering Technologies and Social Sciences (ICETSS)*. IEEE, 2017, pp. 1–6.
- [16] S. Long, F. Cao, S. C. Han, and H. Yang, "Vision-and-language pretrained models: A survey," *arXiv preprint arXiv:2204.07356*, 2022.
- [17] L. Xie, L. Wei, X. Zhang, K. Bi, X. Gu, J. Chang, and Q. Tian, "Towards agi in computer vision: Lessons learned from gpt and large language models," *arXiv preprint arXiv:2306.08641*, 2023.
- [18] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat *et al.*, "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.
- [19] J. Feng, Z. Yang, F. Xu, H. Yu, M. Wang, and Y. Li, "Learning to simulate human mobility," in *Proc. KDD*, 2020.
- [20] H. Sun, C. Yang, L. Deng, F. Zhou, F. Huang, and K. Zheng, "Periodicmove: shift-aware human mobility recovery with graph neural network," in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 2021, pp. 1734–1743.
- [21] C. Song, T. Koren, P. Wang, and A.-L. Barabási, "Modelling the scaling properties of human mobility," *Nature Physics*, vol. 6, no. 10, pp. 818–823, 2010.
- [22] H. Xue, F. Salim, Y. Ren, and N. Oliver, "Mobtcast: Leveraging auxiliary trajectory forecasting for human mobility prediction," *Advances in Neural Information Processing Systems*, vol. 34, pp. 30 380–30 391, 2021.
- [23] Y. Luo, Q. Liu, and Z. Liu, "Stan: Spatio-temporal attention network for next location recommendation," in *Proceedings of the web conference 2021*, 2021, pp. 2177–2185.

- [24] Y. Liu, K. Zhang, Y. Li, Z. Yan, C. Gao, R. Chen, Z. Yuan, Y. Huang, H. Sun, J. Gao *et al.*, “Sora: A review on background, technology, limitations, and opportunities of large vision models,” *arXiv preprint arXiv:2402.17177*, 2024.
- [25] A. Garza and M. Mergenthaler-Canseco, “Timegpt-1,” *arXiv preprint arXiv:2310.03589*, 2023.
- [26] H. Wu, T. Hu, Y. Liu, H. Zhou, J. Wang, and M. Long, “Timesnet: Temporal 2d-variation modeling for general time series analysis,” *arXiv preprint arXiv:2210.02186*, 2022.
- [27] C. Chu, H. Zhang, P. Wang, and F. Lu, “Simulating human mobility with a trajectory generation framework based on diffusion model,” *International Journal of Geographical Information Science*, pp. 1–32, 2024.
- [28] M. Jin, S. Wang, L. Ma, Z. Chu, J. Y. Zhang, X. Shi, P.-Y. Chen, Y. Liang, Y.-F. Li, S. Pan *et al.*, “Time-llm: Time series forecasting by reprogramming large language models,” *arXiv preprint arXiv:2310.01728*, 2023.
- [29] Z. Li, L. Xia, J. Tang, Y. Xu, L. Shi, L. Xia, D. Yin, and C. Huang, “Urbangpt: Spatio-temporal large language models,” *arXiv preprint arXiv:2403.00813*, 2024.
- [30] Y. Tashiro, J. Song, Y. Song, and S. Ermon, “Csdi: Conditional score-based diffusion models for probabilistic time series imputation,” *Advances in Neural Information Processing Systems*, vol. 34, pp. 24 804–24 816, 2021.
- [31] J. Ho, A. Jain, and P. Abbeel, “Denosing diffusion probabilistic models,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 6840–6851, 2020.
- [32] Z. Zhou, J. Ding, Y. Liu, D. Jin, and Y. Li, “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*, 2023, pp. 1–12.
- [33] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, “Line: Large-scale information network embedding,” in *Proceedings of the 24th international conference on world wide web*, 2015, pp. 1067–1077.
- [34] Y. Liu, T. Han, S. Ma, J. Zhang, Y. Yang, J. Tian, H. He, A. Li, M. He, Z. Liu *et al.*, “Summary of chatgpt-related research and perspective towards the future of large language models,” *Meta-Radiology*, p. 100017, 2023.
- [35] B. Zhang, T. Wang, C. Zhou, N. Conci, and H. Liu, “Human trajectory forecasting using a flow-based generative model,” *Engineering Applications of Artificial Intelligence*, vol. 115, p. 105236, 2022.
- [36] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, E. Kaiser, and I. Polosukhin, “Attention is all you need,” *Advances in neural information processing systems*, vol. 30, 2017.
- [37] Z. Kong, W. Ping, J. Huang, K. Zhao, and B. Catanzaro, “Dif-fwave: A versatile diffusion model for audio synthesis,” *arXiv preprint arXiv:2009.09761*, 2020.
- [38] J. Ho and T. Salimans, “Classifier-free diffusion guidance,” *arXiv preprint arXiv:2207.12598*, 2022.
- [39] F. Torabi, G. Warnell, and P. Stone, “Behavioral cloning from observation,” *arXiv preprint arXiv:1805.01954*, 2018.
- [40] Q. Long, H. Wang, T. Li, L. Huang, K. Wang, Q. Wu, G. Li, Y. Liang, L. Yu, and 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] S. Hoteit, S. Secci, S. Sobolevsky, C. Ratti, and G. Pujolle, “Estimating human trajectories and hotspots through mobile phone data,” *Computer Networks*, vol. 64, pp. 296–307, 2014.
- [42] M. Li, S. Gao, F. Lu, and H. Zhang, “Reconstruction of human movement trajectories from large-scale low-frequency mobile phone data,” *Computers, Environment and Urban Systems*, vol. 77, p. 101346, 2019.
- [43] S. Gambs, M.-O. Killijian, and M. N. del Prado Cortez, “Next place prediction using mobility markov chains,” in *Proceedings of the first workshop on measurement, privacy, and mobility*, 2012, pp. 1–6.
- [44] D. Kong and F. Wu, “Hst-lstm: A hierarchical spatial-temporal long-short term memory network for location prediction,” in *Twenty-Seventh International Joint Conference on Artificial Intelligence IJCAI-18*, 2018.