Diff-DGMN: A Diffusion-based Dual Graph Multi-attention Network for POI Recommendation

Jiankai Zuo and Yaying Zhang, *Member, IEEE*

***Abstract*—Effective Point-of-Interest (POI) recommendation systems play a pivotal role in modern location-aware applications and human mobility, facilitating customized suggestions for users' upcoming exploration destinations. Understanding the intricate dynamics of user movement, which are often influenced by a multitude of factors, remains a formidable task. Moreover, discrepancies between the acquired representation distribution and the authentic target distribution of user interests also present a notable obstacle. To tackle these problems, we make an attempt to bridge the gap by introducing diffusion models and propose a Diffusion-based Dual Graph Multi-attention Network (Diff-DGMN). Specifically, we have constructed two types of graphs: one is a user-oriented local POI transition graph, and the other is a global-based POI distance graph. Subsequently, we put forward two graph learning representation modules to capture the sequential encoding of users and the geographic representations of nodes, respectively. Furthermore, an attention-based location prototype generation module is introduced to merge the captured sequential encoding and geographic representation, yielding richer semantic interaction features. In the end, we obtain the final results by leveraging the forward diffusion process and corresponding its reverse-time generation to sample users' future preferences from the posterior distribution. Our Diff-DGMN model demonstrates its remarkable recommendation performance through extensive experimentation on five real-world datasets. Compared with the most state-of-the-art methodologies, Diff-DGMN has improved performance in accuracy, normalized discounted cumulative gain (NDCG), and mean reciprocal rank (MRR) by 8.04%, 8.63%, and 9.09%, respectively. Our codes are available at https://github.com/JKZuo/Diff-DGMN.**

***Index Terms*—POI recommendation, Diffusion model, Graph neural network, Self-attention, Location-based social networks.**

I. Introduction

THE burgeoning growth of Location-based Social Networks (LBSNs) and location-aware applications, such as Yelp, Twitter, Facebook, and Foursquare, has attracted considerable interest in the advancement of next Point-of-Interest (POI) recommendation services [1]–[3]. Given the assumption that users' subsequent visits are strongly influenced by past check-in records, the majority of previous methodologies utilize sequential models to depict users' mobility patterns. Markov Chains [4] have been employed to understand the impact of preceding visits on subsequent user decisions. Recurrent Neural Networks (RNNs) and variants (e.g., LSTM and GRU) have been extensively utilized in prior studies [5]–[7], capitalizing on their ability to learn users' long-term behavior patterns. Additionally, many recent studies [8]–[12] have enhanced recommendation systems by integrating self-attention mechanisms with explicit spatio-temporal information. Compared to traditional sequential recommendation systems [13], [14], POI recommendations not only ought to capture user behavior characteristics from historical interactions, such as personalized preferences and habitual visitation patterns, but also consider the impact of geographical locations. It is crucial as POIs that are physically closer tend to be intuitively more appealing to users. Owing to their robust ability to process spatial structure data, models based on Graph Neural Networks (GNNs) have emerged to effectively capture high-order correlations among POIs as the predominant strategies in this domain [15]–[23].

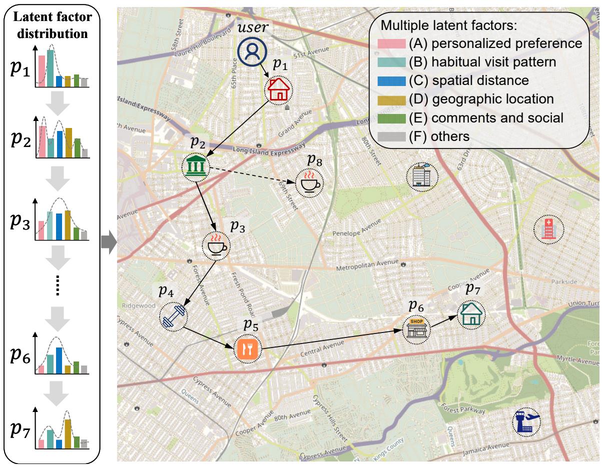


Fig. 1: Next POI visit is influenced by multiple latent factors and the dynamic interest distribution.

Although significant progress has been made in the aforementioned POI recommendation approaches, the majority of existing methods still face two key challenges [2], [10], [24]. **(i) Inability to effectively model the complexity and stochasticity of user mobility.** This challenge arises from the fact that the next location visited by a user is influenced by multiple latent factors [10], [21], [23], [25], including (a) personalized preference, (b) habitual visit pattern, (c) spatial distance, (d) geographical location, (e) comments and sociability, and (f) others. The personalized preference refers to the significant differences among users regarding the same POI, where the visit frequency can directly reflect this characteristic. For habitual visit patterns, a user might typically follow a visit to a library with a stop at a specific coffee shop. However, such a pattern can change depending on the geographical locations of POIs. As illustrated in Fig. 1, a user habitually visits the

This work was partly supported by the National Key Research and Development Program of China under Grant 2022YFB4501704 and the National Natural Science Foundation of China under Grant 72342026.

The authors are with the Key Laboratory of Embedded System and Service Computing, Ministry of Education, Tongji University, Shanghai 201804, China. (Corresponding author: yaying.zhang@tongji.edu.cn).

coffee shop (p_8) after the library (p_2), but due to a planned visit to the gym (p_4) with a friend, they check-in the coffee shop conveniently located p_3 instead. Moreover, after dining at p_5, the user chooses a store (p_6) close to his friend's home (p_7) in terms of a short spatial distance. Therefore, it is necessary to utilize these explicit features to model sophisticated user movement patterns.

Another challenge encountered is: **(ii) the disparity between the captured representation distribution and the actual target distribution of user interests.** This disparity stems from two primary causes: (a) the representations captured inherently contain noise; (b) there is a dynamic shift in the distribution of user interests. Since the interests and preferences of users evolve over time, and as relationships between POIs undergo updates, the ability of recommendation methods to naturally adapt to such dynamic variations is crucial for accurately predicting the user's next movement. Thanks to the intrinsic ability of the diffusion model [26]–[28] to generate distributions, it becomes feasible to model the multi-latent aspects of visit intentions and the dynamic user interests as probabilistic distributions. From the standpoint of the forward diffusion process, the overall transition between distributions is decomposed into a sequence of stochastic processes. This transition can be more effectively encapsulated using specialized model architectures, such as deep neural networks. Beyond restoring and denoising existing features, diffusion models are also capable of generating novel user interests, which may not be readily observable in historical trajectories. By exploring the latent space of data generation, diffusion methods can uncover hidden patterns of user preferences, thereby offering a more diverse and appealing set of novel POIs that meet the future preferences of users.

In this work, we present an effort to connect the diffusion model framework with the field of POI recommendation, introducing a novel approach termed Diff-DGMN. To finely depict personalized preferences and habitual visit patterns, we design a direction-aware sequence graph multi-scale representation module to gain a POI sequence encoding on the user-oriented POI transition graph. In addition, a global-based distance graph geographical representation module is introduced to model spatial relationships between POIs. Finally, a diffusion-based user preference sampling module is proposed to generate a transportation path from the captured representation distribution to the target distribution via a reverse-time generation process and a guiding strategy (i.e., a context-aware condition embedding) based on user historical visit patterns.

The main contributions of this paper are summarized as follows.

- We propose a dual-graph-driven representation module to well model the complexity and stochasticity of user mobility. Our approach leverages a direction-aware sequence graph to meticulously encode high-order sequential patterns derived from individual user preferences and habitual behaviors. Additionally, our global-based distance graph dynamically reflects the spatial relationships and geographical influences among different POIs.
- We introduce an innovative diffusion-based user preference sampling module aimed at generating a pure (noise-free) location archetype vector, which is capable of depicting the diffusion path from a source distribution to the target distribution and allowing for the exploration of novel user interests. It enhances the capability to make future POI recommendations by uncovering latent preferences and evolving user tastes.
- Extensive experiments are conducted on five real-world LBSN datasets. The results have validated that the proposed Diff-DGMN model has superior overall performance compared to state-of-the-art baseline methods.

The remainder of the paper is organized as follows. Section II investigates existing research on POI recommendations and diffusion models. Section III presents preliminary descriptions and the problem statement. Section IV introduces the Diff-DGMN model proposed in this paper. Section V reports experimental procedures and results analysis. Finally, Section VI provides the conclusion.

II. Related work

In this section, we will provide an overview of the existing representative studies from two perspectives: deep learning-based POI recommendations and diffusion models in POI recommendations.

A. Deep Learning-based POI Recommendations

In the realm of deep learning-based POI recommendations, methodologies can generally be categorized into three distinct types: 1) Recurrent Neural Network (RNN) based methods, 2) Attention-based models, and 3) Graph Neural Network (GNN) based approaches. Each category leverages unique architectures to tackle the specific challenges posed by POI recommendations.

RNN-based methods utilize LSTM or GRU to model the sequential behavior of users as they interact with different POIs. Liu et al. [5] put forward a spatio-temporal recurrent neural network (ST-RNN), integrating distinct transition matrices for different time and space dimensions. Zhao et al. [6] modified the traditional LSTM architecture by adding new gates, enhancing its ability to learn the dependency relationships between successive visits. Wu et al. [7] proposed a preference learning model called PLSPL, which took advantage of an attention layer and LSTM to capture users' long- and short-term preference patterns, respectively. By capturing temporal dependencies in user activity sequences, RNN-based methods can predict future interactions and recommend POIs that align with users' historical patterns. However, they fall short in learning more complex semantic interactions and gaining insights into novel user interests when traveling to unfamiliar areas.

Attention-based models focus on significant parts of the input sequences, allowing the model to prioritize more relevant information [8] when making recommendations. This is particularly useful in handling contexts where specific interactions or temporal features hold more predictive power for future POI visits. STAN [9] addressed both non-adjacent and non-successive visits by incorporating self-attention with a spatiotemporal correlation matrix. BayMAN [10] examined robust

POI recommendations by employing a Bayesian-enhanced graph and introduced multi-perspective attention mechanisms that consider time, distance, and semantic aspects. DRAN [11] presented a disentangled self-attention aggregation module to capture dynamic user preferences in subsequent time slots. PG2Net [12] combined Bi-LSTM and attention to learn the user's mobility tendency from the perspectives of individual and collective signals.

GNN-based approaches are employed to exploit the relationships between users and POIs, encoded as graphs. GNNs are adept at handling complex structures [15], enabling them to infer the underlying patterns of connectivity in user-POI interactions. Ju et al. [16] developed a kernel-based graph neural network (KBGNN) that integrates geographical and sequential data derived from two distinct graphs. Qin et al. [17] introduced a disentangled dual-graph model, DisenPOI, which leveraged both sequential and geographical interactions and disentangled their influences with a self-supervision technique. Xu et al. [18] designed a graph Transformer framework to integrate distinct spatial and temporal graph encoders, allowing it to extract unique properties and understand the comprehensive user-POI interactions effectively. Furthermore, many methods (e.g., HKGNN [19], EEDN [20], SLS-REC [22], and STHGCN [23]) employed hypergraphs to enhance recommendations, owing to their capability to capture high-order relationships, while preserving a natural coherence among the nodes contained in hyperedges. Especially, STHGCN [23] came up with a hypergraph transformer designed to extract trajectory-level features from both intra-user and inter-user collaborative trajectories.

However, the aforementioned methods have not adequately considered the inconsistency between learned representation distributions and target distributions of dynamic user interests in real-life recommendation scenarios. While some approaches may prioritize accuracy in predicting user preferences based on historical visits, they frequently overlook the dynamics and context-specific factors that influence user behavior. Different from previous works, we integrate high-order sequential features into user spatial preferences, thus paving the way for more accurate and context-aware recommendations. In addition, we also propose a novel paradigm that introduces diffusion models into user preference sampling that can effectively bridge the gap between captured user preferences and actual user interests.

B. Diffusion Model and Its Applications

A diffusion model [26] is a type of deep generative method that operates in two stages: forward diffusion and reverse generation. Inspired by non-equilibrium thermodynamics [29], denoising diffusion probabilistic models (DDPMs) [28] are employed in image generation. Additionally, by applying Stochastic Differential Equations (SDEs) to model the diffusion process [30], the reverse evolution can be interpreted as a Langevin sampling from a specified prior distribution. Within the realm of generative techniques, the advantage of diffusion processes enables models to more ideally represent complicated sample space distributions, compared to classical approaches like Variational Auto-Encoders (VAEs) [31] and Generative Adversarial Networks (GANs) [32].

In recent years, diffusion models have achieved significant success across various domains, including computer vision [28], spatiotemporal mining [33], [34], and recommendation systems [14], [35]–[39]. For example, Lin et al. [34] developed a spectral diffusion model (SpecSTG) to solve spatio-temporal graph learning. Ma et al. [37] proposed a plug-in diffusion (PDRec) for the sequential recommendation. In addition, Qin et al. [38] proposed a diffusion model for POI recommendation, which sampled from the posterior distribution of user embeddings and intended to focus on user geographical preferences. Long et al. [39] introduced a diffusion-based cloud-edge-device collaborative learning for POI recommendation (DCPR) that was mainly designed at the application level to ensure user privacy and timely on-device POI recommendation. The global diffusion of DCPR aimed to construct the next category embedding from Gaussian noise based on historical category sequences.

Different from the aforementioned methods, we propose a novel diffusion-based preference generator that can depict the diffusion pathway from a source distribution to a target distribution, enabling the exploration of time-evolving user interests. Moreover, we design various noise schedules for POI recommendation and discuss their impact on diffusion processes. While diffusion models have become widespread in diverse fields, further exploration to bridge the gap between the next POI recommendation and diffusion models is still needed. One of the key strengths of diffusion models is their ability to explore and generate new data points that are not explicitly present in the training set but are reasonable based on the learned distribution. In the context of POI recommendations, it means the model is capable of suggesting novel locations that users have not visited but might find appealing based on their diffusion-modulated preferences.

III. Preliminaries

A. Next POI Recommendation

Let $\mathcal{U} = \{u_1, u_2, \dots, u_U\}$ indicate a set of users and $\mathcal{P} = \{p_1, p_2, \dots, p_P\}$ represent a set of POIs. Each POI $p = (lon, lat, cat) \in \mathcal{P}$ refers to a specific geographical location composed of longitude, latitude, and category. In addition, $\mathcal{T} = \{t_1, t_2, \dots, t_T\}$ means a set of timestamps.

Definition 1 (Check-in Trajectory Sequence). For a user $u \in \mathcal{U}$, the corresponding trajectory sequence is denoted as $\mathcal{T_S}(u) = \{(p_i^u, t_i^u) | i = 1, 2, \dots, n\}$ in which each check-in tuple (p_i^u, t_i^u) represents the user u visited (checked-in) a POI p_i^u at timestamp t_i^u.

Definition 2 (User-oriented POI Transition Graph). For all historical trajectories of a user $u \in \mathcal{U}$, we can construct a directed local POI transition graph $\mathcal{G}_u = \langle \mathcal{V}_u, \mathcal{E}_u, \mathcal{A}_u \rangle$, where $\mathcal{V}_u$ contains all POIs that u checked-in, and each edge $e_{i,j}^u = \langle p_i^u, p_j^u \rangle \in \mathcal{E}_u$ means the user u checked-in POI p_j^u after p_i^u. The edge weight $a_{i,j}^u \in \mathcal{A}_u$ measures the probability of a successive visit from p_i^u to p_j^u.

Definition 3 (Global-based POI Distance Graph). To model spatial dependencies for all POIs within a city, we

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2024.3446048

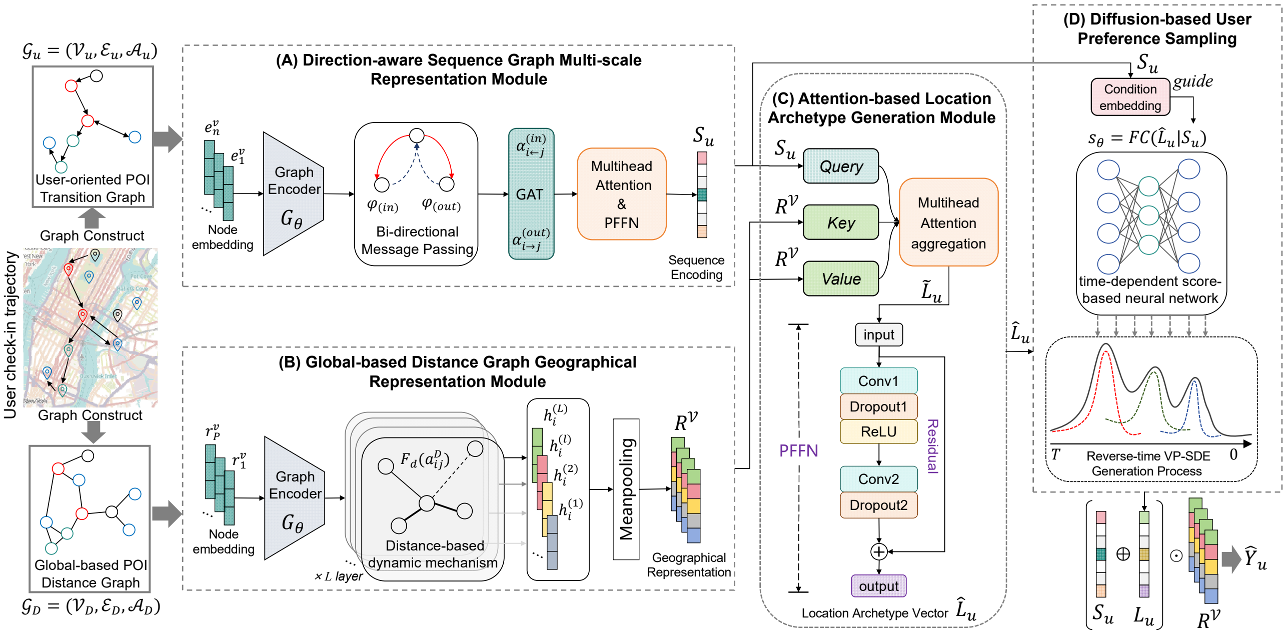

$\mathcal{G}_u = (\mathcal{V}_u, \mathcal{E}_u, \mathcal{A}_u)$
User-oriented POI Transition Graph
Graph Construct
User check-in trajectory
Graph Construct
Global-based POI Distance Graph
$\mathcal{G}_D = (\mathcal{V}_D, \mathcal{E}_D, \mathcal{A}_D)$
(A) Direction-aware Sequence Graph Multi-scale Representation Module
e_n^v
e_1^v
Node embedding
Graph Encoder G_θ
$\varphi_{(in)}$
$\varphi_{(out)}$
Bi-directional Message Passing
$\alpha_{i\leftarrow j}^{(in)}$
GAT
$\alpha_{i\rightarrow j}^{(out)}$
Multihead Attention & PFFN
S_u
Sequence Encoding
(B) Global-based Distance Graph Geographical Representation Module
r_P^v
r_1^v
Node embedding
Graph Encoder G_θ
$F_d(a_{ij}^D)$
Distance-based dynamic mechanism
× L layer
$h_i^{(L)}$
$h_i^{(l)}$
$h_i^{(2)}$
$h_i^{(1)}$
Meanpooling
$R^\mathcal{V}$
Geographical Representation
(C) Attention-based Location Archetype Generation Module
S_u
Query
$R^\mathcal{V}$
Key
$R^\mathcal{V}$
Value
Multihead Attention aggregation
$\tilde{L}_u$
input
Conv1
Dropout1
ReLU
Conv2
Dropout2
Residual
PFFN
output
Location Archetype Vector $\hat{L}_u$
$\hat{L}_u$
(D) Diffusion-based User Preference Sampling
S_u
Condition embedding
guide
$s_\theta = FC(\hat{L}_u|S_u)$
time-dependent score-based neural network
T
0
Reverse-time VP-SDE Generation Process
S_u
L_u
$R^\mathcal{V}$
$\hat{Y}_u$

Fig. 2: The overall framework of the proposed Diffusion-based Dual Graph Multi-attention Network (Diff-DGMN) model.

can construct an undirected POI distance graph $\mathcal{G}_D = (\mathcal{V}_D, \mathcal{E}_D, \mathcal{A}_D)$, where $\mathcal{V}_D = \mathcal{P}$ contains all POIs of the entire city, and each edge $e_{i,j}^D = (v_i, v_j) \in \mathcal{E}_D$ reveals the spatial distance between any two pairs of POIs (locations). The edge weight $a_{i,j}^D \in \mathcal{A}_D$ depicts the geographical closeness between location p_i and p_j.

Definition 4 (Next POI Recommendation). Given the sets of users $\mathcal{U}$ and POIs $\mathcal{P}$, a set of historical check-in trajectory sequences $\mathcal{T}_\mathcal{S}^\mathcal{H} = \{\mathcal{T}_\mathcal{S}(u)|u = 1, 2, \ldots, U\}$ of all users, where $\mathcal{T}_\mathcal{S}(u) = \{(p_1^u, t_1^u), (p_2^u, t_2^u), \ldots, (p_n^u, t_n^u)\}$, the problem definition is to generate a ranked POI list from all candidate locations $\mathcal{P}$. We recommend the Top-k POIs to the user u who is most likely to visit at the upcoming activity location.

B. Stochastic Differential Equation

The stochastic differential equation (SDE) of the forward diffusion process can be formulated as Eq. (1), commonly known as the Itô form [30], and it describes the evolution of a stochastic process $y(t)$, $t \in [0, T]$.

$$\mathrm{d}y(t) = \underbrace{f(y(t), t)}_{\text{drift term}} \mathrm{d}t + \underbrace{g(y(t), t)}_{\text{diffusion term}} \mathrm{d}\omega(t), \tag{1}$$

where this preceding term $f(y(t), t)$ is the drift term, representing the deterministic component of the process. It determines the average or expected rate of change of $y(t)$ at each time point t. The latter $g(y(t), t)$ is the diffusion term, representing the random or stochastic component of the process. $\mathrm{d}t$ is the differential of time and $\mathrm{d}\omega(t)$ denotes the differential of the Wiener process (or Brownian motion).

The corresponding reverse-time SDE [40] can be formulated as Eq. (2).

$$\mathrm{d}y(t) = [f(y, t) - g(y, t)^2 \nabla_{y(t)} \log p_t(y)]\mathrm{d}t + g(y, t)\mathrm{d}\hat{\omega}(t), \tag{2}$$

where $\mathrm{d}\hat{\omega}(t)$ expresses the differential of Brownian motion when the time point t is reversed, $t \in [T, 0]$, and $\nabla_{y(t)} \log p_t(y)$ is the gradient of the log marginal probability.

The forward SDE converts the data into noise, while the reverse-time SDE (also known as the generation process) learns to convert noise back into data. When solving the inverse process, a score neural network $s_\theta(y, t) \approx \nabla_{y(t)} \log p_t(y)$ can be trained to estimate the gradient of the log marginal probability.

IV. Methodology

The overall framework of our proposed Diff-DGMN model is illustrated in Fig. 2. Diff-DGMN comprises primarily two graph learning modules that obtain the sequence encoding of users S_u and geographical representation of locations $R^\mathcal{V}$, respectively. Later, we propose an attention-based generation module to merge the captured two representations S_u and $R^\mathcal{V}$, achieving a location archetype vector $\hat{L}_u$ that encapsulates richer semantic interactions. Ultimately, a diffusion-based sampling module is conducted to generate recommendations tailored to future user preferences. Notations used in this paper are listed in Table I.

A. User-oriented POI Transition Graph Learning

In order to capture individual user visit preferences and behavior patterns, and reflect the local regularity of user transitions between different POIs, we propose a direction-aware sequence graph multi-scale representation module to gain the POI sequence encoding S_u on the user-oriented POI transition graph $\mathcal{G}_u$. The detailed process is depicted in Fig. 3. Specifically, $E^\mathcal{V} = \{e_1^v, e_2^v, \ldots, e_n^v\}$ is the input node embedding and the output node feature $H^G = \{h_1^v, h_2^v, \ldots, h_n^v\}$ is aggregated by the graph convolutional network (GCN), as Eq. (3).

$$H^G = G_\theta(E^\mathcal{V}, \mathcal{A}_u) = \sigma\left(\tilde{D}^{-\frac{1}{2}} \tilde{\mathcal{A}}_u \tilde{D}^{-\frac{1}{2}} E^\mathcal{V} W_g\right), \tag{3}$$

where $G_\theta(\cdot, \cdot)$ signifies graph convolution operations, $\tilde{\mathcal{A}}_u = \mathcal{A}_u + I_N$ is the sum of the adjacency matrix $\mathcal{A}_u$ of graph

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2024.3446048

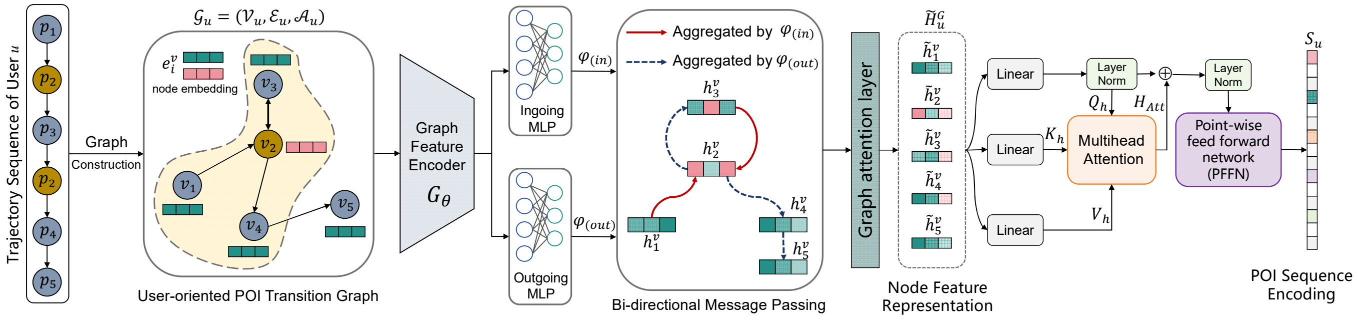


Fig. 3: The direction-aware sequence graph multi-scale representation module.

TABLE I: Detailed notations used in our paper.

Notations	Descriptions
$\mathcal{U}, \mathcal{P}, \mathcal{T}$	user set, POI set, timestamp set
u, p, t	user $u \in \mathcal{U}$, POI $p \in \mathcal{P}$, timestamp $t \in \mathcal{T}$
$\mathcal{T}_{\mathcal{S}}(u)$	check-in trajectory sequence of user u
$\mathcal{G}_u, \mathcal{G}_D$	POI transition graph, POI distance graph
$\mathcal{V}_u, \mathcal{E}_u, \mathcal{A}_u$	vertex set, edge set, adjacency matrix of $\mathcal{G}_u$
$\mathcal{V}_D, \mathcal{E}_D, \mathcal{A}_D$	vertex set, edge set, adjacency matrix of $\mathcal{G}_D$
$e^u_{i,j}, e^D_{i,j}$	edge weight of $\mathcal{G}_u$ and $\mathcal{G}_D$
$e^{\Delta t}_{ij}, e^{\Delta d}_{ij}$	time and distance interval embeddings
$S(v_i), h^v_i$	neighbor set of node v_i, feature vector of node v_i
$e^p_i, E^{\mathcal{P}}$	POI embedding of node v_i, POI embedding set
$e^v_i, E^{\mathcal{V}}$	input embedding of node v_i, input embedding set
h^v_i, H^G	output feature of node v_i, output node feature set
$\tilde{h}^v_i, \tilde{H}^G$	representation of node v_i, node representation set
$\alpha^{(in)}_{i\leftarrow j}, \alpha^{(out)}_{i\rightarrow j}$	ingoing and outgoing attention weights
Q_h, K_h, V_k	query matrix, key matrix, value matrix
$\tilde{\alpha}_{ij}, H_{Att}$	updated attention weight, multi-scale attention feature
$r^v_i, R^{\mathcal{V}}$	geographic feature, geographic representation set
e^r_i, e^c_i	region embedding and category embedding of POI
$m_{i\leftarrow j}$	message transmission feature from node v_j to v_i
S_u, L_u	user sequence encoding, location archetype vector
$\mathrm{d}t, T$	differential of time, diffusion size
$\beta(t), \mathrm{d}\omega(t)$	noise schedule function, Brownian motion
W_*, b_*, m_*	trainable weight, bias, message matrices
d, n_h, L	embedding size, number of heads, number of layers
$\hat{y}^u_i, \hat{Y}_u$	probability of u visiting POI i, ranked probability list
λ, ψ, ζ	balance coefficient, regularization factor, weight factor
$\mathcal{L}_{div}, \mathcal{L}_{rec}, \mathcal{L}_{tot}$	Fisher divergence, recommendation loss, total loss

$\mathcal{G}_u$ and an identity matrix I_N, which is done to add the self-loop of nodes and consider their own features during feature propagation. $\tilde{D}$ (diagonal matrix) represents the degree matrix of $\tilde{\mathcal{A}}_u$ and W_g denotes a trainable weight matrix. $\sigma(\cdot)$ means a nonlinear activation function.

In particular, for any node v_i on the user-oriented POI transition graph $\mathcal{G}_u$, we have Eq. (4) to obtain the corresponding node feature $h^v_i \in \mathbb{R}^d$.

$$h^v_i = G_\theta(e^v_i, a^u_{i,j}|\mathcal{G}_u), \tag{4}$$

Traditional graph convolutional layers, in the process of aggregating features from neighboring nodes, do not account for the influence of the distance and time intervals between consecutive check-in records within the input trajectory sequences. However, it is crucial for modeling the behavioral preferences of users. The rationale is that for two check-in records with the same visited POIs, $r_1 : p_1 \xrightarrow{\Delta t} p_2$ and $r_2 : p_1 \xrightarrow{\Delta t'} p_2$, one occurring within a short duration and the other separated by an extended long time interval (i.e., $\Delta t < \Delta t'$), the correlation in the former is typically stronger than in the latter due to the diminishing temporal relevance over prolonged periods [41]. Therefore, the time and distance intervals between check-ins should not be neglected when learning node feature representations.

Considering the above, we combine time $e^{\Delta t}_{ij}$ and distance $e^{\Delta d}_{ij}$ interval embeddings into the corresponding POI embedding to form the input node embedding $e^v_i \in \mathbb{R}^d$, as shown in Eq. (5). Besides, for a given set of POIs $\mathcal{P}$, we can learn their embeddings that satisfy a normal distribution with a mean of 0 and a variance of σ^2, that is $E^{\mathcal{P}} = \{e^p_1, e^p_2, \ldots, e^p_{|\mathcal{P}|}\} \sim \mathcal{N}(0, \sigma^2)$, where each element $e^p_i \in \mathbb{R}^d$, $\sigma^2 = 1/d$, and d represents the embedding dimension.

$$e^v_i = e^p_i + \sum\nolimits_{j\in S(v_i)} e^{\Delta t}_{ij} + \sum\nolimits_{j\in S(v_i)} e^{\Delta d}_{ij}, \tag{5}$$

where $S(v_i)$ signifies the set of neighbors of the node v_i.

Considering that standard 0/1 adjacency matrices or numerical adjacency matrices are difficult to measure the transition probability of user check-in patterns, we construct a transfer-wise adjacency matrix, which is calculated as Eq. (6).

$$a^u_{i,j} = \frac{freqcount\,(v_i \rightarrow v_j)}{Count\,(v_i)} \in \mathcal{A}_u, \tag{6}$$

where $a^u_{i,j} \in \mathcal{A}_u$ is the edge weight of transition graph $\mathcal{G}_u$, $freqcount\,(v_i \rightarrow v_j)$ represents the number of the POI transition from v_i to v_j, and $Count\,(v_i)$ is the total number of visits on v_i for user u.

As bi-directional information is of equal significance in scenarios of POI recommendation [42], we expect the message-passing for node feature propagation should encapsulate attributes from antecedent visits as well as prospective check-ins. For this purpose, an attention-based message-passing function is proposed to exchange messages from in-going and out-going edges, thereby aggregating the neighbor node features in a bi-directional manner, as shown in Eq. (7).

$$\begin{cases} \alpha^{(in)}_{i\leftarrow j} = h^v_j \times \varphi(in) \\ \alpha^{(out)}_{i\rightarrow j} = h^v_i \times \varphi(out) \\ \alpha^{(total)}_{ij} = Concat[\alpha^{(in)}_{i\leftarrow j}; \alpha^{(out)}_{i\rightarrow j}], \end{cases} \tag{7}$$

where $\alpha^{(in)}_{i\leftarrow j}$ and $\alpha^{(out)}_{i\rightarrow j}$ represent the attention weight between the target node i and its neighboring node j from the in-going and out-going directions, respectively. $\varphi(in)$ and $\varphi(out)$ are two weighted coefficient matrices learned from in- and out-going multi-layer perceptrons (MLPs).

To ensure the stability of feature propagation processes, we then perform a $softmax$ operation on the attention weights of each neighbor node to get an updated attention weight $\tilde{\alpha}_{ij}$, as Eq. (8). The updated $\tilde{\alpha}_{ij}$ can assign distinct importance weights to each node, thereby accentuating the POIs that users are more likely to visit.

$$\tilde{\alpha}_{ij} = \frac{exp\left(\alpha_{ij}^{(total)}\right)}{\sum_{k \in S(v_i)} exp\left(\alpha_{ik}^{(total)}\right)}, \tag{8}$$

Then, we employ the updated attention weight and neighbor node features h_j^v to receive an updated node feature representation $\tilde{h}_i^v \in \mathbb{R}^d$ of node v_i, as Eq. (9). In the end, by aggregating the features of each node based on structural information of the entire graph, we can output the whole node representation $\tilde{H}_u^G = \{\tilde{h}_1^v, \tilde{h}_2^v, \dots, \tilde{h}_n^v\} \in \mathbb{R}^{n \times d}$ of the user-oriented POI transition graph $\mathcal{G}_u$.

$$\tilde{h}_i^v = LeakyReLU\left(\sum\nolimits_{j \in S(v_i)} \tilde{\alpha}_{ij} \cdot W_h \cdot h_j^v\right), \tag{9}$$

where W_h is the trainable parameter and $LeakyReLU(\cdot)$ denotes the activation function whose negative slope is 0.01.

Up to this point, the captured representation within matrix $\tilde{H}_u^G \in \mathbb{R}^{n \times d}$ provides a structured spatial relationship between different nodes while strengthening the attention to pivotal POIs (nodes) in a bi-directional manner. To effectively achieve interactions among POIs within diverse representation subspaces, we employ a multihead attention mechanism to learn the correlations from multiple aspects. Given the input $\tilde{H}_u^G$, we first use linear mappings and a layer normalization to transform it into query matrix $Q_h = LayerNorm(\tilde{H}_u^G W_h^Q)$, key matrix $K_h = \tilde{H}_u^G W_h^K$, and value matrix $V_h = \tilde{H}_u^G W_h^V$, where three mapping matrices W_h^Q, W_h^K, $W_h^K \in \mathbb{R}^{d \times d}$. Next, the feature information is interacted by each head_i with scaled dot-product multihead attention [43] from diverse representation sub-spaces, as Eq. (10). Later, combining each head_i output as $H_{Att} \in \mathbb{R}^{n \times d}$ that captures multi-scale feature interactions from $\tilde{H}_u^G$ matrix.

$$\begin{cases} \mathrm{head}_i = \mathrm{Attention}(Q_h W_q^i, K_h W_k^i, V_h W_v^i) \\ \mathrm{Attention}(Q, K, V) = Softmax\left(\frac{QK^\top}{\sqrt{d/n_h}}\right) V \\ H_{Att} = \mathrm{MultiAtt}(Q, K, V) = \left(\overset{n_h}{\underset{i=1}{\Vert}} \mathrm{head}_i\right) W_O, \end{cases} \tag{10}$$

where W_q^i, W_k^i, $W_v^i \in \mathbb{R}^{d \times d/n_h}$ are transformation matrices of Q_h, K_h, and V_h corresponding to head_i, respectively. n_h is the number of heads, $\Vert$ signifies concatenation operation, and $W_O \in \mathbb{R}^{d \times d}$ is the output weight matrix.

In an effort to allow networks to learn the identity mapping [44], we implement an operation similar to residual connections on H_{Att}, that is, $H'_{Att} = LayerNorm(H_{Att} + Q_h) \in \mathbb{R}^{n \times d}$ to facilitate the training of deeper networks. Finally, a point-wise feed-forward network (PFFN), comprising two layers of convolutions, dropouts and nonlinear activation, as well as a residual connection, is leveraged to achieve the POI sequence encoding $S_u = f_{\text{PFFN}}(H'_{Att}) \in \mathbb{R}^d$.

B. Global-based POI Distance Graph Learning

Unlike local POI transition graph $\mathcal{G}_u$ that focuses on individual user behavior, the global POI distance graph $\mathcal{G}_D$ is constructed based on the geographic distance of POIs, reflecting the spatial relationships and geographical closeness between locations. In particular, for any node v_i on the global POI distance graph $\mathcal{G}_D$, we have Eq. (11) to obtain its node geographical representation $r_i^v \in \mathbb{R}^d$.

$$r_i^v = G_\theta(e_i^v, a_{i,j}^D | \mathcal{G}_D), \tag{11}$$

where e_i^v is the input node feature vector and $a_{i,j}^D \in \mathcal{A}_D$ is the weight of edge $e_{i,j}^D = \langle v_i, v_j \rangle \in \mathcal{E}_D$.

The node feature vector of v_i is calculated by Eq. (12).

$$e_i^v = e_i^p + e_i^r + e_i^c, \tag{12}$$

where e_i^p represents the POI embedding, e_i^r is the corresponding region embedding associated with its geographical location, and e_i^c is the category embedding.

The initial weights on each edge are determined by Eq. (13). We leverage an exponential decay to simulate the law of decreasing geographic correlation as distance increases.

$$a_{i,j}^D = \begin{cases} exp(-d(v_i, v_j)^2) & , \quad 0 < d(v_i, v_j) < \delta_d \\ 0 & , \quad otherwise \end{cases} \tag{13}$$

where $d(v_i, v_j)$ indicates the Haversine distance in kilometers between v_i and v_j calculated based on their latitude and longitude coordinates. The hyper-parameter δ_d denotes a distance threshold.

In order to capture the interaction information of spatial dependencies between different locations, we stack multi-layer graph convolution networks to expand the receptive field. Particularly, the message between target node v_i and its nearby node v_j on l^{th} layer is propagated as Eq. (14). Because closer nodes often exhibit more similarities than distant nodes, a dynamic mechanism $F_d(a_{i,j}^D)$ is introduced to adjust the weight of message transmission based on distance features. It is specifically implemented by a neural network composed of multiple fully connected layers, which learns how to adjust dynamic weights according to the initial weight $a_{i,j}^D$ on edge.

$$m_{i \leftarrow j}^{(l)} = \frac{F_d(a_{i,j}^D) \cdot W_m^{(l)} \cdot h_j^{(l-1)}}{\sqrt{|\mathcal{V}_i| \times |\mathcal{V}_j|}}, \tag{14}$$

where $W_m^{(l)}$ is a trainable weight matrix and $h_j^{(l-1)}$ is the feature vector of node v_j on $(l-1)^{th}$ layer, initial $h_j^{(0)} = e_j^v$. $|\mathcal{V}_i|$ and $|\mathcal{V}_j|$ represents the amount of neighbors for v_i and v_j, respectively.

Furthermore, the updated node information $h_i^{(l)}$ on l^{th} layer is aggregated by its own feature and neighboring features, as Eq. (15).

$$h_i^{(l)} = LeakyReLU\left(m_{i \leftarrow i}^{(l)} + \sum\nolimits_{j \in S(v_i)} m_{i \leftarrow j}^{(l)}\right), \tag{15}$$

where $m_{i \leftarrow i}^{(l)} = W_m^{(l)} h_i^{(l-1)}$ is the message aggregated by the node itself and $S(v_i)$ signifies a set of neighbors of node v_i.

We can obtain the final node representation r_i^v of each node by applying a mean pooling across all layers, as

Eq. (16). The whole node geographical representation $R^{\mathcal{V}} = \{r_1^v, r_2^v, \ldots, r_{|\mathcal{P}|}^v\} \in \mathbb{R}^{|\mathcal{P}| \times d}$ of all POIs within a city.

$$r_i^v = Meanpooling(h_i^{(0)}, h_i^{(1)}, \ldots, h_i^{(L)}), \tag{16}$$

C. Attention-based Archetype of Candidate Location

At present, we have obtained the POI embedding $E^{\mathcal{P}} = \{e_1^p, e_2^p, \ldots, e_{|\mathcal{P}|}^p\}$, sequence encoding S_u captured from a user-oriented POI transition graph $\mathcal{G}_u$, and geographic representation $R^{\mathcal{V}} = \{r_1^v, r_2^v, \ldots, r_{|\mathcal{P}|}^v\}$ of all POIs from the global POI distance graph $\mathcal{G}_D$. We aim to find an approach to calculate the probability $\hat{y}_{p_i}^u$ of user u visiting at the location (POI) p_i based on the features already obtained, which can be formalized as Eq. (17). Furthermore, we parameterize this probability distribution with a location archetype vector $L_u \in \mathbb{R}^d$, as Eq. (18).

$$\hat{y}_{p_i}^u = p(p_i|\mathcal{G}_u, \mathcal{G}_D) = p(p_i|S_u, R^{\mathcal{V}}), \tag{17}$$

$$p(p_i|S_u, R^{\mathcal{V}}) = p(p_i|L_u) \times p(L_u|S_u, R^{\mathcal{V}}), \tag{18}$$

where $L_u \in \mathbb{R}^d$ represents a location archetype vector for user u. Among them, the former can be defined as Eq. (19).

$$p(p_i|L_u) = CosineSim(r_i^v, L_u), \tag{19}$$

where $CosineSim(\cdot, \cdot)$ signifies the cosine similarity function.

Therefore, it is necessary for us to sample L_u from the posterior distribution $p(L_u|S_u, R^{\mathcal{V}})$ to receive the result $\hat{y}_{p_i}^u$. A simple solution is to fuse two features S_u and $R^{\mathcal{V}}$ and feed them into a fully connected layer, but it will lead to the learned representation of the proposed model being coarse-grained. To extract richer semantic interactions, we design an Attention-based Location Archetype Generation module. Firstly, we treated S_u as a query matrix and $R_u^{\mathcal{V}}$ as both a key matrix and a value matrix, performing a multi-head attention aggregation operation, which are formulated as Eq. (20), Eq. (21) and Eq. (22).

$$\alpha_i^h = \frac{(W_q^h \cdot S_u)(W_k^h \cdot R_u^{\mathcal{V}})^\top}{\sqrt{d/n_h}}, \tag{20}$$

$$\tilde{\alpha}_i^h = \frac{exp\left(\alpha_i^h\right)}{\sum\limits_{j=1}^{n} exp\left(\alpha_j^h\right)}, \tag{21}$$

$$\tilde{L}_u = \mathop{\Big\|}_{h=1}^{n_h} \left(\sum_{i=1}^{n} \tilde{\alpha}_i^h \cdot (W_v^h \cdot R_u^{\mathcal{V}}) \right) W_O, \tag{22}$$

where W_q^h, W_k^h and W_v^h indicate the query, key and value projection matrices, individually. $R_u^{\mathcal{V}} = \{r_1^v, r_2^v, \ldots, r_n^v\} \in \mathbb{R}^{n \times d}$ is a subset of $R^{\mathcal{V}} \in \mathbb{R}^{|\mathcal{P}| \times d}$, extracted from the user's check-in sequence trajectory $\mathcal{T}_{\mathcal{S}}(u) = \{(p_1^u, t_1^u), (p_2^u, t_2^u), \ldots, (p_n^u, t_n^u)\}$. $\tilde{\alpha}_i^h$ is interpreted as the attention score of the i^{th} check-in. n_h means the number of heads, $\|$ signifies concatenation operation, and W_O denotes an output weight matrix.

Then, a point-wise feed-forward network (PFFN) is employed to output the location archetype vector $\hat{L}_u$, as Eq. (23).

$$\hat{L}_u = (\tilde{L}_u + (ReLU(\tilde{L}_u W_p^1 + b_p^1))W_p^2 + b_p^2), \tag{23}$$

where $\{W_p^1, b_p^1, W_p^2, b_p^2\}$ are all learnable weight coefficients.

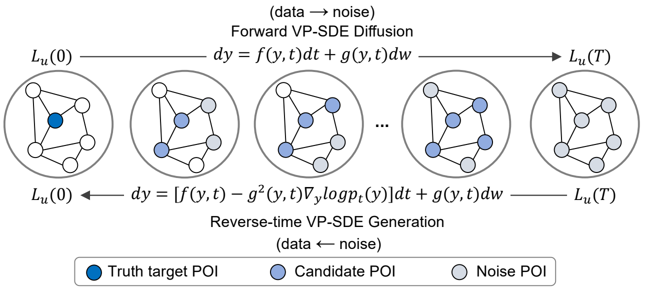


Fig. 4: Illustration of forward diffusion and reverse generation.

D. Diffusion-based User Preference Sampling

Although $\hat{L}_u \in \mathbb{R}^d$ reflects the user-specific personalized preference captured from local and global perspectives, there is still some noise involved. Inspired by the achievements in diffusion models, we propose a Diffusion-based User Preference Sampling module to generate a pure (noise-free) location archetype vector from noisy $\hat{L}_u$, leveraging the variance-preserving stochastic differential equation (VP-SDE). Specifically, this module is divided into two steps: 1) Forward VP-SDE Diffusion Process; 2) Reverse-time VP-SDE Generation Process. The forward diffusion process aims to convert the pure ground truth (target POI) into noise by continuous time sampling. Then build a model to learn this relative reverse-time process, step by step eliminate noise, and reconstruct the original pure ground truth. It can be described as shown in Fig. 4.

Therefore, for the forward process ($t \in [0, T]$), the proposed model starts with the representation r_i^v of the truth target POI p_i, i.e., $L_u(0) = r_i^v$, and ends with the noisy location archetype, i.e., $L_u(T) = \hat{L}_u$. We model this diffusion evolution with Eq. (24).

$$\mathrm{d}L_u(t) = -\frac{1}{2}\beta(t)L_u(t)\mathrm{d}t + \beta(t)^{\frac{1}{2}}\mathrm{d}\omega(t), \tag{24}$$

where $\beta(t)$ denotes a time-dependent noise function to control the speed of adding noise, which can be linearly increasing or non-linear with time t. $\mathrm{d}\omega(t)$ is the differential of Brownian motion that represents the random components of a dynamical system. The addition of random perturbations gradually brings the data closer to the target distribution (usually a Gaussian distribution).

The crucial aspect of the diffusion phase involves designing a noise schedule $\beta(t)$ to determine the timing and intensity of noise injection at time step t, thereby facilitating uncertainty modeling. We design five types of noise schedule functions for POI recommendation, as Eq. (25). The following Section V will discuss the impact of different functions on recommendation performance and model efficiency.

$$\begin{cases} \beta_1(t) = \beta_{\min} + (\beta_{\max} - \beta_{\min})\frac{t}{T} \\ \beta_2(t) = \beta_{\min} + (\beta_{\max} - \beta_{\min})\sin^2(\frac{\pi t}{2T}) \\ \beta_3(t) = \beta_{\min}\exp\left(\log(\frac{\beta_{\max}}{\beta_{\min}})\frac{t}{T}\right) \\ \beta_4(t) = \beta_{\min} + (\beta_{\max} - \beta_{\min})(\frac{t}{T})^\alpha \\ \beta_5(t) = \beta_{\min}\exp\left(\gamma(\frac{t}{T})\right), \end{cases} \tag{25}$$

where $\beta_{\max}$ and $\beta_{\min}$ are hyperparameters, t represents the current time step, and T denotes the diffusion size.

Correspondingly, for the reverse-time process ($t \in [T, 0]$), it can be formalized as Eq. (26) according to Eq. (2), which starts with $L_u(T) = \hat{L}_u$, and the final output $L_u(0)$ should be as similar to the ground truth r_i^v as possible.

$$\begin{aligned}&\mathrm{d}L_u(t) = \\ &\left[-\tfrac{1}{2}\beta(t)L_u(t) - \beta(t)\nabla_{L_u}\log p_t(L_u)\right]\mathrm{d}t + \beta(t)^{\frac{1}{2}}\,\mathrm{d}\hat{\omega}(t),\end{aligned} \tag{26}$$

where $\nabla_{L_u}\log p_t(L_u)$ indicates the gradient of the log marginal probability.

However, since this marginal probability is difficult to determine, we need to utilize a way to equivalently replace it. Thanks to previous research [30], we are able to train a time-dependent score-based neural network to estimate this marginal probability, i.e., $s_\theta(L_u(t), t) \approx \nabla_{L_u}\log p_t(L_u)$. In addition, to guarantee that the sampled location archetype is in alignment with the user's prior personalized preferences, we introduce the sequence encoding S_u as a context-aware condition embedding into the score-based neural network. Hence, the sampling progress is meticulously guided by the personalized preferences of the user, as Eq. (27).

$$s_\theta(L_u(t), t) = \mathrm{FC}(Concat[L_u(t); S_u]), \tag{27}$$

where $\mathrm{FC}(\cdot)$ represents the stacked multiple fully connected layers with $ReLU$ and $BatchNorm$ operations.

We employ the Fisher divergence as a measure of the loss between the estimated probability distribution $s_\theta(L_u(t), t)$ and the actual marginal probability $\nabla_{L_u}\log p_t(L_u)$ to optimize this score-based neural network, as Eq. (28).

$$\begin{aligned}\mathcal{L}_{div} = \arg\min_{\theta}\ & \mathbb{E}_{t\sim U(0,T)}\{\mathbb{E}_{L_u(t)|L_u(0)} \\ & \{||s_\theta(L_u(t), t) - \nabla_{L_u}\log p_t(L_u(t)|L_u(0))||_2^2\}\},\end{aligned} \tag{28}$$

where $t \sim U(0, T)$ represents the time sampled from an uniform distribution and $||\cdot||_2$ means ℓ_2-$norm$.

In the end, the diffusion-based user preference sampling module can effectively utilize the structural features of the graph and user behavior, learn potential user preferences for the next POI through reverse generation processes, and encode the corresponding output as $L_u = L_u(0) \in \mathbb{R}^d$.

E. Prediction and Optimization

We generate the score probability $\hat{Y}_u \in \mathbb{R}^{|\mathcal{P}|}$ of user u for the next POI, leveraging the obtained sequence encoding $S_u \in \mathbb{R}^d$, location archetype vector $L_u \in \mathbb{R}^d$ and geographic representation $R^{\mathcal{V}} \in \mathbb{R}^{|\mathcal{P}|\times d}$, as Eq. (29).

$$\hat{Y}_u = \lambda S_u \cdot (R^{\mathcal{V}})^\top + (1-\lambda)L_u \cdot (R^{\mathcal{V}})^\top, \tag{29}$$

where $\hat{Y}_u = \{\hat{y}_1^u, \hat{y}_2^u, \ldots, \hat{y}_{|\mathcal{P}|}^u\}$, each element $\hat{y}_i^u$ denotes the probability of user u visiting POI i. λ is a balance coefficient.

We utilize a cross-entropy recommendation loss [41], denoted as Eq. (30), to optimize all trainable parameters $\Theta = \{d, n_h, L, W_*, b_*, \theta\}$, incorporating ℓ_2 regularization to mitigate the over-fitting phenomenon.

$$\mathcal{L}_{rec} = -\sum_{i=1}^{|\mathcal{P}|} y_i^u \log(\hat{y}_i^u) + (1-y_i^u)\log(1-\hat{y}_i^u) + \psi\left\|\Theta\right\|_2^2, \tag{30}$$

Algorithm 1: Learning Algorithm of Diff-DGMN

```
Input: User set $\mathcal{U}$, POI set $\mathcal{P}$, Check-in trajectory
       sequence set $\mathcal{T}_{\mathcal{S}}^{\mathcal{H}}$, Diffusion size $T$
Output: Score probability $\hat{Y}_u = \{\hat{y}_1^u, \hat{y}_2^u, \ldots, \hat{y}_{|\mathcal{P}|}^u\}$
Initialize: Construct the global POI distance graph
            $\mathcal{G}_D = (\mathcal{V}_D, \mathcal{E}_D, \mathcal{A}_D)$ based on $\mathcal{P}$; Model
            parameters $\Theta = \{d, n_h, L, W_*, b_*, \theta\}$
while not converged do
    Sample user sequences $\mathcal{T}_{\mathcal{S}}(u) = \{(p_i^u, t_i^u)\}_{i=1}^n$
    Construct the user-oriented POI transition graph
      $\mathcal{G}_u = (\mathcal{V}_u, \mathcal{E}_u, \mathcal{A}_u)$
    Encode $\mathcal{G}_u$ to the POI sequence encoding $S_u \in \mathbb{R}^d$
      based on Eq. (4) to Eq. (10)
    Encode $\mathcal{G}_D$ to the geographical representation
      $R^{\mathcal{V}} \in \mathbb{R}^{|\mathcal{P}|\times d}$ based on Eq. (11) to Eq. (16)
    Obtain initial location archetype vector $\hat{L}_u$ based
      on the captured representations $S_u$ and $R^{\mathcal{V}}$
    Initialize $L_u(t=0) \leftarrow \hat{L}_u$, the Brownian motion
      $\mathrm{d}\omega(t)$, and differential of time $\mathrm{d}t$
    for $t$ in Range $T$ do
        Conduct forward diffusion process via Eq. (24)
        Calculate $\mathrm{d}L_u$ via Eq. (26)
        if Train with Fisher divergence $\mathcal{L}_{div}$ then
            Sample $t \sim U(0, T)$
            Optimize a time-dependent score-based
              neural network $s_\theta(L_u, t)$ to approximate
              the marginal probability $\nabla_{L_u}\log p_t(L_u)$
        Update $L_u(t) \leftarrow L_u(t-1) + \mathrm{d}L_u$ by reverse
    Optimize Diff-DGMN with $\mathcal{L}_{total} = \zeta\mathcal{L}_{div} + \mathcal{L}_{rec}$
return a ranked score probability list $\hat{Y}_u$ via Eq. (29)
```

where y_i^u is the ground truth in which $y_i^u = 1$ if the user's actual check-in is location i, otherwise, $y_i^u = 0$ and ψ is the corresponding weight of the ℓ_2 regularization.

The total loss of Diff-DGMN is formulated as Eq. (31).

$$\mathcal{L}_{tot} = \zeta\mathcal{L}_{div} + \mathcal{L}_{rec}, \tag{31}$$

where ζ represents a weight factor to balance two types of losses. Algorithm 1 summarizes the recommendation process of the proposed Diff-DGMN model.

F. Computational Analysis

In terms of computational complexity, our proposed Diff-DGMN model primarily involves three parts. (i) Dual-graph-driven representation: its complexity is asymptotically $\mathcal{O}(|\mathcal{E}_u|d_1 + |\mathcal{E}_D|d_1)$, where $|\mathcal{E}_u|$ and $|\mathcal{E}_D|$ represent the edge size of the POI transition graph $\mathcal{G}_u$ and POI distance graph $\mathcal{G}_D$, respectively, and d_1 denotes the embedding size of nodes. (ii) Attention-based location archetype vector generation: $\mathcal{O}(n^2 d_2 \cdot n_h)$, where n is the sequence length, d_2 denotes the embedding size of the captured feature, and n_h is the number of attention heads. (iii) Diffusion-based user preference sampling: $\mathcal{O}(2Td_3)$ for the forward and reverse processes, where T represents the diffusion size and d_3 is the embedding size

TABLE II: Basic data statistics (# denotes the number of).

Dataset	IST	JK	SP	NYC	LA
#Users $\|\mathcal{U}\|$	9,208	5,358	3,722	1,083	965
#POIs $\|\mathcal{P}\|$	11,871	10,706	12,829	9,989	2,541
#Check-ins $\|\mathcal{D}\|$	529,067	300,324	287,642	179,468	37,181
#Edges $\|\mathcal{E}_D\|$	2,593,873	1,315,879	845,298	1,447,008	67,956
Avg. Degree	218.50	121.97	65.88	144.86	26.74
Avg. Visit	57.45	56.05	77.28	165.71	38.53
Avg. Density	0.005	0.005	0.006	0.017	0.015

of diffusion. Since three embedding sizes (d_1, d_2, d_3) are the same, we uniformly use d to represent them. In summary, the total complexity is $\mathcal{O}((|\mathcal{E}_u| + |\mathcal{E}_D| + n^2 \cdot n_h + T) \cdot d)$.

V. Experiment and Result Analysis

A. Experiment Settings

1) Datasets: We evaluate our proposed Diff-DGMN model on five cities: Istanbul (IST) in Turkey, Jakarta (JK) in Indonesia, Sao Paulo (SP) in Brazil, New York City (NYC), and Los Angeles (LA) in the USA. They all are collected from the most popular location-based social networks (LBSNs) service providers, Foursquare [45]. Among them, JK, SP, and NYC are widely used datasets for POI recommendations. IST and LA are new datasets extracted from a global-scale check-in dataset[1] based on latitude and longitude ranges. The time range of all datasets is about 22 months from Apr. 2012 to Jan. 2014. The detailed statistics are summarized in Table II. The dataset of check-in records $\mathcal{D}$ is arranged in chronological order and divided into train $\mathcal{D}_{train}$, validation $\mathcal{D}_{val}$, and test sets $\mathcal{D}_{test}$ using the 8:1:1 split ratio, respectively. $Avg\_Degree = |\mathcal{E}_D| \div |\mathcal{P}|$ represents the average amount of neighbors of each node in distance graph $\mathcal{G}_D$. $Avg\_Visit = |\mathcal{D}| \div |\mathcal{U}|$ reflects the average frequency of POI visits for each user. $Avg\_Density = |\mathcal{D}| \div |\mathcal{U}| \div |\mathcal{P}|$ measures the density of datasets. By comparing the above indicators, it can be found that the five datasets are heterogeneous.

2) Evaluation Metrics: We adopt three commonly utilized metrics for POI recommendation to evaluate the performances of models. The larger values of the three metrics indicate better performance results. We recommend the POIs with Top-k highest scores and select $k = \{1, 5, 10, 20\}$ for evaluation.

(a) **Acc@k** measures the count of accurately recommended POIs within the Top-k recommendation list and its definition is as Eq. (32).

$$Acc@k = \frac{\#|Hit|}{|\mathcal{D}_{test}|}, \tag{32}$$

where $\#|Hit|$ is the total count of the ground-truth target POI that appears in the Top-k recommendation list, and $|\mathcal{D}_{test}|$ indicates the size of the test dataset.

(b) **MRR** is the mean reciprocal rank that measures the mean position of the correct recommended POI in the Top-k ranked recommendation list, as Eq. (33).

$$MRR = \frac{1}{|\mathcal{D}_{test}|} \sum_{i=1}^{|\mathcal{D}_{test}|} \frac{1}{Rank_i}, \tag{33}$$

where $Rank_i$ the position of target POI p_i in the recommendation list. If recommended POIs are not in the Top-k positions, then $Rank_i = \infty$.

(c) **NDCG@k** is the normalized discounted cumulative gain that evaluates the quality of recommendations within the Top-k positions, as Eq. (34).

$$NDCG@k = \begin{cases} \frac{1}{\log_2(Rank_i+1)} & , \quad Rank_i \leq k \\ 0 & , \quad Rank_i > k \end{cases} \tag{34}$$

where k represents the number of recommended POIs.

3) Hyperparameters Selection: During the training phase, we select the Adam optimizer to update all parameters of the Diff-DGMN model with $learning\_rate = 1e^{-3}$, $weight\_decay = 1e^{-3}$, and $max\_epoch = 100$. The embedding size uniformly is $d = 64$ and $batch\_size = 1024$ for all datasets. The distance threshold δ_d is fixed to $1km$ to construct POI distance graph $\mathcal{G}_D$. The number of heads n_h in the multi-head self-attention aggregation is set to $\{2, 4, 4, 2, 2\}$[2], respectively. The number of layers L in GCN is set to $\{2, 3, 4, 2, 2\}$. The balance coefficient $\lambda = \{0.7, 0.8, 0.5, 0.5, 0.7\}$. The weight of ℓ_2 regularization $\psi = 0.001$, dropout probability $dp = 0.4$, and weight factor $\zeta = 0.2$ for all datasets. We choose the linear noise schedule $\beta_1(t)$ from initial $t = 0$ to the maximum diffusion size $T = 1$ with $\mathrm{d}t = 0.01$ at each step. Our experiments were conducted in a PyTorch environment on a Linux server equipped with an NVIDIA Tesla V100-SXM2 GPU card with 128GB of memory.

4) Baseline Methods: To verify the performance of the proposed Diff-DGMN model, we compare it with seven state-of-the-art baseline methods.

- LightGCN (2020) [13] is a collaborative filtering-based approach utilizing graph convolution networks to generate recommendations.
- STAN (2021) [9] employs a dual-layer self-attention mechanism and a multi-modality embedding to achieve interactions, enabling it to predict the next POI for users from non-adjacent and non-successive check-in records.
- DRAN (2022) [11] introduces a propagation way to learn graph-based disentangled representations by purifying two POI graphs. This approach fully leverages influences between distances and transitions for representation.
- GETNext (2022) [21] utilizes spectral GCNs to capture enhanced trajectory flows between different POIs, and employs a Transformer to decode global collaborative signals for the next POI recommendation.
- STHGCN (2023) [23] is a novel hypergraph transformer that effectively integrates the hypergraph structure encoding with implicit spatio-temporal features.
- SLS-REC (2024) [22] develops a spatio-temporal Hawkes attention hypergraph network alongside a dynamic propagation based GNN, designed to capture users' short-term and long-term preferences, respectively.
- PG2Net (2024) [12] applies a graph embedding method, combines Bi-LSTM and attention, to project users' visit sequence into a latent space to capture corresponding sequential dependencies.

[1] https://sites.google.com/site/yangdingqi/home/foursquare-dataset

[2] For brevity, in subsequent sections, the parameter settings for different datasets will be assumed to follow the order {IST, JK, SP, NYC, LA}.

TABLE III: Recommendation performance comparison with baselines on five heterogeneous datasets.

Dataset	Model	Acc@1	Acc@5	Acc@10	Acc@20	NDCG@5	NDCG@10	NDCG@20	MRR
IST	LightGCN	0.1803	0.2651	0.2992	0.3265	0.2160	0.2237	0.2506	0.2279
	STAN	0.2141	0.2996	0.3461	0.3974	0.2586	0.2736	0.2866	0.2548
	DRAN	0.2435	0.3250	0.3685	0.4181	0.2858	0.2998	0.3123	0.2829
	GETNext	0.2643	0.3537	0.3940	0.4387	0.3119	0.3249	0.3361	0.3064
	SLS-REC	0.2680	0.3510	0.3936	0.4431	0.3101	0.3239	0.3363	0.3056
	PG2Net	0.2854	0.3674	0.4105	0.4601	0.3283	0.3421	0.3546	0.3244
	STHGCN	0.3052	0.3875	0.4293	0.4772	0.3486	0.3620	0.3741	0.3445
	Diff-DGMN	**0.3478**	**0.4292**	**0.4694**	**0.5170**	**0.3903**	**0.4033**	**0.4153**	**0.3861**
	Improvement	+13.96%	+10.76%	+9.34%	+8.34%	+11.96%	+11.41%	+11.01%	+12.08%
JK	LightGCN	0.2748	0.3634	0.4003	0.4441	0.3219	0.3338	0.3449	0.3161
	STAN	0.3269	0.4159	0.4554	0.4863	0.3637	0.3865	0.3968	0.3678
	DRAN	0.3682	0.4661	0.5072	0.5493	0.4203	0.4335	0.4442	0.4134
	GETNext	0.3964	0.4942	0.5360	0.5779	0.4488	0.4623	0.4775	0.4421
	SLS-REC	0.4016	0.4977	0.5398	0.5771	0.4523	0.4660	0.4762	0.4458
	PG2Net	0.4306	0.5126	0.5465	0.5800	0.4742	0.4851	0.4938	0.4678
	STHGCN	0.4300	0.5121	0.5461	0.5715	0.4735	0.4846	0.4935	0.4670
	Diff-DGMN	**0.4525**	**0.5309**	**0.5656**	**0.6027**	**0.4941**	**0.5053**	**0.5147**	**0.4891**
	Improvement	+5.09%	+3.57%	+3.49%	+3.91%	+4.20%	+4.16%	+4.23%	+4.55%
SP	LightGCN	0.3014	0.3812	0.4297	0.4500	0.3437	0.3661	0.3713	0.3410
	STAN	0.3524	0.4309	0.4655	0.5028	0.3941	0.4051	0.4147	0.3891
	DRAN	0.3707	0.4540	0.4881	0.5250	0.4182	0.4292	0.4386	0.4134
	GETNext	0.4080	0.4818	0.5124	0.5494	0.4473	0.4572	0.4665	0.4429
	SLS-REC	0.4170	0.4885	0.5200	0.5572	0.4551	0.4652	0.4746	0.4507
	PG2Net	0.4274	0.5024	0.5338	0.5661	0.4674	0.4776	0.4857	0.4622
	STHGCN	0.4721	0.5312	0.5561	0.5833	0.5035	0.5115	0.5184	0.4995
	Diff-DGMN	**0.5307**	**0.5858**	**0.6109**	**0.6355**	**0.5605**	**0.5673**	**0.5743**	**0.5564**
	Improvement	+12.41%	+10.28%	+9.85%	+8.94%	+11.32%	+10.91%	+10.78%	+11.39%
NYC	LightGCN	0.3828	0.4341	0.4565	0.4811	0.3845	0.3969	0.4032	0.3963
	STAN	0.5355	0.5847	0.6045	0.6292	0.5616	0.5681	0.5742	0.5583
	DRAN	0.5118	0.5632	0.5826	0.6013	0.5392	0.5455	0.5502	0.5351
	GETNext	0.5393	0.5902	0.6106	0.6331	0.5667	0.5733	0.5789	0.5631
	SLS-REC	0.5505	0.5952	0.6173	0.6382	0.5743	0.5815	0.5868	0.5718
	PG2Net	0.5667	0.6096	0.6298	0.6514	0.5894	0.5959	0.6014	0.5869
	STHGCN	0.5812	0.6238	0.6443	0.6641	0.6031	0.6098	0.6147	0.6004
	Diff-DGMN	**0.6416**	**0.6768**	**0.6907**	**0.7055**	**0.6608**	**0.6653**	**0.6691**	**0.6583**
	Improvement	+10.39%	+8.50%	+7.20%	+6.23%	+9.57%	+9.10%	+8.85%	+9.64%
LA	LightGCN	0.2042	0.2603	0.3066	0.3527	0.2247	0.2393	0.2509	0.2218
	STAN	0.2899	0.3706	0.4033	0.4402	0.3322	0.3427	0.3521	0.3263
	DRAN	0.3379	0.3981	0.4266	0.4565	0.3687	0.3779	0.3853	0.3648
	GETNext	0.3495	0.4254	0.4593	0.5025	0.3902	0.4013	0.4122	0.3861
	SLS-REC	0.3823	0.4509	0.4853	0.5212	0.4185	0.4296	0.4386	0.4147
	PG2Net	0.4082	0.4671	0.4918	0.5209	0.4397	0.4477	0.4551	0.4359
	STHGCN	0.4469	0.4922	0.5223	0.5549	0.4759	0.4841	0.4913	0.4729
	Diff-DGMN	**0.4871**	**0.5343**	**0.5563**	**0.5798**	**0.5123**	**0.5195**	**0.5255**	**0.5097**
	Improvement	+9.00%	+8.55%	+6.51%	+4.49%	+7.65%	+7.31%	+6.96%	+7.78%

* The improvement rate refers to the performance enhancement of Diff-DGMN (**bold**) over the suboptimal baseline (underline).

B. Performance Comparison with Baselines

The comparison of recommendation performance between the proposed Diff-DGMN model and various baselines across five heterogeneous datasets is presented in Table III. Overall, the Diff-DGMN model outperforms the current state-of-the-art STHGCN method by $3.49\% \sim 13.96\%$ in terms of the Acc@k metric, a $4.16\% \sim 11.96\%$ improvement in NDCG@k, and a $4.55\% \sim 12.08\%$ increase in MRR. These progress are primarily attributed to our model's ability to learn the complexity and stochasticity of user mobility by capturing the high-order sequential encoding from individual user preferences and generating a noise-free location archetype vector for exploring novel user interests. Next, we will discuss in detail from three perspectives: baseline models, heterogeneous datasets, and computational complexity.

In terms of different baselines, attention-based methods (such as STAN, DRAN, and PG2Net) outperform the collaborative filtering-based GCN model (i.e., LightGCN). This superiority is attributed to the attention mechanism's enhanced capability to comprehend complex relationships between nodes (POIs), including but not limited to long-distance dependencies and implicit node interactions. In particular, STAN takes advantage of the dual-layer attention to capture the dependencies among non-adjacent and non-successive visits. DRAN further explicitly unravels the intricate impacts of POIs on users by separately modeling various latent dimensions of POIs. However, they are all inferior to hypergraph-based approaches (like SLS-REC and STHGCN). This observation underscores the utility and complex semantic representation capabilities of graph-structured models for representing sequential patterns. Leveraging various expressive hypergraph

TABLE IV: Time complexity analysis of baseline models.

Model	Time Complexity	Avg. Inference (s)
STAN	$\mathcal{O}((n^2 + n\|\mathcal{P}\| + n) \cdot d)$	98.32
DRAN	$\mathcal{O}((\|\mathcal{E}\|L + n\|\mathcal{P}\| + n^2 \cdot n_h) \cdot d)$	121.05
GETNext	$\mathcal{O}((\|\mathcal{E}\| + nd + n^2 \cdot n_h) \cdot d)$	107.15
PG2Net	$\mathcal{O}((\|\mathcal{E}\| + nh^2 + n^2 + n\|\mathcal{P}\|) \cdot d)$	139.18
SLS-REC	$\mathcal{O}((\|\mathcal{V}_h\|d + \|\mathcal{E}_h\|k + n^2 + \|\mathcal{U}\|^2) \cdot d)$	203.08
STHGCN	$\mathcal{O}((\|\mathcal{V}_h\|d + \|\mathcal{E}_h\|k + n^2 \cdot n_h) \cdot d)$	168.68
Diff-DGMN	$\mathcal{O}((\|\mathcal{E}_u\| + \|\mathcal{E}_D\| + n^2 \cdot n_h + T) \cdot d)$	115.57

networks, the models are enhanced with the ability to capture the high-order similarities between POIs that emerge from sequences. STHGCN employs a similarity-based approach to detect significant correlations among check-in sequences and implements a hypergraph transformer that integrates collective signals between users and within individual users, along with spatio-temporal contexts. Our proposed Diff-DGMN model not only considers features already present in baseline models but also emphasizes the role of bidirectional information. Diff-DGMN introduces two types of weight matrices for in-going $\varphi(in)$ and out-going $\varphi(out)$ during the aggregation of node features, encapsulating both previous and subsequent visits.

Although our model achieves performance improvements across all five heterogeneous datasets, in terms of both hit accuracy rate (Acc@k) and ranking quality (NDCG@k and MRR), the rate of performance enhancement is not consistent across the five datasets. The most notable improvements are observed in IST ($Acc@1 = +13.96\%$, $MRR = +12.08\%$), followed by SP ($Acc@1 = +12.41\%$, $MRR = +11.39\%$) and NYC ($Acc@1 = +10.39\%$, $MRR = +9.64\%$). However, the progress on LA ($Acc@1 = +9.00\%$, $MRR = +7.78\%$) and JK ($Acc@1 = +5.09\%$, $MRR = +4.55\%$) is less obvious.

Drawing upon the fundamental statistical information from Table II, we can infer that the slight performance enhancement on the JK dataset is primarily due to two factors. On the one hand, the average number of connections per node is not high ($avg\_degree = 121.97$), which may result in relatively simplistic or insufficiently interconnected user-POI interactions, consequently affecting the model's ability to learn nuanced preferences. On the other hand, JK exhibits a lower-than-average visitation rate ($avg\_visit = 56.05$), which could imply that the reduced average visit per POI might be insufficient for learned preference features to be reliable. Although LA exhibits the lowest statistical values across all metrics, our Diff-DGMN model has still achieved certain advancements. This can be attributed to the diffusion-based user preference sampling that denoises the captured representation distribution and establishes transfer paths to the target distribution. Diff-DGMN is capable of sampling future preferences of users from the posterior distribution for known historical behaviors, even in the face of the most obvious cold-start problem in LA.

Table IV shows the computational complexity of baseline models and the average inference time of five datasets. Fig. 5 is the comparison of training time on five datasets. For the STAN model, its time cost is primarily due to the dual-module attention computation. In the prediction layer, the significant cost arises from the point-to-point attention matching $\mathcal{O}(n|\mathcal{P}|)$ between historical trajectory points and all POIs. However, since no graph structure is used, the overall complexity is the lowest among all baselines (Its training time is $8.5min/epoch$ on the NYC dataset). The DRAN, GETNext, and PG2Net models introduce graph convolution and transformer structures, resulting in longer training times compared to the STAN model. SLS-REC and STHGCN are hypergraph-based models that require aggregation updates from hyperedges to hypernodes, i.e., $\mathcal{O}(|\mathcal{V}_h|d + |\mathcal{E}_h|k)$, where k represents the average number of nodes per hyperedge. Among the baseline models, SLS-REC has the longest training time (e.g., $22.5min/epoch$ on the NYC) due to the additional inclusion of user-user interactive contrastive learning $\mathcal{O}(|\mathcal{U}|^2 d)$. Our Diff-DGMN model has a moderate training time (e.g., $9.8min/epoch$ on the NYC), which could effectively balance the computational complexity while ensuring the recommendation accuracy. Among the five datasets, IST is the largest size ($|\mathcal{E}| \approx 2.6 \times 10^6$), which results in all models spending over $20min/epoch$ on training. But the average inference time of all models is acceptable (the longest being SLS-REC at $203.08s$, and our Diff-DGMN at $115.57s$).

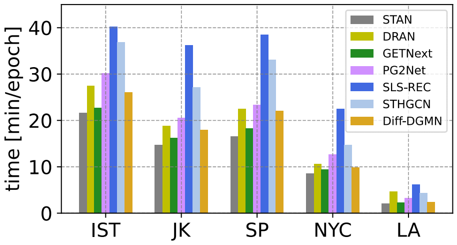


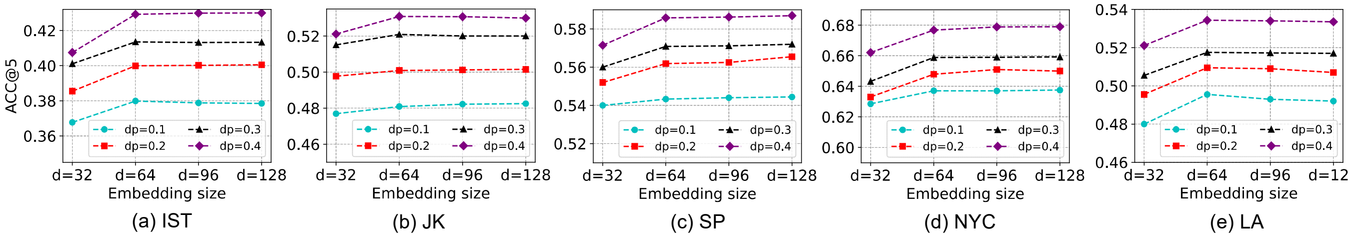
Fig. 5: Comparison of training time on five datasets.

C. Hyperparameter Analysis

We investigated the impact of four crucial hyperparameters (i.e., the number of heads n_h in the multi-head self-attention, the number of layers L in GCN, embedding size d, and dropout probability d_p). The results are depicted in Fig. 6 and Fig. 7. We can find the following observations:

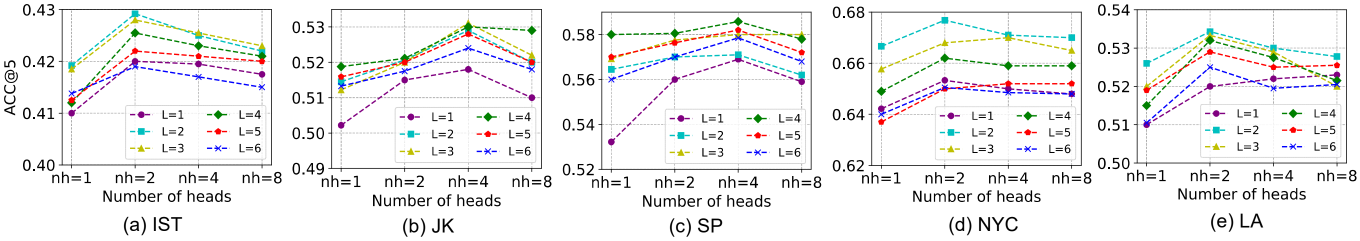
(i) We incrementally escalated the embedding size d from 32 to 128 in increments of 32 and the dropout probability d_p from 0.1 to 0.4. Generally, our model achieved satisfactory performance at a medium embedding size ($d = 64$). Enhancing d further introduced additional computational complexity while yielding only nominal performance gains. Since a user's interest in POIs can be influenced by a multitude of factors, which may not be fully apparent in the training data, increasing dropout probability d_p can emulate this uncertainty. This reduction in reliance on specific training samples aids in enhancing the Diff-DGMN model's predictive capacity for unseen data.

(ii) The number of heads n_h in the multi-head attention had variation from 1 to 8 to facilitate effective interactions among POIs within distinct representation subspaces. The layers L of the stacked GCN were increased from 1 to 6 to obtain a wider receptive field. Both a single head ($n_h = 1$) and an excessive number of heads ($n_h = 8$) led to diminished results, indicating that an optimal selection is either 2 or 4. Similarly, performance degraded when the number of GCN layers exceeded 4, which possibly might be the over-smoothing issue inherent in GCN architectures.



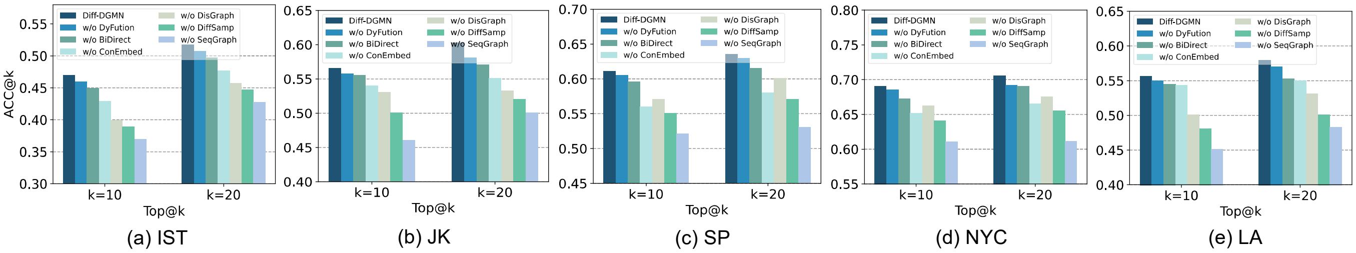
(a) IST (b) JK (c) SP (d) NYC (e) LA

Fig. 6: Performance of Diff-DGMN under different values of the embedding size d and dropout probability d_p.



(a) IST (b) JK (c) SP (d) NYC (e) LA

Fig. 7: Performance of Diff-DGMN under different values of the number of heads n_h and the number of layers L.



(a) IST (b) JK (c) SP (d) NYC (e) LA

Fig. 8: Ablation study results and the performance of different variants on five heterogeneous datasets.

(iii) It is crucial to enable the model to learn common features of different user behaviors when dealing with recommendation systems with complex user mobility and highly personalized preferences. There is an optimal combination of n_h and L for each dataset, i.e., $\{(2,2),(4,3),(4,4),(2,2),(2,2)\}$, underscoring the necessity of considering the specific characteristics of the different dataset when tuning these parameters.

D. Ablation Study

To validate the contribution of each core component in our Diff-DGMN, we performed an ablation study. Diff-DGMN is referred to as the foundational model from which we remove various components to derive six distinct variants.

- **w/o SeqGraph**: this variant removes the direction-aware sequence graph multi-scale representation module.
- **w/o BiDirect**: this variant replaces the bi-direction encoder $\varphi(\cdot)$ with a plain GAT in POI transition graph $\mathcal{G}_u$.
- **w/o DisGraph**: this variant removes the global-based distance graph geographical representation module.
- **w/o DyFution**: this variant replaces the dynamic function encoder $F_d(\cdot)$ with a plain GCN in distance graph $\mathcal{G}_D$.
- **w/o DiffSamp**: this variant removes the diffusion-based user preference sampling and obtained $\hat{L}_u$ from the attention-based location archetype generation module is directly output for calculation of the score probability $\hat{Y}_u$.
- **w/o ConEmbed**: this variant replaces the context-aware condition embedding S_u into the score-based neural network $FC(\hat{L}_u|S_u)$ with only $FC(\hat{L}_u)$.

The results of the ablation study are displayed in Fig. 8, where the "SeqGraph" component is shown to be the most critical. Removing this component results in an average decrease of 18% in ACC@10 and 16% in ACC@20 across five datasets, underscoring the necessity of modeling users' personalized preferences and habitual visitation patterns. The second most critical component is "DiffSamp". The diffusion-based user preference sampling can generate more pertinent and novel POIs that cater to the user's future preferences, thereby enhancing the recommendation. Eliminate this component, as the performance degradation of the Diff-DGMN model becomes more severe when the number of recommended POIs is high (i.e., $k = 20$). Furthermore, the components "DisGraph" and "ConEmbed" demonstrate varying levels of importance across different datasets. The former emphasizes utilizing the spatial distribution patterns of POIs for recommending subsequent locations, while the latter focuses on addressing users' personalized needs. Specifically, the POI geographic distributions in the cities of IST, JK, and LA demonstrate a more pronounced clustering effect (e.g., POIs within the same business district may share similar customer bases), making the global-based distance graph geographical representation more prominent in these three datasets.

In contrast, users in SP and NYC exhibit more personalized preferences, with their next visits being more significantly influenced by historical visitation patterns. The context-aware condition embedding S_u into the score-based neural network positively impacts the sampling of an effective location preference. Although Diff-DGMN can benefit directly from the

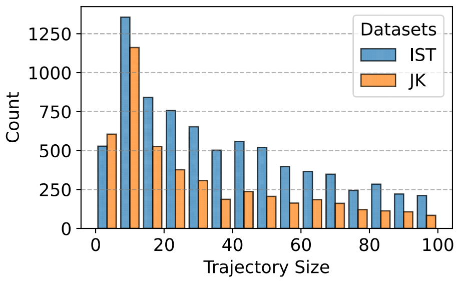


Fig. 9: Statistic of trajectory sizes on IST and JK.

location posterior compared to this variant “w/o ConEmbed”, it experiences a decline in performance due to the omission of user-specific historical visitation patterns during the controlled diffusion process. Compared to the aforementioned four components, “BiDirect” and “DyFution” have a lesser impact on the model. However, there is still a performance gap between them and the original Diff-DGMN model. The bi-direction encoder $\varphi(\cdot)$ considers potential transitions from the current POI node to other nodes, enhancing the modeling of sequence dependencies. The dynamic function encoder $F_d(\cdot)$ of distances adaptively learns the spatial patterns of nodes, rather than statically applying the same treatment to all distances. Users typically prefer to visit POIs closer to their current location; thus, capturing this dependency more effectively reflects the strength of relationships between different locations and can help the recommendation system more accurately predict the user’s next check-in intention.

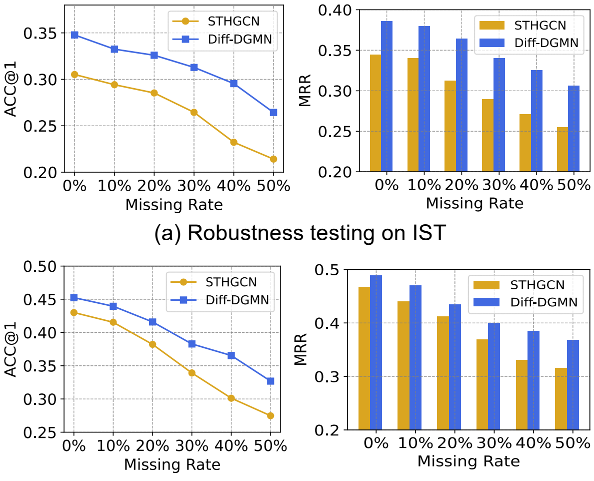
E. Robustness Analysis

To investigate the potential impact of different historical check-in trajectory sequence lengths n on the model, we categorize trajectory types based on their lengths. Specifically, the top 30% of lengths are labeled as long trajectories, the bottom 30% as short trajectories, and the remaining 40% as middle trajectories. Fig. 9 is the statistical histogram distribution of trajectory sizes. We compare our results with the optimal baseline model STHGCN, as shown in Table V. Due to the limited spatio-temporal contextual features of short trajectories, the model may not be able to capture long-term user preferences, resulting in performance degradation compared to long trajectories. Long sequences can help improve the diversity and coverage of recommendations, as the model can utilize richer interactions between users and POIs. In addition, processing trajectories of different lengths requires the model to have good robustness and maintain consistency in recommendation results. On the JK dataset, for the STHGCN model, the top-1 accuracy is 41.87% and 45.55% of short and long trajectories, which is sensitive to trajectory length. For our Diff-DGMN, accuracy is 44.77% and 46.02% which did not exhibit significant performance fluctuations and had better robustness compared with STHGCN.

To further test the robustness of our Diff-DGMN model under data perturbations, we set a missing rate $m_r = \{10\%, 20\%, 30\%, 40\%, 50\%\}$ and randomly deleted a certain amount of historical trajectory data from the training set. The results comparing with STHGCN are shown in Fig. 10. As the missing rate increases from 0% to 50%, the top-1 accuracy metric for both Diff-DGMN and STHGCN shows a decreasing trend. However, Diff-DGMN shows superior performance over STHGCN, maintaining higher accuracy at all levels of data removal. Specifically, STHGCN’s performance drops by 32.95%, while Diff-DGMN’s performance decreases by 25.87%. The MRR metric follows a similar pattern, with Diff-DGMN achieving higher scores across all missing rates compared to STHGCN. The performance gap is particularly noticeable at higher missing rates (i.e., $m_r > 30\%$), where Diff-DGMN shows a slower decline in MRR, further underscoring its robustness. This can be attributed to the learning mechanism of the diffusion model. Most deep learning methods rely heavily on training data and are susceptible to noisy data. In contrast, the denoising process of the diffusion model can mitigate this issue to some extent. Based on this study, we can find that Diff-DGMN maintains higher accuracy even with more data absent, demonstrating better robustness against data perturbations.

TABLE V: Results of trajectory type on IST and JK.

Type	Model	IST		JK	
		Acc@1	MRR	Acc@1	MRR
Short	STHGCN	0.2901	0.3409	0.4187	0.4586
Middle	STHGCN	0.3055	0.3445	0.4259	0.4651
Long	STHGCN	0.3250	0.3585	0.4555	0.4723
Short	Diff-DGMN	0.3358	0.3807	0.4477	0.4801
Middle	Diff-DGMN	0.3463	0.3862	0.4513	0.4885
Long	Diff-DGMN	0.3502	0.3897	0.4602	0.5012



(a) Robustness testing on IST

(b) Robustness testing on JK

Fig. 10: Robustness analysis under different missing rates.

F. Study on Noise Schedule Functions

In this section, we will delve into the impact of SDE-based diffusion models on the performance of recommendation systems in POI recommendation tasks. Think back to the variance-preserving stochastic differential equation (VP-SDE) we designed, as Eq. (24). The $\beta(t)$ function controls the speed of introducing random noise, which is crucial for diffusion models. We have designed five kinds of $\beta(t)$ functions as Eq. (25), which are visualized as Fig. 11 when the parameters is determined to $\beta_{\max} = 20$, $\beta_{\min} = 0.1$, and maximum

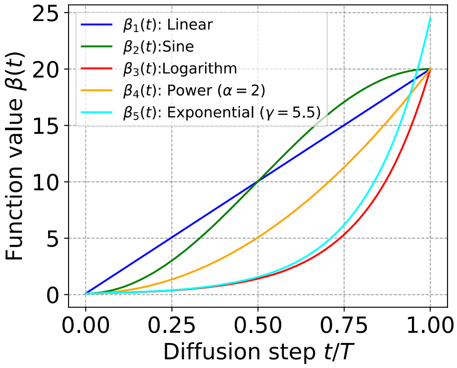


Fig. 11: Diffusion with different noise schedules $\beta(t)$.

diffusion size $T = 1$ with $\mathrm{d}t = 0.01$ at each step. In order to investigate the impact of different noise variations on recommendation performance, we conducted experiments on five cities, and the results are shown in Table VI.

We can observe the following: (1) The accuracy of recommendations is not significantly influenced by different $\beta(t)$ schedules. Noise is not sensitive to recommendation results, indicating the robustness of the proposed Diff-DGMN model. Furthermore, the previous research [46] also highlights that while different schedules may have some impact on the final outcomes, they are unlikely to result in significant fluctuations in earnings. (2) This indirectly confirms that the VP-SDE ensures the preservation of variance in the data generation process, thereby enhancing the stability of the diffusion model. In other words, $\partial\beta(t)/\partial L_u = 0$, this means that the diffusion coefficient is independent of the variable L_u, thus maintaining the variance. (3) Although different $\beta(t)$ may not significantly affect the accuracy of learning paths (i.e., whether learners ultimately reach their goals), they may impact the convergence speed of the learning paths. The linear $\beta_1(t)$ exhibits the fastest convergence speed, while nonlinear functions may introduce more complex noise patterns, potentially causing the recommendation system to more easily fall into local extremum or oscillations during the optimization process, thus affecting convergence speed.

Although there is no direct mathematical theory to explain why linear noise schedules are superior to non-linear, we can try to understand it from the perspective of the Markov property. Discrete diffusion processes can be regarded as Markov chains, where the probability distribution of the current state p_{t+1} depends only on the previous state p_t. Hence, the formula $p_{t+1} = \mathcal{M}_t \cdot p_t$ holds. $\mathcal{M}_t$ is a transition matrix that describes the probability of state transition from t to $t+1$. Then, consider introducing the noise $\beta(t)$ into the diffusion process to adjust the noise level in the state transition matrix $\mathcal{M}_t$. If the noise is linear, we can represent it as Eq. (35). Among them, there are two types of states, and the sum of probabilities for each state is 1 (i.e., the sum of each row of $\mathcal{M}_t$ is 1).

$$\mathcal{M}_t = \begin{bmatrix} 1-\beta_1(t) & \beta_1(t) \\ \beta_1(t) & 1-\beta_1(t) \end{bmatrix} \tag{35}$$

The linear noise $\beta_1(t)$ can directly act on the elements of the transition matrix $\mathcal{M}_t$, making our Diff-DGMN model easier to analyze and understand. By contrast, if the noise function is nonlinear, the construction of $\mathcal{M}_t$ would become more complex. Therefore, linear $\beta_1(t)$ is in line with the Markov property and more suitable for modeling the diffusion process of user location preferences.

TABLE VI: Performance with different noise schedules.

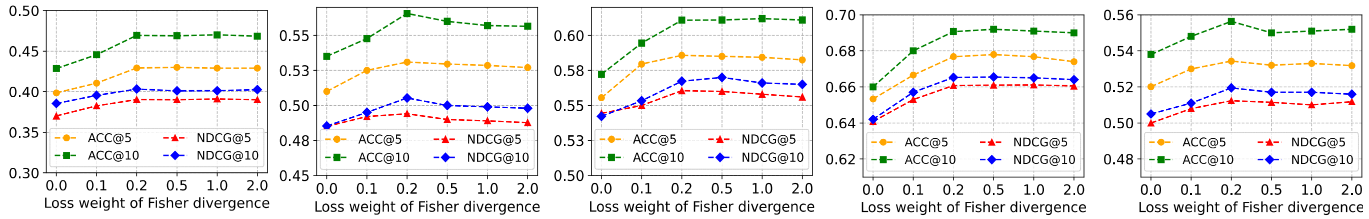
Dataset	Schedule	Acc@5	NDCG@5	MRR	Conv.Epoch
IST	$\beta_1(t)$	**0.4292**	**0.3903**	**0.3861**	**38**
	$\beta_2(t)$	0.4251	0.3833	0.3790	46
	$\beta_3(t)$	0.4255	0.3851	0.3796	62
	$\beta_4(t)$	0.4192	0.3830	0.3740	55
	$\beta_5(t)$	0.4145	0.3799	0.3705	66
JK	$\beta_1(t)$	**0.5309**	**0.4941**	**0.4891**	**68**
	$\beta_2(t)$	0.5271	0.4892	0.4803	75
	$\beta_3(t)$	0.5268	0.4892	0.4801	80
	$\beta_4(t)$	0.5275	0.4909	0.4815	72
	$\beta_5(t)$	0.5112	0.4817	0.4752	88
SP	$\beta_1(t)$	0.5858	**0.5605**	**0.5564**	**70**
	$\beta_2(t)$	0.5815	0.5585	0.5503	82
	$\beta_3(t)$	0.5821	0.5592	0.5534	90
	$\beta_4(t)$	**0.5860**	0.5601	0.5562	87
	$\beta_5(t)$	0.5744	0.5500	0.5482	92
NYC	$\beta_1(t)$	**0.6768**	**0.6608**	**0.6583**	**42**
	$\beta_2(t)$	0.6692	0.6517	0.6488	53
	$\beta_3(t)$	0.6693	0.6504	0.6471	65
	$\beta_4(t)$	0.6710	0.6550	0.6525	55
	$\beta_5(t)$	0.6670	0.6505	0.6482	68
LA	$\beta_1(t)$	**0.5343**	**0.5123**	**0.5097**	**30**
	$\beta_2(t)$	0.5318	0.5090	0.5066	38
	$\beta_3(t)$	0.5321	0.5106	0.5080	42
	$\beta_4(t)$	0.5280	0.5070	0.5042	56
	$\beta_5(t)$	0.5251	0.5033	0.5002	65

The Fisher divergence $\mathcal{L}_{div}$ is proposed to optimize the time-dependent score-based neural network $s_\theta(L_u(t), t)$ to estimate this actual marginal probability $\nabla_{L_u} \log p_t(L_u)$, which makes Diff-DGMN model sample a more fine-grained location that is close to the target POI. In order to investigate the impact of different loss weights of $\mathcal{L}_{div}$ on recommendation performance as Eq. (31), we conducted experiments across five datasets, and results are presented in Fig. 12. The findings on five datasets are consistent. Diff-DGMN achieves optimal performance when the parameter ζ falls within the range of $0.2 \sim 0.5$. When $\zeta < 0.2$, the model's performance suffers due to the lack of a measure for a similarity between the probability distribution learned by the diffusion process at the current time step and the target probability distribution. Conversely, when $\zeta > 0.5$, the Diff-DGMN model overly focuses on minimizing the divergence loss, neglecting the cross-entropy recommendation loss $\mathcal{L}_{rec}$. This leads to a slight decrease or maintenance of accuracy.

G. Discussions

The theory for diffusion models comes from the diffusion process in non-equilibrium thermodynamics [29], where the system gradually reaches a state of equilibrium. Similarly, user preferences also follow a process from irregularity to regularity.

In this work, we model users' visit intentions and dynamic interests as probability distributions, using the stochastic differential equation (SDE) based paradigm to solve these distributions. SDE aligns more closely with the inherent nature of user interests, where the drift term determines the average



(a) IST (b) JK (c) SP (d) NYC (e) LA

Fig. 12: Diffusion with different Fisher divergence weights ζ on five heterogeneous datasets.

component (corresponding to the average expectation of interests), and the diffusion term represents random changes (i.e., the variance of interests). Moreover, the reverse SDE relies only on the gradient field (i.e., a score-based neural network $s_\theta(t)$) at each time point t, which allows us to capture changes in user interests more accurately.

Previous methods, such as score matching with Langevin dynamics [27] and denoising diffusion probabilistic models [28], gradually perturb the data through discrete noise levels. In contrast, our designed noise schedules implement continuous noise, making the transition from data to noise smoother. On the other hand, most recommendation models depend heavily on high-quality supervised data and parameter tuning, making them susceptible to noise and lacking rigorous theoretical guarantees. However, our generative diffusion model can be viewed as a Markov chain, which theoretically ensures convergence to the target distribution after a sufficient number of diffusion steps, allowing models to better handle uncertainties in the data.

As the increasing of data size, the computational complexity will inevitably grow. For larger-size data, e.g., when $|\mathcal{E}| > 5 \times 10^6$, the training time for Diff-DGMN will exceed $60min/epoch$. In this case, we can perform an edge resampling strategy in the dual-graph-driven representation module. It involves removing redundant edges while preserving the overall structure of the graphs. Subsequently, the graphs can be partitioned into multiple mini-batches (subgraphs) and processed in parallel. In the diffusion-based user preference sampling module, sparse matrix operations can be employed to accelerate the diffusion process. For instance, using the singular value decomposition (SVD) to approximate large matrices as low-rank matrices can significantly reduce computational complexity. Furthermore, an incremental update method can be adopted for continuously updating data. We then perform diffusion calculations only on newly added data, rather than recalculating the diffusion process for the entire dataset, thereby effectively reducing unnecessary repeated computations, which enables our model to be applicable to larger-scale data.

VI. Conclusion

In this paper, we proposed a Diffusion-based Dual Graph Multi-attention Network (Diff-DGMN) for the next POI recommendation. By constructing local POI transition graphs based on user check-in trajectories and global POI distance graphs based on geographical distances, we can capture the complex spatial relationships between POIs from multiple perspectives. The local transition graph reveals user behavior patterns and preferences, while the global distance graph reflects the influence of geographical factors on user choices. The diffusion model effectively integrates these complex spatial relationships by simulating the propagation process of information on these graph structures. Over time, user interests and preferences may change, and the relationships between POIs may also be updated. Leveraging diffusion-based user preference sampling, our Diff-DGMN approach can naturally adapt to this dynamic change, providing users with relevant recommendations.

Despite the significant performance achieved, our current research still has its own limitations. First, the generation method relies on prior knowledge, which will be constrained by known information. Furthermore, the proposed model currently adopts a single-scale diffusion process, which prevents it from fully exploiting the feature information at different scales, thus it might limit the model's flexibility. To tackle the first limitation, we might transit from a prior-knowledge-guided generation to a novel-knowledge-guided generation in further work. It will allow the diffusion model to discover novel knowledge during learning processes and incorporate them as the discovery-aware condition embedding into the generation process. As to the second limitation, we could explore the multi-scale diffusion, enabling the model to extract and integrate features at different scales, thereby enhancing the detail and hierarchical quality of the generated results.

References

[1] P. Sánchez and A. Bellogín, "Point-of-interest recommender systems based on location-based social networks: a survey from an experimental perspective," *ACM Computing Surveys*, vol. 54, no. 11s, pp. 1–37, 2022.

[2] C. Gao, Y. Zheng, N. Li, Y. Li, Y. Qin, J. Piao, Y. Quan, J. Chang, D. Jin, X. He *et al.*, "A survey of graph neural networks for recommender systems: Challenges, methods, and directions," *ACM Trans. Recommender Syst.*, vol. 1, no. 1, pp. 1–51, 2023.

[3] D. Yang, D. Zhang, V. W. Zheng, and Z. Yu, "Modeling user activity preference by leveraging user spatial temporal characteristics in LBSNs," *IEEE Trans. Syst. Man Cybern. Syst.*, vol. 45, no. 1, pp. 129–142, 2014.

[4] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme, "Factorizing personalized markov chains for next-basket recommendation," in *Proc. of Int. Conf. on World Wide Web (WWW)*, 2010, pp. 811–820.

[5] Q. Liu, S. Wu, L. Wang, and T. Tan, "Predicting the next location: A recurrent model with spatial and temporal contexts," in *Proc. of AAAI Conf. on Artif. Intell.*, vol. 30, no. 1, 2016.

[6] P. Zhao, A. Luo, Y. Liu, J. Xu, Z. Li, F. Zhuang, V. S. Sheng, and X. Zhou, "Where to go next: A spatio-temporal gated network for next POI recommendation," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 5, pp. 2512–2524, 2022.

[7] Y. Wu, K. Li, G. Zhao, and X. Qian, "Personalized long-and short-term preference learning for next POI recommendation," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 4, pp. 1944–1957, 2022.

[8] L. Zhang, Z. Sun, J. Zhang, Y. Wu, and Y. Xia, "Conversation-based adaptive relational translation method for next POI recommendation with uncertain check-ins," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 10, pp. 7810–7823, 2023.

[9] Y. Luo, Q. Liu, and Z. Liu, "STAN: Spatio-temporal attention network for next location recommendation," in *Proc. of the World Wide Web (WWW)*, 2021, pp. 2177–2185.

[10] J. Xia, Y. Yang, S. Wang, H. Yin, J. Cao, and S. Y. Philip, "Bayes-enhanced multi-view attention networks for robust POI recommendation," *IEEE Trans. Knowl. Data Eng.*, pp. 1–14, 2023.

[11] Z. Wang, Y. Zhu, H. Liu, and C. Wang, "Learning graph-based disentangled representations for next POI recommendation," in *Proc. of Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2022, pp. 1154–1163.

[12] B. Wang, H. Li, W. Wang, M. Wang, Y. Jin, and Y. Xu, "PG2Net: Personalized and group preferences guided network for next place prediction," *IEEE Trans. Intell. Transp. Syst.*, pp. 1–16, 2024.

[13] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, "LightGCN: Simplifying and powering graph convolution network for recommendation," in *Proc. of Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2020, pp. 639–648.

[14] Z. Li, A. Sun, and C. Li, "DiffuRec: A diffusion model for sequential recommendation," *ACM Trans. Inf. Syst.*, vol. 42, no. 3, pp. 1–28, 2023.

[15] C. Wang, M. Yuan, R. Zhang, K. Peng, and L. Liu, "Efficient point-of-interest recommendation services with heterogenous hypergraph embedding," *IEEE Trans. Serv. Comput.*, vol. 16, no. 2, pp. 1132–1143, 2023.

[16] W. Ju, Y. Qin, Z. Qiao, X. Luo, Y. Wang, Y. Fu, and M. Zhang, "Kernel-based substructure exploration for next POI recommendation," in *IEEE Int. Conf. on Data Mining (ICDM)*, 2022, pp. 221–230.

[17] Y. Qin, Y. Wang, F. Sun, W. Ju, X. Hou, Z. Wang, J. Cheng, J. Lei, and M. Zhang, "DisenPOI: Disentangling sequential and geographical influence for point-of-interest recommendation," in *Proc. of ACM Int. Conf. on Web Search and Data Mining (WSDM)*, 2023, pp. 508–516.

[18] X. Xu, T. Suzumura, J. Yong, M. Hanai, C. Yang, H. Kanezashi, R. Jiang, and S. Fukushima, "Revisiting mobility modeling with graph: A graph transformer model for next point-of-interest recommendation," in *Proc. of ACM Int. Conf. on Advances in Geographic Inf. Syst. (SIGSPATIAL)*, 2023, pp. 1–10.

[19] J. Zhang, Y. Li, R. Zou, J. Zhang, Z. Fan, and X. Song, "Hyper-relational knowledge graph neural network for next POI," *World Wide Web*, vol. 27, no. 46, pp. 1–19, 2024.

[20] X. Wang, F. Fukumoto, J. Cui, Y. Suzuki, J. Li, and D. Yu, "EEDN: Enhanced encoder-decoder network with local and global context learning for POI recommendation," in *Proc. of Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2023, pp. 383–392.

[21] S. Yang, J. Liu, and K. Zhao, "GETNext: trajectory flow map enhanced transformer for next POI recommendation," in *Proc. of Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2022, pp. 1144–1153.

[22] J. Fu, R. Gao, Y. Yu, J. Wu, J. Li, D. Liu, and Z. Ye, "Contrastive graph learning long and short-term interests for POI recommendation," *Expert Systems with Applications*, vol. 238, p. 121931, 2024.

[23] X. Yan, T. Song, Y. Jiao, J. He, J. Wang, R. Li, and W. Chu, "Spatio-temporal hypergraph learning for next POI recommendation," in *Proc. of Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2023, pp. 403–412.

[24] Y. Yang, M. Jin, H. Wen, C. Zhang, Y. Liang, L. Ma, Y. Wang, C. Liu, B. Yang, Z. Xu *et al.*, "A survey on diffusion models for time series and spatio-temporal data," *preprint arXiv:2404.18886*, 2024.

[25] Z. Guo, K. Yu, N. Kumar, W. Wei, S. Mumtaz, and M. Guizani, "Deep-distributed-learning-based POI recommendation under mobile-edge networks," *IEEE Internet of Things Journal*, vol. 10, no. 1, pp. 303–317, 2022.

[26] F.-A. Croitoru, V. Hondru, R. T. Ionescu, and M. Shah, "Diffusion models in vision: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 9, pp. 10 850–10 869, 2023.

[27] Y. Song and S. Ermon, "Generative modeling by estimating gradients of the data distribution," *Adv. in Neural Inf. Process. Syst. (NeurIPS)*, vol. 32, pp. 11 895–11 907, 2019.

[28] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Adv. in Neural Inf. Process. Syst. (NeurIPS)*, vol. 33, pp. 6840–6851, 2020.

[29] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," in *Int. Conf. on Machine Learning (ICML)*, 2015, pp. 2256–2265.

[30] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-based generative modeling through stochastic differential equations," in *Int. Conf. on Learning Representations (ICLR)*, 2021.

[31] D. Liang, R. G. Krishnan, M. D. Hoffman, and T. Jebara, "Variational autoencoders for collaborative filtering," in *Proc. of the World Wide Web (WWW)*, 2018, pp. 689–698.

[32] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[33] H. Wen, Y. Lin, Y. Xia, H. Wan, Q. Wen, R. Zimmermann, and Y. Liang, "DiffSTG: Probabilistic spatio-temporal graph forecasting with denoising diffusion models," in *Proc. of ACM Int. Conf. on Advances in Geographic Inf. Syst. (SIGSPATIAL)*, 2023, pp. 1–12.

[34] L. Lin, D. Shi, A. Han, and J. Gao, "SpecSTG: A fast spectral diffusion framework for probabilistic spatio-temporal traffic forecasting," *preprint arXiv:2401.08119*, 2024.

[35] Y. Zhu, C. Wang, and H. Xiong, "Towards graph-aware diffusion modeling for collaborative filtering," *preprint arXiv:2311.08744*, 2023.

[36] W. Wang, Y. Xu, F. Feng, X. Lin, X. He, and T.-S. Chua, "Diffusion recommender model," in *Proc. of Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2023, pp. 832–841.

[37] H. Ma, R. Xie, L. Meng, X. Chen, X. Zhang, L. Lin, and Z. Kang, "Plug-in diffusion model for sequential recommendation," in *AAAI*, 2024.

[38] Y. Qin, H. Wu, W. Ju, X. Luo, and M. Zhang, "A diffusion model for POI recommendation," *ACM Trans. Inf. Syst.*, vol. 42, no. 2, pp. 1–27, 2023.

[39] J. Long, G. Ye, T. Chen, Y. Wang, M. Wang, and H. Yin, "Diffusion-based cloud-edge-device collaborative learning for next POI recommendations," *preprint arXiv:2405.13811*, 2024.

[40] B. D. Anderson, "Reverse-time diffusion equation models," *Stochastic Processes and their Applications*, vol. 12, no. 3, pp. 313–326, 1982.

[41] Z. Wang, Y. Zhu, Q. Zhang, H. Liu, C. Wang, and T. Liu, "Graph-enhanced spatial-temporal network for next POI recommendation," *ACM Trans. Knowl. Discovery from Data*, vol. 16, no. 6, pp. 1–21, 2022.

[42] F. Sun, J. Liu, J. Wu, C. Pei, X. Lin, W. Ou, and P. Jiang, "BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer," in *Proc. of ACM Int. Conf. on Inf. Knowl. Manage. (CIKM)*, 2019, pp. 1441–1450.

[43] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Adv. in Neural Inf. Process. Syst. (NeurIPS)*, vol. 30, 2017.

[44] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *14th European Conference on Computer Vision (ECCV)*. Springer, 2016, pp. 630–645.

[45] D. Yang, B. Qu, J. Yang, and P. Cudre-Mauroux, "Revisiting user mobility and social relationships in LBSNs: a hypergraph embedding approach," in *Proc. of the World Wide Web*, 2019, pp. 2147–2157.

[46] Z. He, T. Sun, K. Wang, X. Huang, and X. Qiu, "DiffusionBERT: Improving generative masked language models with diffusion models," in *Proc. of the Annu. Meeting Assoc. Comput. Linguistics (ACL)*, vol. 1, 2023, pp. 4521–4534.

Jiankai Zuo received the B.S. degree in computer science from Shenyang Aerospace University, Shenyang, China, in 2020. He is currently pursuing the Ph.D. degree in computer science with the Key Laboratory of Embedded System and Service Computing, Tongji University, Shanghai. His research interests include spatial-temporal data mining, intelligent transportation systems, and POI recommendation.

Yaying Zhang (Member, IEEE) received the B.S. degree in computer science and the M.S. degree in electrical engineering from the Shandong University of Science and Technology, Shandong, China, respectively, and the Ph.D. degree in computer science from Shanghai Jiaotong University, Shanghai, China, in 2004. She is currently a Professor with the Key Laboratory of Embedded System and Service Computing, Tongji University, Shanghai. Her research interests include spatial-temporal data analysis, data mining, and intelligent transportation systems.