IDDIFF: INTENT-AWARE DETERMINISTIC DIFFUSION MODEL FOR NEXT LOCATION PREDICTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Next location prediction aims to infer users' next destinations from their historical check-in trajectories. Although diffusion models offer a promising direction due to powerful generative capabilities, existing diffusion-based approaches encounter two critical challenges: **(i)** check-in trajectories naturally entangle visits driven by diverse intents. Existing methods rely on raw check-in trajectory representations as contextual conditions to guide diffusion models, which can lead to learned representations that are inconsistent with the user's authentic intents, ultimately yielding suboptimal performance; **(ii)** our experiment reveals a limitation when applying standard Gaussian noise to location prediction. The injected randomness may disrupt personalized preference structures, limiting their ability to model users' behaviors. To address these challenges, we propose **IDDiff**, an **I**ntent-aware **D**eterministic **Diff**usion framework for next location prediction. Specifically, the Sequence-aware Trajectory Representation Module constructs a user's global trajectory-level intent representation. It then divides the trajectory into multiple local subtrajectories and clusters semantically similar sub-trajectory representations to construct trajectory intent prototypes, which serve as contextual conditions to guide the diffusion model. Furthermore, we develop a Intent Refinement Module to address the structural disruption and reconstruction uncertainty caused by stochastic noise. Rather than injecting random Gaussian perturbations, the module constructs a predefined and controllable preference transformation trajectory, thereby preserving personalized mobility preference structures. Extensive experiments on five real-world datasets demonstrate that IDDiff consistently outperforms state-of-the-art methods. The implementation code is publicly available at https://anonymous.4open.science/r/IDDiff-7B4F.

1 INTRODUCTION

The rapid growth of Location-Based Social Networks (LBSNs), such as Foursquare, Gowalla, Yelp, and Weibo, has generated large-scale spatio-temporal check-in data, providing valuable opportunities to understand personalized mobility patterns and support location-aware services Wang et al. (2024b); Zuo & Zhang (2024); Song et al. (2025); Lin et al. (2026). Next location prediction is a fundamental task in location-based services and human mobility analysis, aiming to infer a user's next destination from historical check-in trajectories Song et al. (2025); Chen et al. (2025); Qin et al. (2025).

Existing methods have evolved from early RNN-based sequential modeling to attention-based spatio-temporal interaction modeling, and further to graph- and hypergraph-based approaches for capturing higher-order dependencies among locations Liu et al. (2016); Feng et al. (2018); Luo et al. (2021); Chen et al. (2025); Yang et al. (2022); Song et al. (2025); Qin et al. (2025); Lin et al. (2026). More recently, diffusion models have been introduced into recommender systems because of their strong distribution-learning and generative capabilities Ho et al. (2020); Song et al. (2020). By progressively perturbing data and

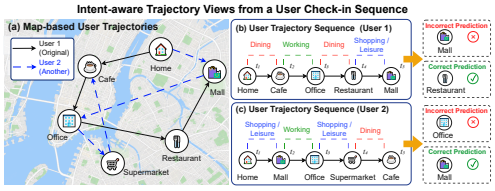


Figure 1: Illustration of temporally discontinuous latent intents in user check-in trajectories.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

learning the reverse denoising process, they can approximate latent user preference distributions Zuo & Zhang (2024); Wang et al. (2024b); Pan et al. (2025); Lan et al. (2026).

Despite their effectiveness, existing diffusion-based methods still suffer from two crucial limitations. **First**, in real-world users' mobility trajectories, visiting behaviors driven by the same intent are often temporally discontinuous. As illustrated in Figure 1, semantically related visits may be separated by check-ins driven by other intents, causing multiple intents to become entangled within the same trajectory. However, directly using raw trajectory representations that entangle multiple intents to guide diffusion model training may cause the learned user behavior representations to deviate from users' true preferences, thereby limiting the ability to model users' actual behavior. This may further lead to incorrect predictions, such as recommending the mall to User 1 or the office to User 2. **Second**, most methods rely on stochastic noise perturbations during the generation process. However, our empirical results show that diffusion processes based on random noise perform suboptimally on the next location prediction task. We conjecture that this is because the uncertainty introduced by random noise hinders the evolution of user preferences along a stable, controllable trajectory and may disrupt the structural consistency of personalized spatiotemporal preferences. Detailed experimental evidence is provided in Section 2.

To tackle these knotty challenges, we propose **IDDiff**, an Intent-aware Deterministic Diffusion framework for next location prediction. We develop a Sequence-aware Trajectory Representation Module. Specifically, we first fusion spatio-temrporal factors to capture the user's global trajectory-level intent representation. Meanwhile, we apply a sliding-window strategy to the user's historical trajectory and construct multiple local subtrajectories. Each subtrajectory is then sequentially encoded to obtain its corresponding local intent representation, after which a clustering method is employed to extract representative intent prototypes. Subsequently, we select the intent prototype that is most semantically similar to the global intent representation to guide the diffusion model. IDDiff introduces a location Distance Graph Encoder to model the complex geographical relationships among locations. In addition, we design an Intent Refinement Module. intent structure-preserving diffusion model. This module utilizes the pooled vector of user trajectories as the diffusion terminus, thereby retaining information consistent with historical visit patterns. Simultaneously, it leverages user intent prototypes for conditional guidance, ensuring that the generated representations remain insulated from the interference of irrelevant intents.

Our contributions. We design an Sequence-aware Trajectory Representation Module that extracts trajectory intent prototypes to serve as intent-guided conditions for the diffusion model while capturing user global historical intent preferences. Our experiments reveal a limitation in diffusion models based on stochastic noise when applied to the next location prediction task. Consequently, we design an Intent Refinement Module that employs a deterministic diffusion terminus to preserve information consistent with historical user visit patterns. Comprehensive experiments conducted on five real-world datasets demonstrate that IDDiff consistently outperforms state-of-the-art location prediction models across all evaluation metrics. Specifically, IDDiff improves by 3.6%, 3.5%, 8.3%, 5.7%, and 7.9% over the strongest baseline in Acc@1, respectively.

2 Rethinking Stochastic Diffusion Model in Next Location Prediction

This section examines the limitations of diffusion models based on stochastic Gaussian perturbation when applied to the next location prediction task.

Datasets. We evaluate IDDiff on five real-world city-scale Foursquare check-in datasets, including Istanbul (IST), Jakarta (JK), Sao Paulo (SP), New York City (NYC), and Los Angeles (LA). [1]Each check-in record contains a user ID, a location ID, a timestamp, and the geographical coordinates of the location. Among them, JK, SP, and NYC are widely used benchmarks for next location prediction, while IST and LA are extracted from a global Foursquare dataset according to latitude and longitude ranges. All datasets cover about 22 months, from April 2012 to January 2014. For each dataset, check-in records are sorted chronologically and split into training, validation, and test sets with a ratio of 8:1:1. This setting follows the next location prediction scenario, where historical trajectories are used to predict future destinations. The statistics are shown in Table 1, including the numbers of users, locations, check-ins, graph edges, and dataset density.

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

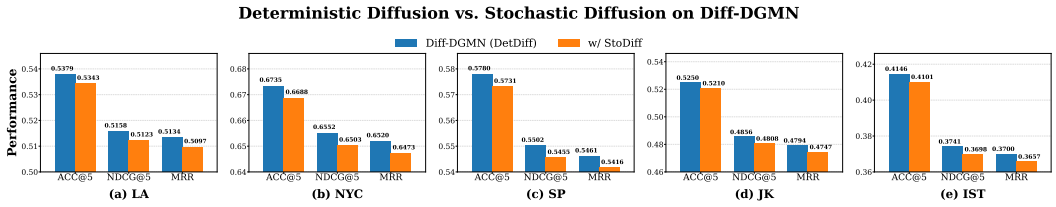


Figure 2: Comparison between deterministic diffusion and stochastic diffusion on Diff-DGMN across five datasets.

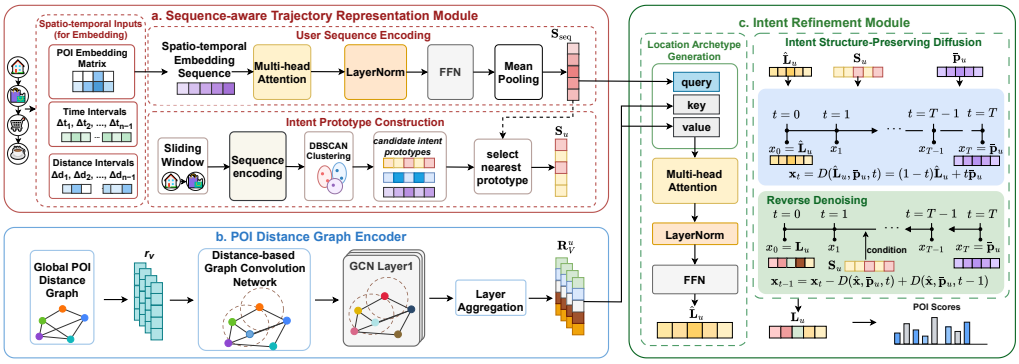


Figure 3: The overall architecture of IDDiff.

Experimental Setup. We adopt Diff-DGMN as the backbone and replace its stochastic diffusion process with deterministic diffusion (Bansal et al., 2023), while keeping all other components and training settings unchanged. Instead of injecting random perturbations into user preferences, the modified model constructs a predefined and controllable transformation trajectory.

Experimental Analysis. As shown in Figure 2, the deterministic variant consistently outperforms the original stochastic Diff-DGMN in terms of Acc@5, NDCG@5, and MRR across all five datasets. Since the backbone architecture and all other components remain unchanged throughout the comparison, these performance improvements can be primarily attributed to replacing stochastic noising with a deterministic preference transformation process. These results lead to a important conclusions. Stochastic noise perturbation is not well suited to next location prediction, as the injected randomness may interfere with intent-related information in user preference representations and increase the uncertainty of preference reconstruction, thereby weakening the model's ability to accurately characterize users' latent visiting intentions.

3 Preliminaries

Next Location Prediction. Let $\mathcal{U} = \{u_1, u_2, \ldots, u_{|\mathcal{U}|}\}$ and $\mathcal{P} = \{p_1, p_2, \ldots, p_{|\mathcal{P}|}\}$ denote the sets of users and locations, respectively. Each location $p \in \mathcal{P}$ is associated with its geographical coordinates $(\mathrm{lon}_p, \mathrm{lat}_p)$. For a user $u \in \mathcal{U}$, the historical check-in trajectory is represented as a time-ordered sequence $\mathcal{S} = \{(p_1^u, t_1^u), (p_2^u, t_2^u), \ldots, (p_n^u, t_n^u)\}$, where $p_i^u \in \mathcal{P}$ and t_i^u denote the location visited by user u and its corresponding timestamp at the i-th check-in. Given $\mathcal{S}_u$, the next location prediction task aims to predict the next visited location p_{n+1}^u by ranking all candidate locations in $\mathcal{P}$ and returning the Top-k most likely locations.

4 Method

In this section, we provide a detailed description of the proposed IDDiff model, whose overall architecture is illustrated in Figure 3. IDDiff consists of three core components: (1) the Sequence-aware Trajectory Representation Module, which extracts representative intent prototypes from users' historical trajectories; (2) the location Distance Graph Encoder, which captures geographical correlations among locations; and (3) the Intent Refinement Module, which generates users' latent intended location preferences.

4.1 SEQUENCE-AWARE TRAJECTORY REPRESENTATION MODULE

In this section, we investigate the temporal discontinuity of visit behaviors driven by the same underlying intent in user mobility trajectories. In designing the Sequence-aware Trajectory Representation Module, we consider user trajectories enriched with spatial and temporal factors from two complementary perspectives to address the entanglement of multiple intents. First, we employ User Sequence Encoding to comprehensively model the intent-related information embedded in users' historical trajectories, thereby capturing their diverse mobility patterns. Second, we introduce an Intent Prototype Construction Module to identify representative intent prototypes and filter out interference from irrelevant intents, thereby enabling the global trajectory representation to more accurately reflect the user's dominant mobility intent.

Spatio-Temporal Encoding. Why is it important to incorporate spatial and temporal factors? A short temporal interval or a geographically close transition indicates strong behavioral correlations between check-ins, while a long interval may imply a shift in user intent. The spatial and temporal interval indices are mapped into trainable embeddings. For each visited location p_i, the input representation is defined as

$$\mathbf{x}_i = \mathbf{e}_i^p + \mathbf{e}_i^s + \mathbf{e}_i^t,$$

where $\mathbf{e}_i^p$, $\mathbf{e}_i^s$, and $\mathbf{e}_i^t$ denote the location, spatial interval, and temporal interval embeddings, respectively. Let $\mathbf{X}_u = [\mathbf{x}_1, \ldots, \mathbf{x}_{n_u}] \in \mathbb{R}^{n \times d}$ denote the user check-in sequence.

User Sequence Encoding . Given the spatio-temporal enhanced sequence representation, we employ an attention-based encoder to capture the user's global intent representation. Specifically, multi-head self-attention is applied as:

$$\begin{cases} \text{head}_h = \text{Attention}\left(\mathbf{X}_u\mathbf{W}_Q^h, \mathbf{X}_u\mathbf{W}_K^h, \mathbf{X}_u\mathbf{W}_V^h\right), \\ \text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Softmax}\left(\dfrac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}}\right)\mathbf{V}, \\ \mathbf{H}_{\text{Att}} = \text{Concat}\left(\text{head}_1, \ldots, \text{head}_h\right)\mathbf{W}_O. \end{cases} \tag{1}$$

where $\mathbf{W}_Q^h$, $\mathbf{W}_K^h$, $\mathbf{W}_V^h$, and $\mathbf{W}_O$ are trainable matrices. The encoded features are further transformed by a feed-forward network, and a padding mask is applied to remove invalid positions. Finally, masked mean pooling over valid check-ins produces the trajectory representation $\mathbf{S}_{\text{seq}}$, which captures user mobility intent with sequential and spatio-temporal information.

Intent Prototype Construction. We construct intent prototypes from local subsequence representations. Specifically, each valid check-in sequence is divided into multiple local subsequences using a sliding window strategy, where each subsequence captures a short-term mobility pattern and can be regarded as a potential carrier of a specific user intent. If the trajectory is shorter than the window size, the whole valid trajectory is used as one subsequence. Given a window size w, the j-th local subsequence is constructed as

$$\mathcal{S}^{(j)} = \begin{cases} [\mathbf{x}_j, \mathbf{x}_{j+1}, \ldots, \mathbf{x}_{j+w-1}], & \substack{1 \le j \le n_u - w + 1, \\ n_u \ge w,} \\ [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{n_u}], & \substack{j=1, \\ n_u < w.} \end{cases} \tag{2}$$

Why is a Sliding-Window Strategy necessary? Directly modeling an entire user trajectory as a single unified representation may entangle multiple visiting intents, particularly when semantically related behaviors occur at temporally non-contiguous positions.

Each subsequence is then encoded by the same attention-based sequence encoder used in User Sequence Encoding, and mean pooling is applied to obtain its subsequence-level intent representation. After that, DBSCAN Ester et al. (1996) is used to cluster these representations and discover recurrent intent patterns without predefining the number of latent intents. For each valid cluster, the mean representation of its assigned subsequences is used as the corresponding intent prototype. These prototypes represent the user's dominant local mobility patterns extracted from historical behaviors. Finally, we compare the whole-trajectory representation with all constructed intent prototypes and select the closest one as the user's dominant intent prototype:

$$\mathbf{S}_u = \arg\min_{\boldsymbol{\mu}_k \in \mathcal{C}_u} \left\| \mathbf{S}_{\text{seq}} - \boldsymbol{\mu}_k \right\|^2, \tag{3}$$

where $\mathcal{C}_u = \{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_{K_u}\}$ denotes the set of intent prototypes constructed for user u, $\boldsymbol{\mu}_k$ represents the k-th intent prototype obtained as the centroid of the corresponding subsequence cluster, and K_u is the total number of intent prototypes constructed for user u. $\mathbf{S}_u$ denotes the selected prototype closest to the whole-trajectory representation. The resulting $\mathbf{S}_u$ is subsequently used as a guidance signal for the reverse process of the Intent Structure-Preserving Diffusion component in the Intent Refinement Module. In this way, the generated representation is guided toward the user's most relevant mobility intent while suppressing interference from irrelevant intents.

4.2 LOCATION DISTANCE GRAPH ENCODER

In this section, we focus on capturing geographical representations from objective spatial distances through graph structures. To capture geographical dependencies among locations, we construct an undirected spatial location graph $G_D = (\mathcal{P}, \mathcal{E}_D, \mathcal{A}_D)$, where $\mathcal{E}_D$ and $\mathcal{A}_D$ denote the edge set and weighted adjacency matrix, respectively. For two locations p_i and p_j, the edge $e^D_{i,j} = \langle p_i, p_j \rangle$ represents their geographical connection, and $a^D_{i,j}$ measures their distance-based closeness. A larger $a^D_{i,j}$ indicates stronger geographical correlation.

Based on the spatial location graph, we employ a distance-aware graph convolution operation to capture geographical dependencies among locations. At the k-th layer, the representation of location p_i is updated by aggregating its neighbors:

$$\mathbf{z}_i^{(k)} = \mathbf{z}_i^{(k-1)} + \sum_{j \in \mathcal{N}_i} \frac{a^D_{i,j}}{\sqrt{|\mathcal{N}_i| \cdot |\mathcal{N}_j|}} \mathbf{W}_C^{(k-2)} \mathbf{z}_j^{(k-2)}. \tag{4}$$

where $\mathcal{N}_i$ and $\mathcal{N}_j$ denote the neighbor sets of p_i and p_j, respectively, and $\mathbf{W}_C^{(k-2)}$ is a learnable transformation matrix. The distance-aware aggregation enables each location to incorporate geographical information from neighboring locations while preserving its previous representation.

To integrate the geographical information captured at different GCN layers, we further aggregate the representations of location p_i across all layers: $\mathbf{r} = \frac{1}{|\kappa|} \sum_{t=0}^{|\kappa|} \mathbf{z}_i^{(t)}$, where κ denotes the set of GCN layers and $\mathbf{z}_i^{(t)}$ represents location p_i at the t-th layer. This aggregation captures both original location semantics and multi-hop geographical dependencies. The representations of the locations are collected as $\mathbf{R}_v^u$ and fed into the Intent Refinement Module to extract personalized geographical intent.

4.3 INTENT REFINEMENT MODULE

Injecting random noise may introduce unnecessary perturbations, thereby disrupting the structure of users' personalized mobility intents. To overcome this limitation, we design an Intent Refinement Module to address the limitation that conventional stochastic diffusion models are not well-suited to the next location prediction task.

Location Archetype Generation. To obtain personalized spatial representations, we apply an attention-based extraction strategy. Specifically, $\mathbf{S}_{\text{seq}}$ serves as the query, while the visited location representations $\mathbf{R}_V^u$ are used as keys and values. A feed-forward network further transforms the aggregated representation to obtain the initial spatial representation $\hat{\mathbf{L}}_u$. Since $\hat{\mathbf{L}}_u$ may contain noisy or incomplete information from historical visits, we further refine it through the deterministic diffusion process.

Intent Structure-Preserving Diffusion. Conventional stochastic noising may disrupt the intent structure embedded in users' historical visiting behaviors and increase the difficulty of recovering personalized user characteristics during the reverse process. Therefore, we use the pooled representation of the user's trajectory as the deterministic terminal state of the diffusion process Zhao et al. (2024). Specifically, we compute a pooled sequence vector from the user's historical check-in representations: $\bar{\mathbf{p}}_u = \text{MeanPool}(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)$, where $\bar{\mathbf{p}}_u$ summarizes the average historical preference of the user.

In the proposed deterministic forward noising process, the location prototype $\hat{\mathbf{L}}_u$ is used as the starting locationnt. Following the implementation, the deterministic forward operator is defined as an interpolation function:

$$\mathbf{x}_t = D(\hat{\mathbf{L}}_u, \bar{\mathbf{p}}_u, t) = (1-t)\hat{\mathbf{L}}_u + t\bar{\mathbf{p}}_u, \quad t \in [0,1]. \tag{5}$$

where $t \in [0,1]$ denotes the normalized diffusion step, $\hat{\mathbf{L}}_u$ is the clean starting state, and $\bar{\mathbf{p}}_u$ is the deterministic terminal state. Deterministic forward progressively maps the representation to the pooled trajectory representation along a predefined path, ensuring that each intermediate state retains user-related global semantics. Consequently, the structural consistency of user mobility representations can be better preserved throughout the forward degradation and reverse restoration processes.

In the reverse stage, we employ a deterministic degradation process to progressively recover the target preference representation:

$$\widehat{\mathbf{x}} = R_\theta(\mathbf{x}_t \oplus \mathbf{S}_u), \tag{6}$$

where $R_\theta(\cdot)$ is an MLP, $\oplus$ denotes the concatenation operation, and $\mathbf{S}_u$ is used as the conditional guidance signal. $\mathbf{S}_u$ represents the intent prototype most relevant to the user's overall trajectory and therefore provides a more targeted semantic condition. By incorporating $\mathbf{S}_u$ as the guidance signal, the reverse process is steered toward the user's dominant mobility intent while suppressing interference from irrelevant or entangled intents. Then, the previous state is obtained by subtracting the degradation at the current step and adding back the forward at the previous step:

$$\mathbf{x}_{t-1} = \mathbf{x}_t - D\left(\widehat{\mathbf{x}}, \bar{\mathbf{p}}_u, t\right) + D\left(\widehat{\mathbf{x}}, \bar{\mathbf{p}}_u, t-1\right). \tag{7}$$

After iteratively applying the reverse update until $t = 0$, the final output is denoted as $\mathbf{L}_u = \mathbf{x}_0$, where $\mathbf{L}_u$ is the refined user preference representation generated by the deterministic diffusion process. Compared with the initial prototype $\hat{\mathbf{L}}_u$, the refined representation $\mathbf{L}_u$ better preserves personalized spatial preference, incorporates intent-guidance, and avoids the instability introduced by stochastic noising.

4.4 Model Optimization

For each candidate location $p_i \in \mathcal{P}$, we calculate its prediction score by measuring its relevance to both the intent-aware user representation and the diffusion-generated preference representation:

$$\hat{y}_i^u = \gamma \mathbf{S}_u \mathbf{r}_{p_i}^\top + (1-\gamma)\mathbf{L}_u \mathbf{r}_{p_i}^\top, \tag{8}$$

where $\mathbf{r}_{p_i} \in \mathbf{R}_V$ denotes the geography-aware representation of location p_i, and γ is a balance coefficient. For model optimization, we employ a cross-entropy loss for the prediction task, loss is formulated as $\mathcal{L}_{rec}$.

The diffusion loss is defined as $\mathcal{L}_{diff} = \mathbb{E}_t\left[\left\|\widehat{\mathbf{x}} - \hat{\mathbf{L}}_u\right\|_2^2\right]$, where $t \in (0,1)$ is uniformly sampled.

The overall training objective is formulated as $\mathcal{L}_{total} = \mathcal{L}_{rec} + \beta\mathcal{L}_{diff}$, where β controls the contribution of the diffusion loss.

5 Experiments

5.1 Experiment Settings

Data Set	IST	JK	SP	NYC	LA
#Users	9,208	5,358	3,722	1,083	965
#locations	11,871	10,706	12,829	9,989	2,541
#Check-ins	529,067	300,324	287,642	179,468	37,181
#Edges	2,593,873	1,315,879	845,298	1,447,008	67,956
Avg. Density	0.005	0.005	0.006	0.017	0.015

Table 1: Basic statistics of the five datasets.

In this section, we first introduce the experimental settings and then conduct extensive evaluations on five real-world datasets to answer the following five research questions: **RQ1:** How does IDDiff perform compared with state-of-the-art next location prediction methods? **RQ2:** How does each key component of IDDiff contribute to its overall prediction performance? **RQ3:** How sensitive is IDDiff to its key hyperparameters? **RQ4:** What is the computational complexity of IDDiff? **RQ5:** Does the proposed intent structure-preserving diffusion model outperform alternative diffusion methods and generalize across architectures? **RQ6:** Can the sequence-aware

Dataset	Model	Acc@1	Acc@5	Acc@10	Acc@20	N@5	N@10	N@20	MRR
LA	STAN	0.2899	0.3706	0.4033	0.4402	0.3322	0.3427	0.3521	0.3263
	PG2Net	0.4082	0.4671	0.4918	0.5209	0.4397	0.4477	0.4551	0.4359
	GTR-Mamba	0.4629	0.5075	0.5295	0.5651	0.4812	0.4966	0.5054	0.4916
	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
	STHGCN	0.4469	0.4922	0.5223	0.5549	0.4759	0.4841	0.4913	0.4729
	SLS-REC	0.3823	0.4509	0.4853	0.5212	0.4185	0.4296	0.4386	0.4147
	Diff-POI	0.4921	0.5333	0.5557	0.5770	0.5136	0.5208	0.5262	0.5128
	Diff-DGMN	0.4871	0.5343	0.5563	0.5798	0.5123	0.5195	0.5255	0.5097
	IDDiff	**0.5098**	**0.5495**	**0.5612**	**0.5901**	**0.5302**	**0.5304**	**0.5404**	**0.5297**
NYC	STAN	0.5355	0.5847	0.6045	0.6292	0.5616	0.5681	0.5742	0.5583
	PG2Net	0.5667	0.6096	0.6298	0.6514	0.5894	0.5959	0.6014	0.5869
	GTR-Mamba	0.6137	0.6459	0.6627	0.6728	0.6230	0.6388	0.6445	0.6329
	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
	STHGCN	0.5812	0.6238	0.6443	0.6641	0.6031	0.6098	0.6147	0.6004
	SLS-REC	0.5505	0.5952	0.6173	0.6382	0.5743	0.5815	0.5868	0.5718
	Diff-POI	0.6381	0.6746	0.6880	0.7037	0.6579	0.6623	0.6662	0.6559
	Diff-DGMN	0.6297	0.6688	0.6842	0.6992	0.6503	0.6553	0.6591	0.6473
	IDDiff	**0.6602**	**0.6941**	**0.6993**	**0.7104**	**0.6817**	**0.6823**	**0.6848**	**0.6801**
SP	STAN	0.3524	0.4309	0.4655	0.5028	0.3941	0.4051	0.4147	0.3891
	PG2Net	0.4274	0.5024	0.5338	0.5661	0.4674	0.4776	0.4857	0.4622
	GTR-Mamba	0.4839	0.5416	0.5680	0.5964	0.5097	0.5206	0.5318	0.5188
	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
	STHGCN	0.4721	0.5312	0.5561	0.5833	0.5035	0.5115	0.5184	0.4995
	SLS-REC	0.4170	0.4885	0.5200	0.5572	0.4551	0.4652	0.4746	0.4507
	Diff-POI	0.5169	0.5769	0.6021	0.6290	0.5488	0.5569	0.5637	0.5459
	Diff-DGMN	0.5146	0.5731	0.5981	0.6252	0.5455	0.5536	0.5605	0.5416
	IDDiff	**0.5597**	**0.6045**	**0.6217**	**0.6499**	**0.5814**	**0.5917**	**0.5927**	**0.5819**
JK	STAN	0.3269	0.4159	0.4554	0.4863	0.3637	0.3865	0.3968	0.3678
	PG2Net	0.4306	0.5126	0.5465	0.5800	0.4742	0.4851	0.4938	0.4678
	GTR-Mamba	0.4335	0.5168	0.5506	0.5855	0.4789	0.4874	0.4981	0.4709
	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
	STHGCN	0.4300	0.5121	0.5461	0.5715	0.4735	0.4846	0.4935	0.4670
	SLS-REC	0.4016	0.4977	0.5398	0.5771	0.4523	0.4660	0.4762	0.4458
	Diff-POI	0.4316	0.5201	0.5574	0.5944	0.4787	0.4908	0.5001	0.4742
	Diff-DGMN	0.4354	0.5210	0.5565	0.5938	0.4808	0.4923	0.5017	0.4747
	IDDiff	**0.4601**	**0.5404**	**0.5701**	**0.6097**	**0.5057**	**0.5149**	**0.5247**	**0.5018**
IST	STAN	0.2141	0.2996	0.3461	0.3974	0.2586	0.2736	0.2866	0.2548
	PG2Net	0.2854	0.3674	0.4105	0.4601	0.3283	0.3421	0.3546	0.3244
	GTR-Mamba	0.3201	0.4018	0.4354	0.4836	0.3552	0.3716	0.3829	0.3490
	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
	STHGCN	0.3052	0.3875	0.4293	0.4772	0.3486	0.3620	0.3741	0.3445
	SLS-REC	0.2680	0.3510	0.3936	0.4431	0.3101	0.3239	0.3363	0.3056
	Diff-POI	0.3327	0.4156	0.4568	0.5030	0.3761	0.3894	0.4010	0.3739
	Diff-DGMN	0.3251	0.4101	0.4541	0.5043	0.3698	0.3840	0.3966	0.3657
	IDDiff	**0.3589**	**0.4348**	**0.4714**	**0.5201**	**0.4007**	**0.4117**	**0.4201**	**0.3917**

Table 2: Performance comparison with baselines on five heterogeneous datasets. The bold values indicate the best performance, and the underlined values indicate the best baseline. All improvements are statistically significant, as determined by a paired t-test with $p \leq 0.05$

trajectory representation module capture discontinuous yet related behaviors and learn discriminative intent prototypes?[2]

Datasets. The dataset configuration follows the setup in the Section 2 titled *Rethinking Stochastic Diffusion Model in Next Location Prediction*.

Evaluation Metrics. We evaluate next location prediction using Accuracy (Acc@k), Mean Reciprocal Rank (MRR), and Normalized Discounted Cumulative Gain (NDCG@k), where $k = \{1, 5, 10, 20\}$. Higher values for all three metrics indicate better model performance.

Baseline Methods. To verify the effectiveness of the proposed IDDiff model, we compare it with nine representative next location prediction methods, including spatio- temporal self-attention models: STAN Luo et al. (2021), PG2Net Wang et al. (2024a), GTR-Mamba Li et al. (2025b); graph-based representation learning methods: DRAN Wang et al. (2022), GETNext Yang et al. (2022), hypergraph-based trajectory modeling ap- proaches: STHGCN Yan et al. (2023), SLS-REC Fu et al. (2024), and diffusion-based methods: Diff-POI Qin et al. (2023), and Diff-DGMN Zuo & Zhang (2024).

Hyper-parameters Settings. All experiments were conducted using an NVIDIA A6000 GPU with 48GB of memory in a PyTorch environment. The embedding dimension d was set to 128, and the model was optimized using Adam with a learning rate of 5×10^{-4}. The balance coefficient γ and

[2]**RQ4**, **RQ5** and **RQ6** are further investigated in the appendix A.1, A.2, and A.3, respectively.

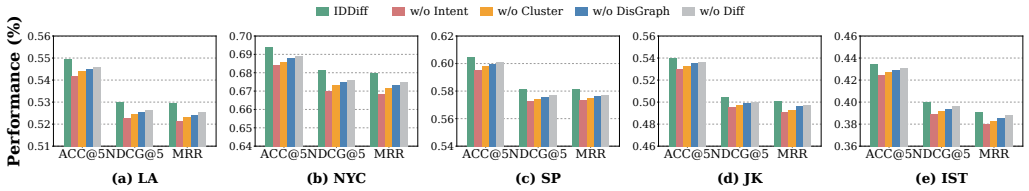


Figure 4: Ablation study results showing the performance of different variants across five heterogeneous datasets: (a) LA, (b) NYC, (c) SP, (d) JK, and (e) IST.

the diffusion-loss coefficient β were set to 0.75 and 0.2, respectively, while the dropout rate dp was set to 0.25. The sliding-window size w and the number of diffusion steps T were set to 4 and 100, respectively.

5.2 Performance Comparison With Baselines(RQ1)

Table 2 reports the performance comparison between IDDiff and nine baselines on five datasets. IDDiff achieves the best results across all datasets and metrics, improving Acc@1 over the strongest baseline by 3.6%, 3.5%, 8.3%, 5.7%, and 7.9% on LA, NYC, SP, JK, and IST, respectively. The consistent improvements across both dense and sparse datasets demonstrate the effectiveness and robustness of IDDiff.

Among the baselines, hypergraph-based methods generally outperform conventional graph-based models, indicating the importance of structured dependency modeling for capturing complex mobility patterns. Diffusion-based methods, such as Diff-POI and Diff-DGMN, also achieve competitive results, showing the potential of generative preference modeling for next location prediction. Compared with these baselines, IDDiff obtains more stable improvements mainly due to two designs. First, the Sequence-aware Trajectory Representation Module captures temporally discontinuous but semantically related user behaviors, reducing intent entanglement in trajectory modeling. Second, the deterministic diffusion process refines user preference representations without introducing random perturbations, thereby better preserving personalized mobility structures.

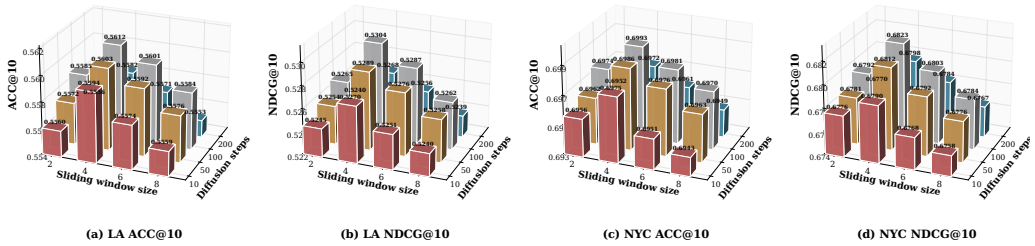
5.3 Ablation Study(RQ2)

To validate the contribution of each core component in IDDiff, we conduct an ablation study by removing or replacing several key modules. The complete IDDiff model is compared with the following variants: **w/o Intent**: removes the sequence-aware prototype construction module and directly uses the whole trajectory representation as the user intent representation. **w/o Cluster**: removes DBSCAN clustering and uses the mean representation of all local subtrajectories instead of representative intent prototypes. **w/o DisGraph**: removes the location distance graph encoder and directly uses trainable location embeddings without modeling geographical dependencies. **w/o Diff**: removes the Intent Refinement Module and directly uses the initial location prototype $\hat{\mathbf{L}}_u$ for prediction.

The ablation results are reported in Figure 4. Removing or replacing any key component leads to performance forward, indicating that each module contributes to the final prediction performance. In particular, **w/o Intent** shows a clear drop, indicating that directly using intent entangled raw trajectory representations to guide the diffusion model leads to learned behavior representations that deviate from users' true intents. The lower performance of **w/o Cluster** verifies the importance of clustering in discovering representative intent patterns from local mobility behaviors. Without clustering, averaging all local subtrajectory representations may introduce noisy or irrelevant signals. The forward of **w/o DisGraph** shows that geographical dependencies among locations are important, since nearby locations or locations in the same functional region often share similar mobility patterns. Finally, **w/o Diff** performs worse than IDDiff, indicating that deterministic diffusion effectively refines the coarse location prototype $\hat{\mathbf{L}}_u$ and generates a more informative spatial preference representation $\mathbf{L}_u$ for next location prediction.

5.4 Hyperparameter Analysis(RQ3)

As shown in Figure 5, we analyze the effects of the sliding window size w and diffusion steps T on the LA and NYC datasets, with $w \in \{2, 4, 6, 8\}$ and $T \in \{10, 50, 100, 200\}$. Model performance



(a) LA ACC@10 (b) LA NDCG@10 (c) NYC ACC@10 (d) NYC NDCG@10

Figure 5: Impact of the sliding window size w and the number of diffusion steps T on the LA and NYC datasets.

first improves and then declines as w increases. A small window provides insufficient context for intent modeling, while an overly large window may mix behaviors associated with different intents. Thus, $w = 4$ achieves the best results on both datasets. Increasing T from 10 to 100 consistently improves performance by enabling a smoother and more fine-grained preference reconstruction. However, further increasing T to 200 brings no additional benefit and slightly degrades performance due to redundant transformations and higher computational cost. Therefore, we set $w = 4$ and $T = 100$ as the default values.

6 Related Work

Next Location Prediction. Next location prediction aims to predict a user's next destination from historical check-in trajectories Liu et al. (2016); Feng et al. (2018); Luo et al. (2021). Recent studies have developed more expressive mobility representations beyond conventional sequential modeling Yang et al. (2022); Yan et al. (2023); Lai et al. (2024); Song et al. (2025); Qin et al. (2025). More recent advances further explore geographical state space models, generative learning, and scenario-aware hypergraphs Qin et al. (2025); Lin et al. (2026); Wang et al. (2024b); Pan et al. (2025). For instance, DMSDRec Li et al. (2025a) reconstructs robust preferences from incomplete trajectories through graph masking and spatio-temporal diffusion. Nevertheless, these methods do not explicitly aggregate semantically related but temporally discontinuous behaviors into unified intent representations, making latent intents difficult to distinguish in intent-mixed trajectories.

Diffusion Model. Diffusion models have recently been applied to prediction systems to learn robust preference representations through iterative corruption and denoising. Li et al. (2023); Liu et al. (2023); Zhao et al. (2024) Recently, diffusion-based models have emerged as a promising approach to addressing challenges in recommender systems. DiffuASR Liu et al. (2023) employs diffusion models to generate item sequences. More recently, Cold Diffusion Bansal et al. (2023) extends the forward process to deterministic degradations such as blurring and masking. For location prediction, Diff-POI Qin et al. (2023) combines sequential and spatial graph encoders with diffusion sampling to model geographical visiting patterns. However, directly using a user's whole-trajectory intent to guide diffusion may introduce irrelevant intents, thereby interfering with target preference generation and reducing intent modeling accuracy.

7 Conclusion

In this work, we propose an intent-aware deterministic diffusion framework for next location prediction. We design a sequence-aware trajectory representation module that captures a user's global historical intent representation while handling temporally discontinuous visits and extracting trajectory intent prototypes. Consequently, we introduce a novel intent refinement module that leverages intent prototypes to guide an intent structure-preserving diffusion model, effectively resolving structural disruption and reconstruction uncertainty caused by random noise. Extensive experiments across five real-world datasets demonstrate that IDDiff consistently outperforms state-of-the-art methods and validate the effectiveness of each key component.

REFERENCES

Arpit Bansal, Eitan Borgnia, Hong-Min Chu, Jie Li, Hamid Kazemi, Furong Huang, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Cold diffusion: Inverting arbitrary image transforms without noise. *Advances in Neural Information Processing Systems*, 36:41259–41282, 2023.

Jiajie Chen, Yu Sang, Peng-Fei Zhang, Jiaan Wang, Jianfeng Qu, and Zhixu Li. Enhancing long- and short-term representations for next poi recommendations via frequency and hierarchical contrastive learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 39, pp. 11472–11480, 2025.

Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, pp. 226–231, 1996.

Jie Feng, Yong Li, Chao Zhang, Funing Sun, Fanchao Meng, Ang Guo, and Depeng Jin. Deepmove: Predicting human mobility with attentional recurrent networks. In *Proceedings of the 2018 world wide web conference*, pp. 1459–1468, 2018.

Jiarun Fu, Rong Gao, Yonghong Yu, Jia Wu, Jing Li, Donghua Liu, and Zhiwei Ye. Contrastive graph learning long and short-term interests for poi recommendation. *Expert Systems with Applications*, 238:121931, 2024.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Yantong Lai, Yijun Su, Lingwei Wei, Tianqi He, Haitao Wang, Gaode Chen, Daren Zha, Qiang Liu, and Xingxing Wang. Disentangled contrastive hypergraph learning for next poi recommendation. In *Proceedings of the 47th international ACM SIGIR conference on research and development in information retrieval*, pp. 1452–1462, 2024.

Pengxiang Lan, Enneng Yang, Yuliang Liang, Jianzhe Zhao, Guibing Guo, and Hai Zhao. Multi-perspective driven expected location preferences for next poi recommendations. In *Proceedings of the 49th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 869–880, 2026.

Yue Li, Jun Zeng, Haoran Tang, Junhao Wen, Min Gao, and Wei Zhou. Dmsdrec: Dynamic structure-aware graph masked autoencoder and spatiotemporal diffusion for next-poi recommendation. *IEEE Transactions on Services Computing*, 18(4):2024–2037, 2025a.

Zhuoxuan Li, Jieyuan Pei, Tangwei Ye, Zhongyuan Lai, Zihan Liu, Fengyuan Xu, Qi Zhang, and Liang Hu. Gtr-mamba: Geometry-to-tangent routing for hyperbolic poi recommendation. *arXiv e-prints*, pp. arXiv–2510, 2025b.

Zihao Li, Aixin Sun, and Chenliang Li. Diffurec: A diffusion model for sequential recommendation. *ACM Transactions on Information Systems*, 42(3):1–28, 2023.

Yuxi Lin, Yongkang Li, Jie Xing, and Zipei Fan. Multifaceted scenario-aware hypergraph learning for next poi recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 15269–15277, 2026.

Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. Predicting the next location: A recurrent model with spatial and temporal contexts. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30, 2016.

Qidong Liu, Fan Yan, Xiangyu Zhao, Zhaocheng Du, Huifeng Guo, Ruiming Tang, and Feng Tian. Diffusion augmentation for sequential recommendation. In *Proceedings of the 32nd ACM International conference on information and knowledge management*, pp. 1576–1586, 2023.

Yingtao Luo, Qiang Liu, and Zhaocheng Liu. Stan: Spatio-temporal attention network for next location recommendation. In *Proceedings of the web conference 2021*, pp. 2177–2185, 2021.

Yinchen Pan, Jun Zeng, Ziwei Wang, Haoran Tang, Junhao Wen, and Min Gao. Hgdrec: Next poi recommendation based on hypergraph neural network and diffusion model. *IEEE Transactions on Services Computing*, 18(3):1445–1458, 2025.

Yifang Qin, Hongjun Wu, Wei Ju, Xiao Luo, and Ming Zhang. A diffusion model for poi recommendation. *ACM Transactions on Information Systems*, 42(2):1–27, 2023.

Yifang Qin, Jiaxuan Xie, Zhiping Xiao, and Ming Zhang. Geomamba: Towards multi-granular poi recommendation with geographical state space model. In *Proceedings of the AAAI conference on artificial intelligence*, volume 39, pp. 12479–12487, 2025.

Chao Song, Zheng Ren, and Li Lu. Integrating personalized spatio-temporal clustering for next poi recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 12550–12558, 2025.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Bin Wang, Huifeng Li, Weipeng Wang, Menghan Wang, Yaohui Jin, and Yanyan Xu. Pg^2 net: Personalized and group preferences guided network for next place prediction. *IEEE Transactions on Intelligent Transportation Systems*, 25(8):8655–8670, 2024a.

Zhaobo Wang, Yanmin Zhu, Haobing Liu, and Chunyang Wang. Learning graph-based disentangled representations for next poi recommendation. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval*, pp. 1154–1163, 2022.

Ziwei Wang, Jun Zeng, Lin Zhong, Ling Liu, Min Gao, and Junhao Wen. Dsdrec: Next poi recommendation using deep semantic extraction and diffusion model. *Information Sciences*, 678: 121004, 2024b.

Xiaodong Yan, Tengwei Song, Yifeng Jiao, Jianshan He, Jiaotuan Wang, Ruopeng Li, and Wei Chu. Spatio-temporal hypergraph learning for next poi recommendation. In *Proceedings of the 46th international ACM SIGIR conference on research and development in information retrieval*, pp. 403–412, 2023.

Song Yang, Jiamou Liu, and Kaiqi Zhao. Getnext: Trajectory flow map enhanced transformer for next poi recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on research and development in information retrieval*, pp. 1144–1153, 2022.

Jujia Zhao, Wang Wenjie, Yiyan Xu, Teng Sun, Fuli Feng, and Tat-Seng Chua. Denoising diffusion recommender model. In *Proceedings of the 47th international ACM SIGIR conference on research and development in information retrieval*, pp. 1370–1379, 2024.

Jiankai Zuo and Yaying Zhang. Diff-dgmn: A diffusion-based dual graph multiattention network for poi recommendation. *IEEE Internet of Things Journal*, 11(23):38393–38409, 2024.

A Appendices

A.1 Complexity Analysis (RQ4)

We analyze the computational complexity of IDDiff. Let B denote the batch size, n denote the maximum trajectory length, d denote the embedding dimension, and $|\mathcal{P}|$ denote the number of locations. Moreover, $|\mathcal{E}_D|$ denotes the number of edges in the global location distance graph, L denotes the number of graph convolution layers, and T denotes the number of deterministic reverse diffusion steps.

For the User Preference Extraction Module, the whole trajectory is encoded using multi-head self-attention, whose dominant computational complexity is $\mathcal{O}(Bn^2d)$. During training, the trajectory is further divided into local sub-trajectories using a fixed-size sliding window. Since the window size is a small constant, the sub-trajectory encoding introduces a linear cost with respect to n. In the worst case, DBSCAN requires pairwise comparisons among the local sub-trajectory representations, resulting in $\mathcal{O}(Bn^2d)$. Therefore, the overall complexity of the intent-aware preference extraction process remains

$$\mathcal{O}(Bn^2d). \tag{9}$$

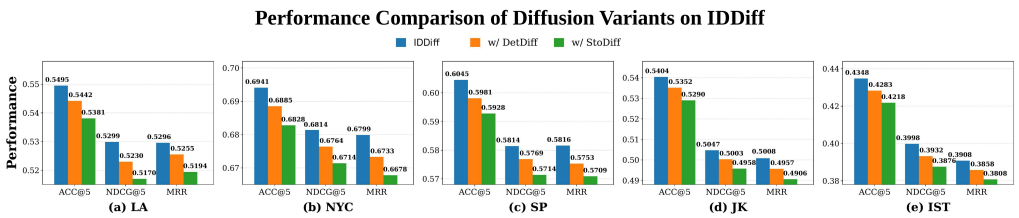


Figure 6: Comparison between deterministic diffusion and stochastic diffusion on five datasets.

For the Geographic Context Propagation Module, each graph convolution layer performs feature transformation over all locations and message propagation along the edges of the global location distance graph. Its complexity is

$$\mathcal{O}\left(L\left(|\mathcal{P}|d^2 + |\mathcal{E}_D|d\right)\right). \tag{10}$$

Since the global geographical representations are shared by all users and remain unchanged during inference, they can be precomputed once and cached for subsequent predictions. Therefore, this graph encoding cost does not need to be repeatedly incurred for every user during online inference.

For the Deterministic Diffusion Preference Generation Module, each reverse step applies a lightweight score network to a d-dimensional preference representation, leading to a complexity of $\mathcal{O}(Bd^2)$. Accordingly, the T-step deterministic generation process requires

$$\mathcal{O}(BTd^2). \tag{11}$$

Unlike stochastic diffusion methods that commonly require multiple samples, our deterministic generation process produces a stable preference representation through a single reverse trajectory. Finally, calculating the prediction scores over all candidate locations requires $\mathcal{O}(B|\mathcal{P}|d)$.

During online inference, the global geographical representations can be cached and reused. Therefore, the user-dependent inference complexity is

$$\mathcal{O}\left(B\left(n^2d + Td^2 + |\mathcal{P}|d\right)\right). \tag{12}$$

This complexity mainly arises from trajectory self-attention, deterministic reverse generation, and candidate location scoring, while avoiding repeated computation over the global location distance graph.

A.2 Analysis of Intent Structure-Preserving Diffusion (RQ5)

To further evaluate the effectiveness of the proposed deterministic diffusion mechanism, we compare IDDiff with two diffusion variants, namely **w/ DetDiff** and **w/ StoDiff**. The **w/ StoDiff** variant adopts conventional stochastic Gaussian diffusion, in which Gaussian noise is progressively injected into the user preference representation during the forward process and removed through multi-step reverse denoising. In contrast, **w/ DetDiff** replaces stochastic Gaussian perturbation with a deterministic diffusion process (Bansal et al., 2023). However, it still deterministically degrades the original preference representation toward a noisy terminal state, rather than explicitly specifying a semantically meaningful endlocationnt. IDDiff further constructs a deterministic interpolation trajectory between the user-specific location prototype $\hat{\mathbf{L}}_u$ and the pooled sequence representation $\bar{\mathbf{p}}_u$, thereby avoiding both stochastic perturbations and an uninformative noise-oriented terminal state.

As shown in Figure 6, the overall performance consistently follows the order IDDiff $>$ **w/ DetDiff** $>$ **w/ StoDiff** in terms of ACC@5, NDCG@5, and MRR across all five datasets. In particular, **w/ DetDiff** consistently outperforms **w/ StoDiff**, demonstrating that removing stochastic Gaussian sampling can reduce the uncertainty introduced during preference corruption and reconstruction. Nevertheless, IDDiff achieves further improvements over **w/ DetDiff** on LA, NYC, SP, JK, and IST. This indicates that merely replacing stochastic perturbation with deterministic noising is insufficient, because transforming the preference representation toward a noise state may still weaken information related to users' latent visiting intentions.

The advantage of IDDiff mainly stems from its stable and semantically controlled preference transformation process. Unlike **w/ DetDiff**, which only deterministically converts the original representation into a noisy state, IDDiff defines the pooled sequence representation $\bar{\mathbf{p}}_u$ as the terminal

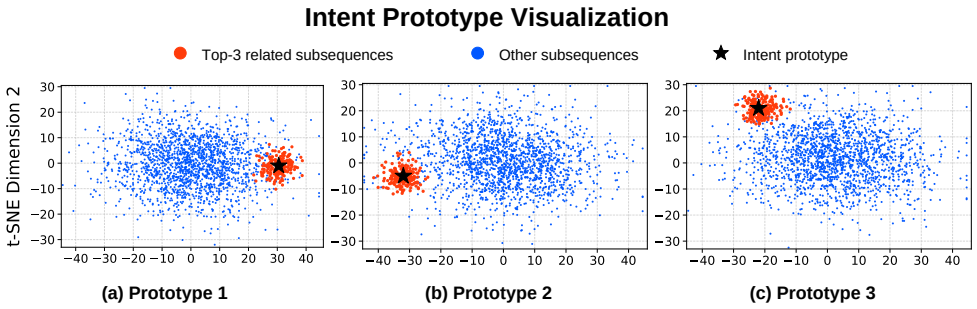


Figure 7: Visualization of intent prototype representations.

locationnt of the forward trajectory. Consequently, each intermediate state is constructed along a predefined interpolation path from the user-specific location prototype $\hat{\mathbf{L}}_u$ to $\bar{\mathbf{p}}_u$, rather than toward unstructured noise. During the reverse process, the degraded representation is progressively restored under the guidance of the intent-aware user representation $\mathbf{S}_u$. Therefore, **w/ DetDiff** verifies the advantage of deterministic diffusion over stochastic diffusion, while the further improvement of IDDiff demonstrates the importance of introducing a meaningful and user-related terminal representation. Overall, these results show that the proposed diffusion design can preserve intent-related preference information more effectively and generate more accurate user preference representations for next location prediction.

A.3 Intent Prototype Visualization (RQ6)

To further investigate whether the proposed intent-aware prototype construction method can capture distinguishable mobility intents, we visualize the local sub-trajectory representations learned by the model for different users.

We conduct the visualization experiment on the NYC dataset. First, the check-in records of each user are chronologically ordered according to their timestamps. To ensure that the selected trajectories can sufficiently reflect complex and temporally discontinuous mobility intents, we prioritize users with high location-category diversity, a relatively balanced distribution of recurrent behaviors, and frequent occurrences of identical or similar visiting behaviors at non-adjacent temporal positions. In addition, the historical trajectories of the candidate users are required to be sufficiently long to generate multiple local sub-trajectories and form valid intent clusters. Based on these criteria, we select three users with IDs 185, 730, and 529 as Prototype 1, Prototype 2, and Prototype 3, respectively, for visualization.

For each selected user, we employ a sliding window of size $w = 4$ to divide the historical trajectory into multiple local sub-trajectories. The local sub-trajectories are then encoded using the same trained sequence encoder as that employed in the main model, and their representations are clustered using the same DBSCAN configuration. Meanwhile, the complete historical trajectory of each user is encoded to obtain the corresponding global trajectory intent vector. For each valid cluster, the representations of its local sub-trajectories are aggregated through mean pooling to obtain the corresponding intent prototype. We then calculate the similarity between each intent prototype and the global trajectory intent vector and select the three most similar intent prototypes for each user. Finally, t-SNE is applied to jointly project the local sub-trajectory representations and the corresponding global trajectory intent vector into a two-dimensional space.

Figure 7 presents the visualization results for the three representative users. The blue locationnts denote the local sub-trajectories associated with the three intent prototypes most similar to the corresponding global trajectory intent vector, while the purple locationnts denote unrelated local sub-trajectories. The star marker represents the global trajectory intent vector of the corresponding user. We observe that the blue sub-trajectories form relatively compact and distinguishable regions around the global trajectory intent vector, whereas the purple sub-trajectories are generally distributed farther away. This finding indicates that, even when semantically related local visiting behaviors are

temporally scattered, the model can still aggregate them into a consistent latent intent space. These results demonstrate that the learned intent prototypes can effectively capture meaningful latent mobility intents and verify the effectiveness of the proposed intent-aware prototype construction method in alleviating the intent entanglement caused by directly encoding the complete trajectory.