DiffMove: Group Mobility Tendency Enhanced Trajectory Recovery via Diffusion Model

Qingyue Long, Can Rong, Huandong Wang, *Member, IEEE,* Rajib Shaw, Yong Li, *Senior Member, IEEE*

***Abstract*—In the real world, trajectory data is often sparse and incomplete due to low collection frequencies or limited device coverage. Trajectory recovery aims to recover these missing trajectory points, making the trajectories denser and more complete. However, this task faces two key challenges: 1) The excessive sparsity of individual trajectories makes it difficult to effectively leverage historical information for recovery; 2) Sparse trajectories make it harder to capture complex individual mobility preferences. To address these challenges, we propose a novel method called DiffMove. Firstly, we harness crowd wisdom for trajectory recovery. Specifically, we construct a group tendency graph using the collective trajectories of all users and then integrate the group mobility trends into the location representations via graph embedding. This solves the challenge of sparse trajectories being unable to rely on individual historical trajectories for recovery. Secondly, we capture individual mobility preferences from both historical and current perspectives. Finally, we integrate group mobility tendencies and individual preferences into the spatiotemporal distribution of the trajectory to recover high-quality trajectories. Extensive experiments on three real-world datasets demonstrate that DiffMove outperforms existing state-of-the-art methods. Further analysis validates the robustness of our method.**

***Index Terms*—Trajectory recovery, generative models, diffusion models.**

I. INTRODUCTION

Human mobility data reveals people's activity patterns and mobility intentions, making it of significant value in various fields such as epidemic control [1], [2], [3], urban planning [4], [5], [6], [7], and transportation engineering [8], [9]. However, the sparsity of real-life trajectory data, whether due to low sampling rates of mobile data devices or privacy constraints imposed by users, presents significant challenges and negatively impacts the performance of downstream applications. This limitation has a significant impact in fields such as urban planning and traffic management, as accurate estimates of hourly crowd flows are crucial for effective scheduling and responsive urban management [10]. As a result, trajectory recovery is essential for improving the accuracy and utility of applications in these domains.

Manuscript received 21 March 2025; revised 12 December 2025; accepted 21 May 2026. (Corresponding authors: Huandong Wang; Yong Li.)

Qingyue Long, Can Rong, Huandong Wang, and Yong Li 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)

Rajib Shaw is with the Graduate School of Media and Governance, Keio University, Kanagawa 252-0882, Japan (e-mail: shaw@sfc.keio.ac.jp)

Formally, the task of trajectory recovery involves restoring missing trajectory records based on existing partial trajectory data. Throughout the evolution of the technology, traditional methods have primarily relied on rule-based frameworks [11], [12], [13], [14], [15], with the fundamental assumption that human mobility patterns can be described by physical rules (such as movement speed constraints or geographically reachable areas). These methods focus on modeling basic features, such as analyzing the correlation between neighboring trajectory points using the principle of spatial continuity [16], [17], or predicting recurring trajectory patterns based on time periodicity assumptions [18]. While these approaches may be feasible in simple scenarios, their simplified modeling often fails to fully capture the diversity and complexity of real-world human mobility, particularly when capturing personalized behaviors, which can result in significant bias. With technological advances, data-driven methods have gradually become mainstream in research. These methods break free from the limitations of traditional rules and use tools such as deep learning models [19], attention mechanisms [20], and variational autoencoders [19] to automatically extract multidimensional features from vast amounts of movement data, including implicit spatiotemporal correlations, behavioral habits, and higher-order semantic patterns. This end-to-end learning paradigm significantly enhances the model's ability to adapt to complex scenarios, thereby improving the accuracy and robustness of trajectory recovery. Nevertheless, these data-driven methods focus on regular mobility patterns and often neglect the specific mobility preferences embedded in trajectory data, which can significantly affect the quality of trajectory recovery.

Therefore, developing a trajectory recovery method that simultaneously integrates both group commonalities and individual characteristics has become a key solution to enhance the practical value of trajectory recovery. However, constructing such a method faces the following critical challenges:

- **Excessive sparsity in individual trajectories severely limits the effective use of historical information for recovery.** When trajectories are overly sparse, spatiotemporal dependencies can only be inferred from a limited number of data points, which impedes the accurate identification of past mobility patterns. As a result, this limitation significantly complicates the precise recovery of trajectories.
- **Sparse trajectories make it harder to capture the dynamic and varied mobility preferences of individuals.** People's preferences for different locations can differ greatly, often

reflecting their attachment to specific places. However, individuals typically weigh multiple factors when choosing travel destinations. For example, a person planning to shop may prioritize malls and compare their preferences among them. This interplay of preferences and behaviors poses a significant challenge for accurate trajectory modeling.

To address the above challenges, we propose a method named **DiffMove**. We build our framework based on diffusion models because trajectory recovery is inherently not a deterministic point prediction task, but a problem of modeling the conditional distribution of user mobility behaviors. In real-world scenarios, mobility exhibits high uncertainty, while traditional methods (e.g., AttnMove) typically output a single estimated point. In contrast, generative models can learn the full distribution and better preserve the uncertainty and diversity of mobility behaviors. Second, among generative modeling approaches, we choose diffusion models rather than GANs or VAEs mainly due to considerations of stability and generation quality. GANs are prone to mode collapse when handling high-dimensional spatiotemporal sequences, and VAEs often produce overly smooth and blurry outputs. Diffusion models, through noise injection and step-by-step denoising, provide a more stable training pathway and are better suited for high-dimensional data. **DiffMove** consists of three critical modules, *i.e.*, 1) group tendency extraction module, 2) individual preference extraction module, and 3) mobility behavior diffusion module. In the group tendency extraction module, we construct a group tendency graph using all trajectories. This graph uses the transition frequency between locations as the edge weight, representing the degree of group tendency towards each location. Then, we integrate this group tendency information into the location embeddings through graph embedding techniques, considering both first-order and second-order proximities. This approach effectively addresses the first challenge. Given the potential noise and sparsity in trajectories, the individual preference extraction module improves accuracy by incorporating historical trajectories related to the current trajectory to capture individual mobility preferences better. We first extract historical preferences from past trajectories using a history processor and current preferences from current trajectories with a current processor. These preferences are then combined using the proposed inter-trajectory attention mechanism. In the mobility behavior diffusion module, we use a carefully designed diffusion model to integrate individual mobility preferences into the distribution modeling of mobility behaviors. This step ensures the precise capture of individual mobility preferences, thus addressing the second challenge. Given that group mobility tendency and individual mobility preferences influence mobility behavior, we aim to recover high-quality trajectories by integrating both factors. Unlike prior mobility modeling frameworks such as graph-enhanced predictors (e.g., PeriodicMove [21]), user behavior recovery models, and diffusion-based trajectory generators (e.g., TrajGDM [22]), DiffMove is explicitly designed for trajectory recovery and formulates the task as conditional inference rather than deterministic prediction or unconditional generation. DiffMove integrates both group mobility priors and individual preference representations into the diffusion process, enabling behavior-consistent reconstruction of missing trajectory segments.

Overall, our contributions can be summarized as follows:

- We propose a group mobility tendency enhanced trajectory recovery method named DiffMove, which leverages a diffusion model to simultaneously capture group tendencies and individual preferences.
- To capture group mobility tendencies, we construct a group tendency graph and integrate this information into location representations using graph embedding techniques. For modeling individual mobility preferences, we extract preferences from both historical and current trajectories. Finally, leveraging a carefully designed diffusion model, we effectively integrate group tendencies and individual preferences, thereby modeling the distribution of mobility behavior.
- Extensive experiments on three real-world datasets demonstrate the effectiveness of our DiffMove model. Further analysis confirms the robustness of our approach.

II. Related Work

A. Trajectory Recovery

Trajectory recovery has been studied from two main methodological directions. Rule-based approaches design physical or heuristic constraints, such as map-matching under road network topology or interpolation based on spatiotemporal continuity, and recover missing points accordingly. For instance, Xiao et al. developed a conditional random field framework that incorporates contextual relationships among cellular signaling records (CDRs) to reconstruct urban mobility paths [16]. Complementary strategies employ interpolation mechanisms, in which missing trajectory points are estimated via linear or Gaussian process regression, leveraging spatiotemporal correlations in mobility data [12] [18]. Data-driven approaches instead learn complex mobility patterns directly from data, using deep sequential models to capture nonlinear temporal dependencies and high-order behavioral correlations. For instance, Feng et al. developed a hybrid architecture integrating long short-term memory (LSTM) modules with periodic pattern recognition to reconcile historical trajectory regularities with real-time movement predictions [23]. The recently proposed Bi-STDDP model advances this paradigm by establishing bidirectional spatiotemporal attention mechanisms that jointly optimize user-specific behavioral vectors and geographic contextual features [24]. In addition to general human trajectory recovery methods, several recent studies focus on map-constrained trajectory reconstruction, primarily targeting vehicle mobility rather than human movement. MTrajRec introduces a seq2seq multi-task framework that jointly models road-segment sequences and travel distances, explicitly requiring a detailed road network as input [25]. Similarly, RNTrajRec leverages a spatial–temporal transformer augmented with road topology, utilizing constraints from road connectivity to reduce the search space of feasible paths [26]. More recently, MM-STGED employs a hierarchical micro–macro graph encoder–decoder to refine map-constrained trajectory reconstruction under strict road-network constraints [27].

In summary, existing trajectory prediction and recovery methods that incorporate graph structures or user behaviors have made substantial progress. However, most of them still produce deterministic point estimates and do not explicitly address missing-data inference under behavioral priors. Diffusion-based trajectory models such as TrajGDM [22] learn unconditional trajectory distributions but do not focus on the recovery of missing segments. This work follows a data-driven approach and targets trajectory recovery by learning a conditional diffusion model tailored to this task.

B. Diffusion Models

Most studies on diffusion models build upon Denoising Diffusion Probabilistic Models (DDPM), which employ variational inference. The fundamental idea of DDPM is to gradually modify the data distribution through a stepwise diffusion mechanism and subsequently reconstruct it by learning an inverse transformation. This method enables the development of a versatile and computationally efficient generative framework [28]. The diffusion model has shown impressive performance in image synthesis tasks [29], [30], [31], [32], [33], [34].

Due to its strong generative abilities, the diffusion model has expanded beyond image synthesis to audio synthesis. The DiffWave model is an efficient vocoder that delivers high-quality speech with a small footprint and is faster than real-time generation. It has also led to significant advancements in category-based labeling and unconditional audio generation [35]. The diffusion model has also found applications in time series modeling. For time series imputation, CSDI [36] introduced a novel interpolation method based on a fraction-based diffusion model. This approach leverages self-supervised training to optimize the model, allowing it to capture temporal correlations effectively. Similarly, SSSD [37] employs a conditional diffusion model in conjunction with a structured state space framework to address extended temporal dependencies in time series, demonstrating strong capabilities in both interpolation and prediction. In the context of time series forecasting, TimeGrad [38] proposes an autoregressive approach for generating multivariate probabilistic time series by sequential sampling from the underlying data distribution. This approach leverages a diffusion-based probabilistic model, drawing upon principles related to energy-based methods.

Recently, diffusion models have also been applied to human mobility modeling. ControlTraj [39] introduces topology-constrained diffusion for controllable trajectory generation, incorporating road network structure as guidance to produce realistic and map-consistent synthetic trajectories. CoDiffMob [40] proposes collaborative noise priors for urban mobility generation, leveraging shared spatial patterns among users to improve the quality of generated trajectories. While these works demonstrate the strong potential of diffusion models in mobility-related tasks, they primarily focus on unconditional trajectory generation rather than the recovery of missing segments from partial observations, which is the target of this work.

TABLE I
SUMMARY OF NOTATIONS.

Notation	Description
u	A user.
$l_u^{i,n}$	The location of user u within a specified time period on the n-th time slot of the i-th day.
$\mathcal{T}_u^i$, $\mathcal{T}_u^I$	The historical trajectory of user u on day i and the current trajectory of user u.
G	A group preference graph.
e_l	The embedding of location l.
e_t	The temporal representation of the n-th time interval.
$\bar{e}_u^{i,n}$	The temporal-aware representation corresponding to the n-th time interval within the i trajectory of user u.
$\alpha_{n,k}^{(h)}$	The degree of similarity computed by the h-th head between time interval n in the current trajectory and time interval k in the historical trajectory.
$\check{e}_u^{p,n}$	The historical shift-aware representation of the n-th time interval in the p-th trajectory of user u.
$\check{e}_u^{i,n}$	The current shift-aware representation of the n-th time interval in the i-th trajectory of user u.
e_u^n	The final trajectory representation of the n-th time interval.
e_{u0}^n	The clean final trajectory representation of the n-th time interval.
e_{ut}^n	The representation in a noising step t.
$\hat{e}_{u0}^n$	The clean sample.
$\mathcal{F}_\theta(\cdot)$	The denoising network.
$\mathcal{L}_{simple}$, $\mathcal{L}_d$	The simplified loss and distance-aware loss.
λ_d	The weight of distance loss.

Overall, diffusion models have generally performed well in time series modeling, audio synthesis, and picture synthesis. However, limited research explores the application of diffusion models to trajectory recovery. This task differs from general time series modeling as it requires the consideration of mobility-specific features inherent to trajectories.

III. PRELIMINARIES

This section presents the concept and process of the denoising diffusion probabilistic model after introducing the concepts and notations used in this work.

A. Problem definition

Definition 1 (Time Interval). Raw mobility trajectories in all three datasets are highly sparse due to their collection mechanisms: GeoLife relies on irregular GPS reporting triggered by device usage, the Tencent dataset is based on event-triggered cellular signaling logs, and Foursquare consists of user-initiated check-ins at points of interest. These heterogeneous collection mechanisms result in uncontrolled and uneven sampling intervals. Therefore, following common practice in sparse trajectory modeling (e.g., DeepMove [23], PeriodicMove [21], AttnMove [20]), we discretize the trajectories into a fixed 30-minute interval to ensure consistent temporal granularity across users and different days.

Definition 2 (Trajectory). A trajectory represents the sequence of locations visited by a user over time. The trajectory of user u on day i can be expressed as a sequence of spatiotemporal coordinates, denoted by $\mathcal{T}_u^i = \langle l_u^{i,1}, \ldots, l_u^{i,n}, \ldots, l_u^{i,N} \rangle$, where $l_u^{i,n}$ indicates the user's location at the n-th time slot within a specified interval (e.g., every 30 minutes), and N is the total number of time slots. It is important to note that if the location for a particular time slot n is not observed, the value $l_u^{i,n}$ is marked as *null*, signifying a missing location.

Definition 3 (Current and Historical Trajectory). Let I represent a target day, and $\mathcal{T}_u^I$ denote the trajectory of user u on that day. This trajectory $\mathcal{T}_u^I$, corresponds to the user's current path, while the historical trajectories are denoted as $\{\mathcal{T}_u^1, \mathcal{T}_u^2, \ldots, \mathcal{T}_u^{I-1}\}$, representing the user's movement on the days before day I.

Problem Statement (Trajectory Recovery). The goal is to reconstruct the full trajectory for the current day by filling in the missing locations or null values in $\mathcal{T}_u^I$, leveraging the historical trajectories $\{\mathcal{T}_u^1, \mathcal{T}_u^2, \ldots, \mathcal{T}_u^{I-1}\}$ associated with user u.

B. Denoising Diffusion Probabilistic Model

Unlike deterministic prediction tasks, trajectory recovery aims to model the conditional distribution of missing trajectory points. Human mobility is inherently uncertain, making single-point estimators insufficient. Intuitively, diffusion models address this by gradually corrupting data with Gaussian noise in a forward process and then learning to invert the corruption step by step, which provides a stable mechanism for approximating complex distributions and outperforms GANs (mode collapse) and VAEs (over-smoothing) when modeling high-dimensional spatiotemporal data.

Formally, a DDPM consists of two Markov processes. The forward process progressively injects Gaussian noise into the original sample and is defined as

$$q(\mathbf{x}_{1:T} \mid \mathbf{x}_0) := \prod_{t=1}^{T} \mathcal{N}\Big(\mathbf{x}_t;\ \sqrt{1-\beta_t}\,\mathbf{x}_{t-1},\ \beta_t \mathbf{I}\Big), \quad (1)$$

where β_t is a small positive constant controlling the noise intensity, so that $\mathbf{x}_T$ becomes nearly pure noise. By reparameterization, the sample at any step t admits the closed form $x_t = \sqrt{\bar{\alpha}_t}\,x_0 + \sqrt{1-\bar{\alpha}_t}\,\epsilon$, with $\alpha_t = 1-\beta_t$, $\bar{\alpha}_t = \prod_{i=1}^{t}\alpha_i$, and $\epsilon \sim \mathcal{N}(0, I)$.

The reverse process starts from $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and gradually removes noise to recover a clean sample $\mathbf{x}_0$. Each transition is modeled as a Gaussian with learnable parameters:

$$p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t) := \mathcal{N}(\mathbf{x}_{t-1};\ \boldsymbol{\mu}_\theta(\mathbf{x}_t, t),\ \sigma_\theta(\mathbf{x}_t, t)\mathbf{I}). \quad (2)$$

Following Ho et al. [28], $\boldsymbol{\mu}_\theta$ and σ_θ are reparameterized so that a trainable denoising network $\boldsymbol{\epsilon}_\theta$ directly predicts the noise vector injected during the forward process. Training therefore reduces to a simple noise-prediction objective:

$$\min_\theta \mathcal{L}(\theta) := \min_\theta \mathbb{E}_{\mathbf{x}_0 \sim q(\mathbf{x}_0), \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), t} ||\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta\left(\mathbf{x}_t, t\right)||_2^2, \quad (3)$$

which can be interpreted as a weighted variational bound on the negative log-likelihood of the data.

IV. METHOD

The framework illustrated in Figure 1 depicts our proposed DiffMove model, which consists of three main components: 1) a group tendency extraction module, 2) an individual preference extraction module, and 3) a mobility behavior diffusion module. We provide detailed descriptions of these components in this section. Unlike prior diffusion-based trajectory models such as TrajGDM [22], our framework introduces domain-specific adaptations tailored for trajectory recovery. Missing trajectories can be inferred from both group movement tendencies and individual preferences. The group tendency extraction module captures macroscopic mobility priors through graph-based learning, while the individual preference extraction module models both historical periodic preferences and current mobility intentions. These components jointly provide meaningful conditioning signals to guide the diffusion process.

A. Group Tendency Extraction Module

To address challenge 1, we propose to construct a **group tendency graph** and employ appropriate graph embedding techniques to accurately capture the collective tendencies of the population regarding location transitions (i.e., travel origins and destinations).

1) Graph Construction: The frequency of travel to a specific location reflects the group's tendency to visit that destination. For example, when a location appears frequently across multiple trajectories, it indicates a shared preference for that destination. To effectively model this collective behavior, we propose the construction of a group tendency graph, denoted as G, using all available trajectory data. Formally, G is defined as $G = (V, E)$, where V represents the set of nodes corresponding to all locations visited by the group, and E denotes the set of edges that represent the relationships between pairs of locations. Each edge $e \in E$ is an unordered pair $e = (a, b)$ with an associated weight $w_{ab} > 0$, where w_{ab} captures the frequency of transitions between locations a and b. This weight reflects the group's stronger inclination to travel between a and b. Through this approach, the group tendency graph G accurately represents the mobility preferences of the population across various locations.

2) Group Tendency Extraction with Graph Embedding: Since the edge weight w_{ab} in the group tendency graph reflects the number of transitions between locations a and b, which represents the population's inclination towards these destinations, we aim to incorporate this group tendency data into the location embeddings through a graph embedding module. However, the edges representing group tendencies may be sparse in the constructed graph. This sparsity arises from the common herd effect, in which large crowds primarily move within a few popular areas. As a result, transition frequencies between popular locations are generally higher, inflating the corresponding edge weights and posing challenges for graph learning. To mitigate this herd effect and better account for less frequented locations, we propose incorporating group tendency information from two perspectives: **first-order proximity** and **second-order proximity**.

To effectively perform embedding, it is essential to preserve the structure of the graph network. First, maintaining the local network structure, specifically the pairwise proximity between nodes, is critical. The first-order proximity between nodes can define the local graph structure:

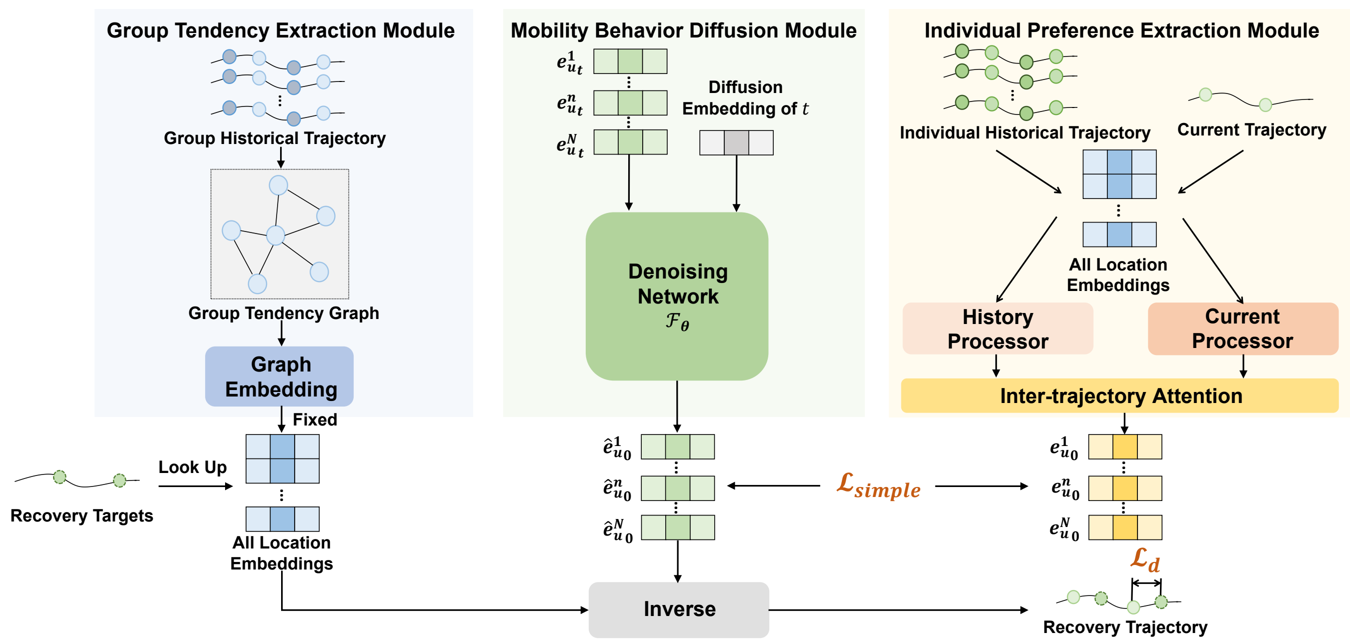


Fig. 1. The framework of DiffMove.

1) First-order Proximity. In the group tendency graph G, for every edge (a, b) between two nodes, the associated weight w_{ab} represents the first-order proximity between a and b. This weight quantifies the strength of the group's tendency to travel between these two locations, effectively capturing local relationships. However, due to the herd effect in group tendency modeling, where popular locations dominate, the graph suffers from sparsity. As a result, relying only on first-order proximity does not fully preserve the graph's structure. A natural extension is to consider nodes with common neighbors, as these nodes are likely to exhibit similar characteristics. To address the sparsity problem, we introduce second-order proximity.

2) Second-order Proximity. Second-order proximity between nodes (a, b) captures how closely their neighboring structures resemble each other. Formally, for a node a, we define its first-order relationships with all nodes as a vector $p_a = (w_{a,1}, \dots, w_{a,|V|})$. Then, the second-order proximity between nodes a and b is quantified by assessing the similarity of their respective vectors p_a and p_b. Incorporating second-order proximity highlights less frequently visited locations (nodes with lower edge weights) and addresses the sparsity challenge by modeling indirect neighborhood information.

3) Group Tendency Extraction. Next, we aim to integrate group tendency information into location embeddings using graph embedding techniques. Given the constructed group tendency graph $G = (V, E)$, the goal of graph embedding is to represent each vertex $v \in V$ as a vector in a lower-dimensional space $\mathbb{R}^d$, where $d \ll |V|$. Specifically, we adopt the LINE (Large-scale Information Network Embedding) method [41], a widely used approach for graph embedding in large-scale information networks, to incorporate group tendency information from both first-order and second-order proximity perspectives. For first-order proximity, we minimize the discrepancy between the predicted and actual edge weights to capture the group tendency information. To address second-order proximity, we perform random walks and negative sampling within the second-order proximity neighborhoods, helping to mitigate issues of sparsity and the herd effect in trajectory data. We separately optimize these two objectives, representing first-order and second-order proximity, using the LINE method. Finally, we combine the embeddings derived from both perspectives to obtain a consolidated embedding for each location. Using the group tendency graph G, the graph embedding process with LINE can be expressed as follows:

$$e_l = \mathrm{LINE}_\theta(G), \tag{4}$$

For each location $l \in L$, we assign a representation vector $e_l \in \mathbb{R}^d$. These embeddings collectively form the embedding matrix $E_l \in \mathbb{R}^{|L+1| \times d}$.

In conclusion, we build a group tendency graph based on the available trajectory data. We then utilize the LINE method for graph embedding, incorporating group tendency information into location embeddings by considering both first-order and second-order proximity relationships.

B. Individual Preference Extraction Module

To address challenge 2, we propose **trajectory processor** and **inter-trajectory attention** to accurately capture individuals' preferences for visiting locations or mobility patterns.

1) Trajectory Processor: The trajectory processor consists of two main parts: the history processor and the current processor. The history processor analyzes multiple past trajectories to derive periodic mobility preferences for each individual. On the other hand, the current processor focuses on identifying the individual's present preferences for location selection, emphasizing the types of places they tend to visit most frequently.

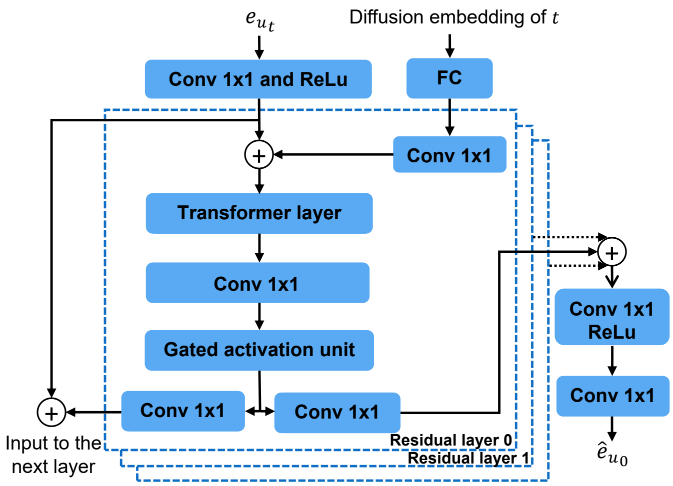


Fig. 2. The architecture of denoising network $\mathcal{F}_\theta$.

History Processor. The mobility patterns of each user are complex and variable, making it vital to capture their individual preferences accurately from their trajectories. However, a single trajectory is typically too sparse to reveal sufficient details about a user's favored destinations and mobility behaviors. As a result, we use the history processor to effectively model individual mobility preferences by leveraging comprehensive, long-term historical trajectory data.

To capture spatio-temporal dependencies, we jointly embed time and location into a dense representation as input. Specifically, for each location $l \in L$, we define an embedding vector $e_l \in \mathbb{R}^d$, obtained by constructing a group tendency graph and applying graph embedding through the group tendency extraction module in Section IV-A. Temporal information also plays a key role in mobility modeling. Following [42], we generate an embedding for each time slot t as follows:

$$\begin{cases} e_t(2i) = \sin\left(t/10000^{2i/d}\right), \\ e_t(2i+1) = \cos\left(t/10000^{2i/d}\right), \end{cases} \tag{5}$$

where the index i indicates the specific dimension, the embedding vectors representing time have the same dimensionality d as those representing locations.

We then obtain a temporal-aware embedding for the n-th time interval of u's i-th trajectory, $\bar{e}_u^{i,n} \in \mathbb{R}^d$, by element-wise addition of the time and location vectors:

$$\bar{e}_u^{i,n} = e_l + e_t, \tag{6}$$

which encodes both where and when the user is located in a compact d-dimensional space.

Since a single historical trajectory is too sparse to reveal periodicity, we design a history aggregator that compresses multiple sparse days into a single prototype trajectory by retaining, at each time interval, the most frequently visited location:

$$\mathcal{T}_u^p = \mathcal{T}_u^1 \oplus \mathcal{T}_u^2 \cdots \oplus \mathcal{T}_u^{i-1}, \tag{7}$$

where $\oplus$ denotes most-frequent-location extraction. The aggregated $\mathcal{T}_u^p$ is then embedded as $\bar{e}_u^{p,n}$ following Eq. (6).

If certain users have no recorded data for a given time slot, aggregation alone cannot fill in the missing locations. We therefore apply a standard multi-head self-attention mechanism over time slots to infer missing entries from other observations. Under head h, the attention weight and the aggregated output at time n are

$$\alpha_{n,k}^{(h)} = \mathrm{softmax}_k\left(\left\langle W_Q^{1(h)}\bar{e}_u^{p,n},\, W_K^{1(h)}\bar{e}_u^{p,k}\right\rangle\right), \tag{8}$$

$$\tilde{e}_u^{p,n(h)} = \sum_{k=1}^{N} \alpha_{n,k}^{(h)}\, W_V^{1(h)}\bar{e}_u^{p,k}, \tag{9}$$

where $W_Q^{1(h)}, W_K^{1(h)}, W_V^{1(h)} \in \mathbb{R}^{d' \times d}$ are learnable projections. The shift-aware embedding is then obtained by concatenating the H heads and adding a residual connection:

$$\check{e}_u^{p,n} = \mathrm{ReLU}\left(\left[\tilde{e}_u^{p,n(1)} \mid \cdots \mid \tilde{e}_u^{p,n(H)}\right] + W^1\bar{e}_u^{p,n}\right), \tag{10}$$

where $W^1 \in \mathbb{R}^{d'H \times d}$ and $\mid$ denotes concatenation. This design allows each time interval to attend to its neighbors and capture long-range periodic dependencies in the user's historical trajectories.

By leveraging the complete history processor, we obtain the vector $\check{e}_u^{p,n}$, which encapsulates the mobility characteristics of each individual historical trajectory.

Current Processor. Users move between various locations throughout the day, with visit frequency and duration reflecting different preference levels. For example, long stays at home or work versus brief visits to shopping or entertainment venues. The current processor models these preferences from the current trajectory $\mathcal{T}_u^i$.

We first embed $\mathcal{T}_u^i$ into $\bar{e}_u^i$ using the same mobility embedding, and then apply the attention mechanism in Eqs. (8)–(9) with the historical input $\bar{e}_u^p$ replaced by $\bar{e}_u^i$ and with separate projection matrices W_Q^2, W_K^2, W_V^2, W^2, yielding the refined current representation $\check{e}_u^{i,n}$.

2) Inter-trajectory Attention: After comprehensively capturing an individual's past and present preferences, the inter-trajectory attention mechanism evaluates the degree to which the reconstructed trajectory depends on interpolating the observed location versus integrating candidate values derived from historical movements.

The similarity between the current and historical trajectories, denoted as α, is defined based on the correlation between their enhanced representations at corresponding time slots, specifically $\check{e}_u^{i,n}$ and $\check{e}_u^{p,k}$ for all $n, k = 1, 2, \ldots, N$. The historical candidates are then aggregated according to α, with a residual connection incorporated to retain the original interpolation results. The network structure remains consistent with Eqs. (8)–(9), employing the projection matrices W_Q^3, W_V^3, W_K^3, and W^3. Ultimately, the refined representation for the n-th time slot, denoted as e_u^n, is obtained by integrating $\check{e}_u^{p,k}$ and $\check{e}_u^{i,n}$.

C. Mobility Behavior Diffusion Module

Due to the variability and dynamic nature of users' mobility patterns, their immediate preferences often exhibit uncertainty. Thus, representing current mobility behavior as a distribution provides a more effective modeling approach. The diffusion-based method systematically refines this distribution in an iterative manner, enabling a precise characterization of users' mobility behaviors.

Our objective is to capture the distribution of mobility behavior, which is learned through a diffusion model. Specifically, for each time slot's final representation e_u^n, the diffusion process can be expressed using a Markov chain, denoted as (e_{u0}^n, e_{u1}^n, ..., e_{ut}^n, ..., e_{uT}^n), with T indicating the total diffusion steps. The process involves progressively introducing Gaussian noise into the spatial and temporal components from e_{u0}^n to e_{uT}^n, eventually transforming them into pure Gaussian noise. The diffusion follows the probabilistic framework given by:

$$q\left(e_{u_t}^n \mid e_{u_{t-1}}^n\right) := \mathcal{N}\left(e_{u_t}^n; \sqrt{1-\beta_t} e_{u_{t-1}}^n, \beta_t I\right), \tag{11}$$

where $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{i=1}^t \alpha_t$.

Rather than directly reconstructing the point e_u^n, we formulate its recovery as a reverse denoising process, iteratively refining $e_{u_T}^n$ back to $e_{u_0}^n$. The complete reverse denoising procedure is expressed as follows:

$$p_\theta\left(e_{u_{0:T}}^n\right) := p\left(e_{u_T}^n\right) \prod_{t=1}^{T} p_\theta\left(e_{u_{t-1}}^n \mid e_{u_t}^n\right), \tag{12}$$

We use a denoising network structure $\mathcal{F}_{\boldsymbol{\theta}}(\cdot)$ similar to DiffWave [35] and CSDI [36] to generate $\hat{e}_{u_0}^n$(Figure 2).

At the final stage of the reverse process, an additional transformation is applied to map the continuous embedding $e_{u_0}^n$ back to the discrete location ID l_i.

D. Training and Sampling

1) Training: Following the approach proposed in [28], we adopt an alternative formulation for signal prediction instead of directly using ϵ_t, as outlined in [43]: $\hat{e}_{u_0}^n = \mathcal{F}_\theta(\hat{e}_{u_t}^n, t)$ with the simple objective [28] as follows,

$$\mathcal{L}_{\text{simple}} = \mathbb{E}_{e_{u_0}^n \sim q(e_{u_0}^n), t \sim [1,T]} \parallel e_{u_0}^n - \mathcal{F}_\theta(\hat{e}_{u_t}^n, t) \parallel_2^2 . \tag{13}$$

Human mobility exhibits inherent regularities, such as spatial continuity. To enforce this property, we introduce a distance-aware loss function, $\mathcal{L}_d$, which constrains the displacement between consecutive mobility steps. Specifically, it measures the Euclidean distances between adjacent points in a trajectory and promotes spatial continuity by minimizing their cumulative sum, as formulated below:

$$\mathcal{L}_d = \frac{1}{N-1} \sum_{n=1}^{N-1} \parallel l_u^{i,n+1} - l_u^{i,n} \parallel_2^2 . \tag{14}$$

Overall, our training loss is

$$\mathcal{L} = \mathcal{L}_{simple} + \lambda_d \mathcal{L}_d, \tag{15}$$

where λ_d is the weighting factor of distance-aware loss of $\mathcal{L}_d$.

Overall, Eqs. (11)–(15) describe how we apply the diffusion model to the trajectory representations. The forward process in Eq. (11) gradually injects Gaussian noise into the mobility representation of each time slot. In contrast, the reverse Markov chain in Eq. (12) learns to denoise these noisy representations back to clean ones. The network F_θ predicts the clean signal at each step, and the simple loss in Eq. (13) encourages accurate reconstruction of the underlying mobility pattern. The distance-aware term in Eq. (14) further enforces spatial continuity by penalizing unrealistically large jumps between consecutive locations, and the combined objective in Eq. (15) balances distribution learning with trajectory smoothness.

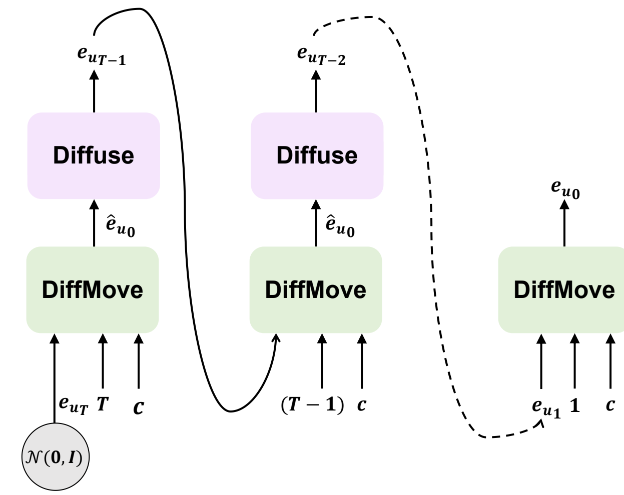


Fig. 3. The sampling process of DiffMove.

TABLE II
BASIC STATISTICS OF MOBILITY DATASETS.

Dataset	City	Duration	#Users	#Loc	#Traj
Geolife	Beijing	5 years	40	3439	896
Tencent	Beijing	1 month	4246	8998	39422
Foursquare	New York	10 months	315	2435	2146

2) Sampling: As described in [28], the generation of $p(e_{u0}^n)$ follows an iterative process. At each timestep t, the model predicts a clean estimate, $\hat{e}_{u0}^n = \mathcal{F}_\theta(\hat{e}_{ut}^n, t)$, which is then perturbed to obtain e_{ut-1}^n. This procedure is repeated from $t = T$ down to $t = 0$, progressively refining the representation until e_{u0}^n is reconstructed, as illustrated in Figure 3.

V. EXPERIMENTS

A. Dataset

We conducted extensive experiments on three real-world datasets: Geolife, Tencent, and Foursquare.

- **Geolife** [44]: This dataset is compiled by Microsoft Research Asia, which captures the mobility patterns of 182 users from April 2007 to August 2012. Each trajectory represents a sequence of time-stamped GPS points, encompassing latitude, longitude, and altitude data.
- **Tencent**: The dataset is collected from Tencent, a prominent social network and location service provider in Beijing, China, during the period from June 1 to June 30, 2018. The trajectory data is derived from GPS locations that are recorded whenever users access location services through the application.
- **Foursquare** [45]: This dataset contains location-based check-in records collected from Foursquare users in New York City.

TABLE III
PERFORMANCE COMPARISON ON THE GEOLIFE DATASET. THE BEST RESULT IN EACH COLUMN IS IN BOLD, WHILE THE SECOND IS UNDERLINED.

	Recall@1	Recall@5	Recall@10	MAP	Distance(m)	Grid Accuracy	CRPS
Linear	0.3624	0.5682	0.6521	0.3843	**2345**	0.1284	–
History	0.2518	0.5012	0.5786	0.2701	5107	0.1159	–
Top	0.2732	0.5188	0.5965	0.2798	5315	0.1174	–
RF	0.2589	0.5123	0.5904	0.2554	6107	0.1197	–
LSTM	0.2808	0.5462	0.6248	0.3472	5746	0.1438	–
DeepMove	0.3403	0.5572	0.6354	0.4123	4858	0.1527	–
BiLSTM	0.3528	0.5639	0.6411	0.4195	5021	0.1584	–
Bi-STDDP	0.3765	0.5871	0.6718	0.4514	4032	0.1694	–
AttnMove	0.3982	0.6417	0.7105	0.4691	3886	0.1756	–
PeriodicMove	0.4074	0.6613	0.7542	0.4805	3527	0.1983	–
BeTR	0.3776	0.6534	0.7282	0.4365	4078	0.1821	–
TrajMoE	0.3852	0.6619	0.7398	0.4443	3945	0.2145	–
TERI	0.3918	0.6724	0.7481	0.4521	3820	0.1887	–
STR	0.4021	0.6894	0.7649	0.4683	3570	0.2296	–
TrajBERT	0.4069	0.6987	0.7725	0.4752	3457	<u>0.2641</u>	–
ARMove	0.4124	0.7065	0.7804	0.4829	3320	0.2563	–
CLMove	<u>0.4180</u>	<u>0.7138</u>	<u>0.7826</u>	<u>0.4905</u>	3201	0.2484	–
TrajGDM	0.4016	0.6845	0.7612	0.4742	3705	0.2052	0.2417
GenMove	0.3975	0.6812	0.7563	0.4608	3692	0.2018	<u>0.2143</u>
UniMob	0.3931	0.6740	0.7509	0.4718	3548	0.2235	0.2274
BCDiff	0.3891	0.6707	0.7437	0.4626	3619	0.2163	0.2304
DiffMove	**0.4272**	**0.7243**	**0.7921**	**0.5017**	<u>3126</u>	**0.2785**	**0.2014**
Improvement	2.2%	1.5%	1.2%	2.3%	-	5.5%	6.0%

Each trajectory consists of time-stamped check-ins associated with specific POIs, including latitude and longitude coordinates.

Pre-processing: To effectively represent spatial information, the city area is segmented into 8998 blocks using Beijing's road infrastructure, with each block representing a unique location averaging approximately 0.265 km^2. For the Foursquare dataset, we follow common practice in POI-based trajectory modeling and treat each unique POI as a discrete location identifier, resulting in 2,435 distinct locations within the New York City area. Fixed time intervals have been widely applied in trajectory modeling (e.g., AttnMove [20], PeriodicMove [21]) by identifying stay points to specify a representative location within each time interval. This approach is extensively used in trajectory data preprocessing and aligns with the stay-move movement pattern demonstrated in prior human mobility studies [13], [46], [47]. Consistent with previous research [48], [23], the temporal resolution adopted is 30 minutes. To maintain dataset reliability, the Tencent dataset excludes trajectories containing less than 34 intervals (around 70% coverage of a single day). It removes users who lack trajectory records spanning at least five days. For the Geolife dataset, trajectories with fewer than 12 intervals and users who do not have a minimum of five days of mobility data are similarly discarded. Table II summarizes the processed mobility datasets.

B. Baselines

We evaluate some baselines on three mobility datasets, encompassing both rule-based and data-driven approaches. Linear, History, Top, and RF are rule-based algorithms designed for trajectory recovery, leveraging insights into human mobility patterns.

- **Linear** [12]: This approach assumes that missing trajectory points can be estimated using a linear function. As a rule-based method, it relies on a strong underlying assumption about mobility patterns.
- **History** [49]: This approach straightforwardly leverages historical trajectories by assigning the most frequently visited location at each time slot as the recovered position.
- **Top** [50]: This approach determines the recovery location for each user by selecting the most frequently visited location across the entire training set, making it a frequency-based method.
- **RF** [49]: The random forest method leverages various trajectory features for recovery. A random forest classifier is trained using attributes such as trajectory entropy, radius, missing time gaps, and the spatial locations before and after these gaps.

The remaining baselines are data-driven approaches that leverage machine learning models to capture more complex mobility features:

- **LSTM** [50]: This approach employs mobility prediction for trajectory recovery by leveraging LSTM networks to model sequential transitions, enabling the inference of the next trajectory point.
- **BiLSTM** [51]: This approach enhances the standard LSTM network by incorporating a bidirectional structure, allowing it to utilize historical and future observations. Leveraging all available locations as spatiotemporal constraints improves trajectory point recovery beyond the limitations of unidirectional modeling.
- **DeepMove** [23]: This trajectory recovery method is based on mobility prediction, utilizing a multimodal embedding recurrent neural network. By jointly embedding multiple factors influencing human movement, it effectively captures

TABLE IV
PERFORMANCE COMPARISON ON THE TENCENT DATASET. THE BEST RESULT IN EACH COLUMN IS IN BOLD, WHILE THE SECOND IS UNDERLINED.

	Recall@1	Recall@5	Recall@10	MAP	Distance(m)	Grid Accuracy	CRPS
Linear	0.6204	0.7453	0.8106	0.6532	1182	0.1973	–
History	0.4702	0.6027	0.6841	0.4896	1687	0.1834	–
Top	0.5895	0.7219	0.7925	0.6146	2513	0.1925	–
RF	0.4832	0.6184	0.7013	0.4904	8138	0.1881	–
LSTM	0.6178	0.7356	0.8042	0.6875	3586	0.2327	–
DeepMove	0.7302	0.8127	0.8679	0.7904	1289	0.2443	–
BiLSTM	0.7005	0.7984	0.8537	0.7823	1354	0.2574	–
Bi-STDDP	0.7105	0.8049	0.8611	0.7872	1107	0.2685	–
AttnMove	0.7646	0.8423	0.8897	0.8249	934	0.2812	–
PeriodicMove	0.7893	0.8664	0.9138	0.8501	786	0.3112	–
BeTR	0.7824	0.8581	0.9036	0.8465	792	0.2971	–
TrajMoE	0.7883	0.8647	0.9094	0.8503	774	0.3047	–
TERI	0.7945	0.8719	0.9158	0.8564	763	0.3184	–
STR	0.8034	0.8816	0.9254	0.8658	<u>702</u>	0.3311	–
TrajBERT	0.8091	0.8874	0.9301	0.8726	728	0.3386	–
ARMove	0.8152	0.8936	0.9365	<u>0.8801</u>	713	<u>0.3535</u>	–
CLMove	<u>0.8197</u>	<u>0.8951</u>	<u>0.9402</u>	0.8794	744	0.3547	–
TrajGDM	0.7751	0.8512	0.8972	0.8393	814	0.2894	0.2135
GenMove	0.7992	0.8765	0.9207	0.8612	752	0.3249	<u>0.1914</u>
UniMob	0.8023	0.7895	0.9184	0.8645	783	0.3451	0.2207
BCDiff	0.7862	0.8603	0.9056	0.8481	803	0.3024	0.2261
DiffMove	**0.8457**	**0.9159**	**0.9548**	**0.8945**	**681**	**0.3694**	**0.1783**
Improvement	3.2%	2.3%	1.6%	1.6%	3.0%	4.1%	6.8%

complex sequential transitions.

- **Bi-STDDP** [24]: This approach simultaneously models spatiotemporal dependencies and user preferences. By leveraging the locations visited immediately before and after the target time period, it effectively reconstructs missing trajectory points.
- **AttnMove** [20]: This method incorporates both intra-trajectory and inter-trajectory attention mechanisms to enhance the modeling of user movement patterns. Additionally, it exploits periodic trends in long-term historical data to improve trajectory recovery.
- **PeriodicMove** [21]: This methodology leverages an attention mechanism integrated with graph neural networks to effectively model complex location transitions, hierarchical periodic behaviors, and temporal variations inherent in human movement patterns.
- **TERI** [52]: It adopts a multi-task prediction framework to model temporal dynamics in sparse trajectories, achieving robust trajectory recovery.
- **TrajBERT** [53]: It applies a BERT-style masked reconstruction framework with spatiotemporal attention to capture contextual dependencies in implicit sparse trajectories, enabling high-quality missing-point recovery.
- **STR** [54]: This work incorporates semantic behavior patterns into the trajectory recovery process, enabling more accurate reconstruction of human mobility from sparse observations.
- **BeTR** [55]: This method models delivery-specific behavioral cues using multi-scale attention fusion to improve sparse trajectory recovery in last-mile logistics scenarios.
- **TrajMoE** [56]: TrajMoE employs a spatially-aware mixture-of-experts design to capture heterogeneous mobility patterns and improve trajectory prediction and recovery performance.
- **ARMove** [57]: This method enhances trajectory recovery by explicitly modeling fine-grained spatial–temporal dependencies within the recommendation context.
- **CLMove** [58]: This work leverages contrastive learning to learn robust trajectory representations, achieving improved recovery accuracy from sparse mobility data.
- **GenMove** [59]: This work proposes a unified trajectory modeling framework using masked conditional diffusion to handle diverse mobility tasks under one generalizable model.
- **TrajGDM** [60]: This approach employs a diffusion-based trajectory modeling framework to learn the underlying mobility patterns within a trajectory dataset.
- **UniMob** [61]: This method employs a diffusion model to jointly model trajectory and flow patterns for unified human mobility prediction.
- **BCDiff** [62]: This method leverages two coupled diffusion models to capture bidirectional temporal dependencies for trajectory prediction.

C. Experimental Settings

To enhance training and evaluation while accounting for unknown missing patterns, 20% of the observations in three datasets are randomly masked as recovery targets. The data is then partitioned chronologically, with the first 70% used for training, the next 10% for validation, and the final 20% for testing.

For evaluation, we employ several key metrics, including *Recall@k*, *MAP*, *Grid Accuracy*, *CRPS*, and *Distance*. *Recall@k* measures the model's ability to recover the ground-truth locations within the top-k predictions, reflecting how well the model ranks candidate locations; higher values indicate

TABLE V
PERFORMANCE COMPARISON ON THE FOURSQUARE DATASET. THE BEST RESULT IN EACH COLUMN IS IN BOLD, WHILE THE SECOND IS UNDERLINED.

	Recall@1	Recall@5	Recall@10	MAP	Distance(m)	Grid Accuracy	CRPS
Linear	0.1105	0.1963	0.2728	0.2124	3085	0.0937	–
History	0.0507	0.1421	0.2056	0.1013	4239	0.0819	–
Top	0.0803	0.1594	0.2175	0.1316	3628	0.0884	–
RF	0.0652	0.1507	0.2102	0.1159	3905	0.0851	–
LSTM	0.1340	0.2413	0.3276	0.2637	2732	0.1129	–
DeepMove	0.2716	0.4530	0.5483	0.3753	2438	0.1285	–
BiLSTM	0.2301	0.4018	0.4925	0.3264	2582	0.1342	–
Bi-STDDP	0.2807	0.4659	0.5578	0.3625	2307	0.1408	–
AttnMove	0.2914	0.5026	0.5707	0.3961	2225	0.1493	–
PeriodicMove	0.3102	0.5214	0.6180	0.4371	2019	0.1769	–
BeTR	0.3033	0.5142	0.6056	0.4285	2130	0.1627	–
TrajMoE	0.3074	0.5189	0.6118	0.4328	2095	0.1695	–
TERI	0.3297	0.5406	0.6304	0.4552	2013	0.1842	–
STR	0.3448	0.5562	0.6469	0.4697	1998	0.1996	–
TrajBERT	0.3521	0.5654	0.6557	<u>0.4804</u>	1889	0.2074	–
ARMove	0.3643	0.5620	0.6325	0.4473	<u>1852</u>	0.2148	–
CLMove	<u>0.3781</u>	<u>0.6028</u>	<u>0.6613</u>	0.4786	1984	**0.2297**	–
TrajGDM	0.2958	0.5079	0.5987	0.4206	2175	0.1554	<u>0.2753</u>
GenMove	0.3375	0.5487	0.6392	0.4628	2007	0.1917	0.3487
UniMob	0.3417	0.5529	0.6430	0.4728	1905	0.2017	0.2840
BCDiff	0.3245	0.5376	0.6281	0.4608	1917	0.2041	0.3018
DiffMove	**0.3958**	**0.6224**	**0.6894**	**0.4936**	**1750**	<u>0.2219</u>	**0.2521**
Improvement	4.7%	3.3%	4.2%	2.7%	5.5%	-	8.4%

more effective recovery. *MAP* evaluates the overall ranking quality by assessing how accurately the model orders possible locations, with higher values indicating superior performance. *Grid Accuracy* quantifies the proportion of test instances for which the predicted grid cell exactly matches the ground-truth grid cell, offering an interpretable measure of discrete spatial correctness. *CRPS* quantifies the discrepancy between the predicted and ground-truth probability distributions; lower values indicate more calibrated and reliable probabilistic predictions. Finally, *Distance* quantifies the spatial deviation between the predicted location center and the actual location, with smaller values indicating higher positional accuracy.

D. Experiment Results

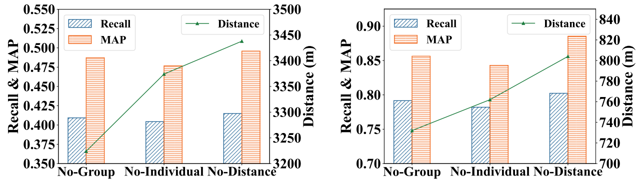
1) Overall Performance: To evaluate the reconstructed mobility trajectories, we utilize seven evaluation metrics and present the outcomes in Table III, IV and V. Based on these experimental results, we derive the following insights:

Firstly, rule-based methods consistently underperform across all three metrics. While they offer high interpretability by modeling mobility patterns with explicit physical meanings, they struggle to capture intricate, time-dependent, and higher-order mobility features that cannot be strictly defined through predefined rules.

Secondly, data-driven approaches achieve notable improvements by directly learning mobility patterns from real-world data without relying on rigid assumptions. This enables them to better model complex mobility behaviors. Methods that effectively incorporate historical trajectories or utilize attention mechanisms show particularly strong performance, especially in cases of sparse trajectory data. Since sequential dependencies may be weak in such scenarios, all observed locations should be considered equally. For instance, if a user is recorded at 7:00 AM and 3:00 PM, accurately inferring their 9:00 PM location may require prioritizing the 7:00 AM observation, as it is more likely to correspond to a residential location rather than a commuting destination.

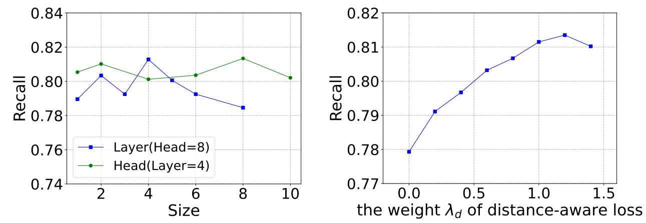
Thirdly, on the Geolife dataset, our method improves Recall@1 and MAP by 2.2% and 2.3% over the best baseline, respectively, and improves Grid Accuracy and CRPS by 5.5% and 6.0%. Although its Distance metric does not surpass the linear model, this may be due to the dataset's small size. However, the improvements in the other metrics confirm its effectiveness in trajectory recovery. On the Tencent dataset, our approach surpasses the best baselines in Recall@1, MAP, Grid Accuracy, and CRPS by 3.2%, 1.6%, 4.1%, and 6.8%, respectively, while also reducing Distance by 3.0%. These consistent gains across all seven metrics confirm the effectiveness of jointly modeling group tendencies and individual preferences on a large-scale urban mobility dataset. On the Foursquare dataset, our method achieves improvements of 4.7% in Recall@1, 2.7% in MAP, 5.5% in Distance, and 8.4% in CRPS over the best baselines. The only metric on which DiffMove does not lead is Grid Accuracy (0.2219 vs. CLMove's 0.2297); we attribute this to CLMove's contrastive representation learning, which is particularly well suited to discrete grid-level matching on POI-based check-in data. Nevertheless, DiffMove maintains a clear and consistent advantage across the remaining six metrics, demonstrating its strong overall performance on sparse check-in trajectories. The lower performance gain on Geolife compared to Tencent and Foursquare is likely attributed to its smaller user base, which makes it more challenging to extract periodic patterns and leverage group-level mobility features for individual trajectory recovery.

Overall, our method outperforms both rule-based and data-



(a) Geolife (b) Tencent

Fig. 4. Impact of each component on two datasets.



(a) Head and layer (b) The weight of distance-aware loss

Fig. 5. Sensitivity of hyper-parameters.

TABLE VI
PERFORMANCE COMPARISON OF GROUP TENDENCY VARIANTS IN DIFFMOVE ON THE GEOLIFE DATASET.

Variant	Recall@1	MAP	Distance (m)
No-Group	0.410	0.488	3230
G-FirstOrder	0.413	0.492	3209
G-Node2Vec	0.418	0.495	3181
G-SecondOrder	0.421	0.497	3165
G-GraphCL	0.422	0.498	3132
G-GRACE	0.425	0.500	3139
DiffMove (G-LINE)	**0.427**	**0.502**	**3126**

driven approaches, demonstrating its effectiveness in integrating individual and group mobility patterns, capturing periodic behaviors from historical data, and utilizing spatiotemporal correlations in current trajectories for enhanced recovery. These advantages highlight its superior performance in trajectory reconstruction.

2) Ablation Study: To assess the contribution of each module, we performed ablation experiments by systematically removing them and analyzing the impact on performance, with results shown in Figure 4. "No-Group" eliminates the group preference extraction module, simplifying the generation of location embeddings. "No-Individual" removes the individual preference extraction module, applying diffusion and denoising directly to fixed location embeddings from the group preference module. "No-Distance Loss" excludes the distance-aware loss $\mathcal{L}_d$ from the objective function. Performance degradation confirms that removing the individual preference extraction module has the most significant impact, emphasizing its role in leveraging historical and current data to capture individual mobility preferences. In contrast, omitting the group preference module has a relatively minor effect, as the location embeddings it generates still enrich the individual preference module, enabling a more comprehensive integration of individual and group mobility patterns. Additionally, excluding $\mathcal{L}_d$ most notably affects the *Distance* metric, reinforcing its role in enforcing spatial continuity constraints by penalizing excessive deviations between consecutive trajectory points.

Table VI reports the performance of different Group Tendency variants on the Geolife dataset. Removing this module (No-Group) leads to a clear performance decline, indicating that group mobility tendencies provide effective priors for trajectory recovery. Using only first-order proximity (G-FirstOrder) yields limited gains, as Geolife exhibits significant sparsity and long-tail patterns, making direct transition frequencies insufficient to characterize spatial relationships. In contrast, second-order proximity (G-SecondOrder) better captures neighborhood structural similarity, resulting in higher recall and lower distance errors. Node2Vec [63] (G-Node2Vec), although capable of modeling indirect relations through random walks, tends to bias toward high-frequency nodes and is less stable on sparse mobility graphs compared with LINE's explicit first- and second-order proximity modeling. We further evaluate two contrastive graph learning methods. GraphCL [64] (G-GraphCL) constructs contrastive views via random edge dropping and node dropping, but this inevitably discards critical transition frequency information encoded in edge weights. GRACE [65] (G-GRACE) adopts finer-grained adaptive augmentation and outperforms GraphCL, yet its edge removal mechanism still distorts the relative distribution of transition frequencies. Both methods surpass Node2Vec but remain inferior to LINE. Overall, the full DiffMove model, which jointly incorporates first- and second-order proximity, achieves the best performance, demonstrating the necessity of combining direct transition preference and structural similarity for high-quality trajectory recovery. Although LINE is a classical method, its explicit optimization of pairwise transition frequencies aligns naturally with the group tendency graph semantics without any augmentation perturbation, confirming that semantic-aligned inductive biases can outperform more complex general-purpose methods in task-specific scenarios.

3) Sensitivity of Hyper-parameters: We analyzed the sensitivity of key hyperparameters, including the number of attention heads and layers, and the weight λ_d of the distance-aware loss. The results presented are based on the Tencent dataset, with similar trends observed in Geolife.

Figure 5(a) shows the variation of *Recall* when varying the number of layers from 1 to 8 while keeping the number of heads fixed at 8. *Recall* reaches its maximum at 4 layers (Recall $\approx$ 0.813) and then gradually decreases as the depth continues to grow, indicating that overly deep architectures introduce redundant capacity and slightly hurt generalization on sparse trajectory data. Similarly, when varying the number of heads from 1 to 10 with the number of layers fixed at 4, *Recall* peaks at 8 heads (Recall $\approx$ 0.813), which suggests that a sufficiently diverse multi-head attention is beneficial for modeling heterogeneous spatio-temporal mobility patterns. Based on these observations, we set the number of layers to 4 and the number of heads to 8.

For the weight λ_d of the distance-aware loss, increasing λ_d consistently improved the *Distance* metric, while *Recall* reached its optimal value at $\lambda_d = 1.2$. Therefore, we selected $\lambda_d = 1.2$ for the Tencent dataset.

TABLE VII
PERFORMANCE W.R.T MISSING RATIOS ON TENCENT DATASET.

Missing Rate		20%	40%	60%	80%
AttnMove	Recall	0.7646	0.7502	0.7267	0.7145
	MAP	0.8249	0.8056	0.7731	0.7628
	Distance	934	1012	1146	1195
DiffMove	Recall	0.8457	0.8301	0.8076	0.7823
	MAP	0.8945	0.8612	0.8428	0.8364
	Distance	681	745	802	835

4) Robustness Analysis: To evaluate the robustness of our model under varying missing ratios, we conducted experiments with results summarized in Table VII. Three key observations emerge from this analysis. First, as the missing ratio increases, the performance of both the best baseline, AttnMove, and our model declines across all three metrics, confirming the negative impact of missing data on trajectory recovery. Second, across all missing ratios, our model consistently outperforms AttnMove, demonstrating its ability to maintain superior performance under different levels of data sparsity. Finally, even with an 80% missing ratio, our model still surpasses AttnMove when tested with a 20% missing ratio, underscoring its robustness in handling highly sparse datasets.

The enhanced robustness of our model is attributed to its ability to jointly capture both group and individual preferences while leveraging a diffusion model to integrate these factors into mobility intention modeling. In contrast, AttnMove primarily relies on individual trajectory information without incorporating these additional design elements. In extreme cases where users have limited trajectory data, our model can still effectively infer mobility intentions by utilizing group-level preferences. By considering a broader spectrum of information, our approach significantly improves performance in scenarios with high data sparsity, reinforcing its robustness.

5) Efficiency Analysis: To further evaluate the deployability of different trajectory generation models in real-world applications, we compare representative methods in terms of parameter size and inference runtime. Table VIII summarizes the parameter counts and inference time of all models under a unified hardware setting (NVIDIA RTX 2080 Ti GPU).

First, regarding parameter size, DiffMove contains 9M parameters, which is at a moderate level—significantly smaller than the Mixture-of-Experts model TrajMoE (40M), the diffusion model UniMob (15M), and the full-step diffusion model TrajGDM (20M). This indicates that DiffMove maintains a compact architecture while preserving expressive capacity. In addition, the parameters of conventional Transformer-based models (such as TrajBERT, STR, and AttnMove) are generally in the range of 5M–10M, which is comparable to DiffMove.

Second, in terms of inference efficiency, diffusion models naturally require more computational overhead due to their multi-step sampling procedures. To provide a transparent comparison, we report the sampling steps T for each diffusion-based model in Table VIII. DiffMove and GenMove both perform $T = 50$ lightweight diffusion steps in the embedding space, leading to inference times of 300ms and 450ms, respectively. In contrast, TrajGDM executes $T = 300$ full diffusion steps directly in the trajectory space incurring a much higher inference cost (900ms). Similarly, BCDiff employs two coupled diffusion models with $T = 200$ steps for bidirectional trajectory generation, resulting in an inference time of approximately 3000ms. UniMob incurs an inference time of 750 ms even with $T = 50$ steps, owing to its larger Diffusion Transformer architecture for joint trajectory and flow modeling. The differences in sampling steps and inference cost primarily stem from the choice of diffusion space: models such as TrajGDM and BCDiff operate in the high-dimensional trajectory space and require more denoising steps to converge, whereas DiffMove performs diffusion in a lower-dimensional embedding space, achieving effective recovery with fewer steps. This embedding-space design is the key factor behind DiffMove's efficiency advantage among diffusion-based models.

TABLE VIII
ESTIMATED PARAMETER SIZE, TRAINING COST, AND INFERENCE RUNTIME OF REPRESENTATIVE TRAJECTORY MODELS ON AN NVIDIA RTX 2080 TI GPU.

Model	Params (M)	Train (s/epoch)	Conv. Epochs	Total Train (h)	Infer. (ms)	Samp. Steps
DiffMove	9	100	180	5.0	300	50
GenMove	10	120	200	6.7	450	50
TrajGDM	20	180	230	11.5	900	300
UniMob	15	150	210	8.8	750	50
BCDiff	30	250	290	20.1	3000	200
CLMove	4	45	100	1.3	120	-
ARMove	6	50	80	1.1	130	-
TrajBERT	10	80	120	2.7	210	-
STR	9	70	130	2.5	190	-
TERI	8	60	100	1.7	160	-
PeriodicMove	7	75	120	2.5	200	-
TrajMoE	40	160	150	6.7	280	-
BeTR	6	55	100	1.5	190	-
AttnMove	5	40	100	1.1	150	-

For non-diffusion baselines, CLMove, ARMove, and AttnMove achieve the fastest inference (120–150ms), while models incorporating graph structures or multi-level attention (e.g., PeriodicMove and STR) show slightly higher inference costs (around 190–200ms). Although DiffMove's inference time (300ms) is higher than these non-diffusion methods, trajectory recovery is typically an offline batch-processing task rather than a real-time application, making the current latency fully acceptable for practical deployment. Moreover, for scenarios with stricter latency requirements, DiffMove is compatible with accelerated sampling strategies such as DDIM [66], which can reduce the sampling steps from 50 to 10-25, potentially bringing the inference time down to 60-150ms with controllable performance degradation.

In summary, DiffMove offers a favorable balance between model capacity and computational cost: it maintains the advantages of diffusion-based modeling while achieving lower parameter size and faster inference than other diffusion-based approaches. These properties make DiffMove suitable for deployment in large-scale human mobility generation scenarios.

6) Sensitivity of Temporal and Spatial Granularity: To investigate the impact of discretization granularity, we evaluate DiffMove on the Tencent dataset under three temporal resolutions (15 min, 30 min, 60 min) and three spatial grid sizes (32×32, 64×64, 128×128).

Temporal granularity. As shown in Table IX, reducing the

TABLE IX
TEMPORAL GRANULARITY SENSITIVITY ON TENCENT DATASET.

Time Interval	Recall@1	Recall@5	MAP	Distance (m)	Grid Accuracy
15 min	0.8103	0.8874	0.8627	524	0.3285
30 min	0.8457	0.9159	0.8945	681	0.3694
60 min	0.8684	0.9312	0.9103	895	0.4012

TABLE X
SPATIAL GRANULARITY SENSITIVITY ON TENCENT DATASET.

Spatial Resolution	Recall@1	Recall@5	MAP	Distance (m)	Grid Accuracy
32×32	0.8793	0.9341	0.9187	862	0.4215
64×64	0.8457	0.9159	0.8945	681	0.3694
128×128	0.7986	0.8791	0.8524	513	0.3042

interval from 30 min to 15 min increases the number of daily time slots from 48 to 96, resulting in a higher missing ratio and greater recovery difficulty. Recall@1 decreases from 0.8457 to 0.8103, and MAP drops from 0.8945 to 0.8627. Nevertheless, Distance improves from 681 m to 524 m, since shorter intervals constrain user displacement within each slot, leading to more precise spatial predictions. When the interval is increased to 60 min, the reduced number of slots (24 per day) simplifies the classification task, yielding higher Recall@1 (0.8684) and MAP (0.9103). However, the coarser temporal resolution causes Distance to increase to 895 m, as each slot aggregates movement over a wider spatial range. Despite these variations, DiffMove maintains stable and competitive performance across all three settings.

Spatial granularity. As reported in Table X, a similar trade-off is observed along the spatial dimension. The coarse 32×32 grid yields only 1,024 candidate cells, leading to higher Recall@1 (0.8793) and Grid Accuracy (0.4215), but a larger Distance (862 m) due to the increased cell area. Conversely, the fine 128×128 grid produces 16,384 candidates, making classification harder and reducing Recall@1 to 0.7986, while achieving the lowest Distance of 513 m owing to the smaller cell coverage. The default 64×64 grid strikes a balanced trade-off between the two extremes.

Overall, the results confirm that DiffMove is robust to both temporal and spatial discretization choices. A consistent precision-coverage trade-off exists across granularities: finer resolution improves spatial localization at the cost of classification accuracy, and vice versa. The default configuration (30 min, 64×64) provides a well-balanced operating point and remains consistent with prior work for fair comparison [20], [21].

VI. CONCLUSION

We propose a novel mobility trajectory recovery method named DiffMove, which captures the group tendency and individual preference simultaneously. To capture the group mobility tendency from the group view, we construct a group tendency graph and fuse the group tendency into location representations with graph embedding. To model the individual mobility preference from the individual view, we first capture individual mobility preferences from their historical and current trajectories and then utilize a diffusion model to refine the spatio-temporal distribution of these trajectories. We conduct extensive experiments on three real-world datasets, and the results verify the effectiveness of our DiffMove model. In the future, we plan to extend our framework. Semantic-aware trajectory recovery can be realized by considering the functional properties of the visited locations.

ACKNOWLEDGMENT

This work is supported in part by the National Key Research and Development Program of China under Grant 2024YFC3307603, and in part by the Science and Technology Innovation Program of Xiongan New Area under Grant 2025XAGG0041.

REFERENCES

[1] D. Balcan and A. Vespignani, "Phase transitions in contagion processes mediated by recurrent mobility patterns," *Nature physics*, vol. 7, no. 7, pp. 581–586, 2011.
[2] C. Yang, Z. Zhang, Z. Fan, R. Jiang, Q. Chen, X. Song, and R. Shibasaki, "Epimob: Interactive visual analytics of citywide human mobility restrictions for epidemic control," *IEEE Transactions on Visualization and Computer Graphics*, 2022.
[3] H. Zhang, P. Li, Z. Zhang, W. Li, J. Chen, X. Song, R. Shibasaki, and J. Yan, "Epidemic versus economic performances of the covid-19 lockdown: A big data driven analysis," *Cities*, vol. 120, p. 103502, 2022.
[4] S. Gao, "Spatio-temporal analytics for exploring human mobility patterns and urban dynamics in the mobile age," *Spatial Cognition & Computation*, vol. 15, no. 2, pp. 86–114, 2015.
[5] K. S. Kung, K. Greco, S. Sobolevsky, and C. Ratti, "Exploring universal patterns in human home-work commuting from mobile phone data," *PloS one*, vol. 9, no. 6, p. e96180, 2014.
[6] L. Bengtsson, J. Gaudart, X. Lu, S. Moore, E. Wetter, K. Sallah, S. Rebaudet, and R. Piarroux, "Using mobile phone data to predict the spatial spread of cholera," *Scientific reports*, vol. 5, no. 1, p. 8923, 2015.
[7] Y. Li, C. Gao, Q. Yao, T. Li, D. Jin, and Y. Li, "Disenhcn: Disentangled hypergraph convolutional networks for spatiotemporal activity prediction," *arXiv preprint arXiv:2208.06794*, 2022.
[8] K. Liu, S. Gao, and F. Lu, "Identifying spatial interaction patterns of vehicle movements on urban road networks by topic modelling," *Computers, Environment and Urban Systems*, vol. 74, pp. 50–61, 2019.
[9] Y. Yue, T. Lan, A. G. Yeh, and Q.-Q. Li, "Zooming into individuals to understand the collective: A review of trajectory-based travel behaviour studies," *Travel Behaviour and Society*, vol. 1, no. 2, pp. 69–78, 2014.
[10] L. Li, Y. Li, and Z. Li, "Efficient missing data imputing for traffic flow by considering temporal and spatial dependence," *Transportation research part C: emerging technologies*, vol. 34, pp. 108–120, 2013.
[11] S. Hoteit, S. Secci, S. Sobolevsky, G. Pujolle, and C. Ratti, "Estimating real human trajectories through mobile phone data," in *2013 IEEE 14th International Conference on Mobile Data Management*, vol. 2. IEEE, 2013, pp. 148–153.
[12] 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.
[13] M. C. Gonzalez, C. A. Hidalgo, and A.-L. Barabasi, "Understanding individual human mobility patterns," *nature*, vol. 453, no. 7196, pp. 779–782, 2008.
[14] S. Isaacman, R. Becker, R. Cáceres, M. Martonosi, J. Rowland, A. Varshavsky, and W. Willinger, "Human mobility modeling at metropolitan scales," in *Proceedings of the 10th international conference on Mobile systems, applications, and services*, 2012, pp. 239–252.
[15] C. Song, Z. Qu, N. Blumm, and A.-L. Barabási, "Limits of predictability in human mobility," *Science*, vol. 327, no. 5968, pp. 1018–1021, 2010.
[16] Z. Xiao, H. Wen, A. Markham, and N. Trigoni, "Lightweight map matching for indoor localisation using conditional random fields," in *IPSN-14 proceedings of the 13th international symposium on information processing in sensor networks*. IEEE, 2014, pp. 131–142.

[17] M. Ficek and L. Kencl, "Inter-call mobility model: A spatio-temporal refinement of call data records using a gaussian mixture model," in *2012 Proceedings IEEE INFOCOM*. IEEE, 2012, pp. 469–477.

[18] H. Yu, A. Russell, J. Mulholland, and Z. Huang, "Using cell phone location to assess misclassification errors in air pollution exposure estimation," *Environmental pollution*, vol. 233, pp. 261–266, 2018.

[19] 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.

[20] 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.

[21] 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.

[22] C. Chu, H. Zhang, and F. Lu, "Trajgdm: A new trajectory foundation model for simulating human mobility," in *Proceedings of the 31st ACM International Conference on Advances in Geographic Information Systems*, 2023, pp. 1–2.

[23] 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.

[24] D. Xi, F. Zhuang, Y. Liu, J. Gu, H. Xiong, and Q. He, "Modelling of bi-directional spatio-temporal dependence and users' dynamic preferences for missing poi check-in identification," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 5458–5465.

[25] H. Ren, S. Ruan, Y. Li *et al.*, "Mtrajrec: Map-constrained trajectory recovery via seq2seq multi-task learning," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021, pp. 1410–1419.

[26] Y. Chen, H. Zhang, W. Sun *et al.*, "Rntrajrec: Road network enhanced trajectory recovery with spatial-temporal transformer," in *2023 IEEE 39th International Conference on Data Engineering (ICDE)*. IEEE, 2023, pp. 829–842.

[27] T. Wei, Y. Lin, Y. Lin, S. Guo, L. Zhang, and H. Wan, "Micro-macro spatial-temporal graph-based encoder-decoder for map-constrained trajectory recovery," *IEEE Transactions on Knowledge and Data Engineering*, vol. 36, no. 11, pp. 6574–6587, 2024.

[28] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in Neural Information Processing Systems*, vol. 33, pp. 6840–6851, 2020.

[29] J. G. Wijmans and R. W. Baker, "The solution-diffusion model: a review," *Journal of membrane science*, vol. 107, no. 1-2, pp. 1–21, 1995.

[30] J. Ho, T. Salimans, A. Gritsenko, W. Chan, M. Norouzi, and D. J. Fleet, "Video diffusion models," *arXiv preprint arXiv:2204.03458*, 2022.

[31] S. Gu, D. Chen, J. Bao, F. Wen, B. Zhang, D. Chen, L. Yuan, and B. Guo, "Vector quantized diffusion model for text-to-image synthesis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 10 696–10 706.

[32] N. Ruiz, Y. Li, V. Jampani, Y. Pritch, M. Rubinstein, and K. Aberman, "Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 22 500–22 510.

[33] B. Kawar, S. Zada, O. Lang, O. Tov, H. Chang, T. Dekel, I. Mosseri, and M. Irani, "Imagic: Text-based real image editing with diffusion models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 6007–6017.

[34] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 10 684–10 695.

[35] Z. Kong, W. Ping, J. Huang, K. Zhao, and B. Catanzaro, "Diffwave: A versatile diffusion model for audio synthesis," *arXiv preprint arXiv:2009.09761*, 2020.

[36] 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.

[37] J. M. L. Alcaraz and N. Strodthoff, "Diffusion-based time series imputation and forecasting with structured state space models," *arXiv preprint arXiv:2208.09399*, 2022.

[38] K. Rasul, C. Seward, I. Schuster, and R. Vollgraf, "Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting," in *International Conference on Machine Learning*. PMLR, 2021, pp. 8857–8868.

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

[40] Y. Zhang, Y. Yuan, J. Ding, J. Yuan, and Y. Li, "Noise matters: Diffusion model-based urban mobility generation with collaborative noise priors," in *Proceedings of the ACM on Web Conference 2025*, 2025, pp. 5352–5363.

[41] 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.

[42] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[43] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, "Hierarchical text-conditional image generation with clip latents," *arXiv preprint arXiv:2204.06125*, 2022.

[44] Y. Zheng, X. Xie, W.-Y. Ma *et al.*, "Geolife: A collaborative social networking service among user, location and trajectory." *IEEE Data Eng. Bull.*, vol. 33, no. 2, pp. 32–39, 2010.

[45] D. Yang, D. Zhang, V. W. Zheng, and Z. Yu, "A large-scale study of user check-ins on foursquare," in *Proceedings of the 8th International Conference on Weblogs and Social Media (ICWSM)*, 2014.

[46] R. Hariharan and K. Toyama, "Project lachesis: parsing and modeling location histories," in *International Conference on Geographic Information Science*. Springer, 2004, pp. 106–124.

[47] Y. Zheng, L. Zhang, X. Xie, and W.-Y. Ma, "Mining interesting locations and travel sequences from gps trajectories," in *Proceedings of the 18th international conference on World wide web*, 2009, pp. 791–800.

[48] G. Chen, A. C. Viana, M. Fiore, and C. Sarraute, "Complete trajectory reconstruction from sparse mobile phone data," *EPJ Data Science*, vol. 8, no. 1, pp. 1–24, 2019.

[49] 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.

[50] Q. Liu, S. Wu, L. Wang, and T. Tan, "Predicting the next location: A recurrent model with spatial and temporal contexts," in *Thirtieth AAAI conference on artificial intelligence*, 2016.

[51] J. Zhao, J. Xu, R. Zhou, P. Zhao, C. Liu, and F. Zhu, "On prediction of user destination by sub-trajectory understanding: A deep learning based approach," in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, 2018, pp. 1413–1422.

[52] Y. Chen, G. Cong, and C. Anda, "Teri: An effective framework for trajectory recovery with irregular time intervals," *Proceedings of the VLDB Endowment*, vol. 17, no. 3, pp. 414–426, 2024.

[53] J. Si, J. Yang, Y. Xiang, L. Chen, and J. Ma, "Trajbert: Bert-based trajectory recovery with spatial-temporal refinement for implicit sparse trajectories," *IEEE Transactions on Mobile Computing*, vol. 23, no. 5, pp. 4849–4860, 2023.

[54] W. Long, Z. Xiao, H. Jiang, Y. Xiong, Z. Qin, Y. Li, and S. Dustdar, "Learning semantic behavior for human mobility trajectory recovery," *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 8, pp. 8849–8864, 2024.

[55] H. Wang, S. Wang, L. Lin, Y. Yang, S. Wang, and H. Wen, "Behavior-aware sparse trajectory recovery in last-mile delivery with multi-scale attention fusion," in *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, 2024, pp. 4931–4938.

[56] C. Han, Y. Yuan, K. Chen, J. Ding, and Y. Li, "Trajmoe: Spatially-aware mixture of experts for unified human mobility modeling," *arXiv preprint arXiv:2505.18670*, 2025.

[57] Y. Zhao, C. Wang, H. Wang, S. Zhu, and L. Chen, "Towards high-quality spatiotemporal recommendation: Trajectory recovery based on spatial and temporal dependencies," *IEEE Transactions on Big Data*, 2025.

[58] Y. Liu, Y. Chen, J. Zhang, Y. Xiao, and X. Wang, "Toward an accurate mobility trajectory recovery using contrastive learning," *Frontiers of Information Technology & Electronic Engineering*, vol. 25, no. 11, pp. 1479–1496, 2024.

[59] Q. Long, C. Rong, H. Wang, and Y. Li, "One fits all: General mobility trajectory modeling via masked conditional diffusion," *arXiv preprint arXiv:2501.13347*, 2025.

[60] 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.

[61] Q. Long, Y. Yuan, and Y. Li, “A universal model for human mobility prediction,” in *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, 2025, pp. 894–905.

[62] R. Li, C. Li, D. Ren, G. Chen, Y. Yuan, and G. Wang, “Bcdiff: Bidirectional consistent diffusion for instantaneous trajectory prediction,” *Advances in neural information processing systems*, vol. 36, pp. 14 400–14 413, 2023.

[63] A. Grover and J. Leskovec, “node2vec: Scalable feature learning for networks,” in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 2016, pp. 855–864.

[64] Y. You, T. Chen, Y. Sui, T. Chen, Z. Wang, and Y. Shen, “Graph contrastive learning with augmentations,” *Advances in neural information processing systems*, vol. 33, pp. 5812–5823, 2020.

[65] Y. Zhu, Y. Xu, F. Yu, Q. Liu, S. Wu, and L. Wang, “Deep graph contrastive representation learning,” *arXiv preprint arXiv:2006.04131*, 2020.

[66] J. Song, C. Meng, and S. Ermon, “Denoising diffusion implicit models,” *arXiv preprint arXiv:2010.02502*, 2020.