LLM-Driven Multi-Perspective Location Completion for Next Location Prediction

Pengxiang Lan[1], Enneng Yang[2], Yuliang Liang[2], Jianzhe Zhao[2], Linying Jiang[2], Guibing Guo[2]*
[1]School of Computer Science and Engineering, Northeastern University, China
[2]Software College, Northeastern University, China,
{pengxianglan, ennengyang, liangyuliang}@stumail.neu.edu.cn,
{zhaojz, jiangly, guogb}@swc.neu.edu.cn

Abstract

Next location prediction aims to infer the next location users are likely to visit based on their historical check-in data. However, existing methods assume that check-in data is complete, overlooking the subjective nature of users' check-in behavior, leading to inaccurate capture of user preferences. Recently, Large Language Models (LLMs) have offered a promising approach to location completion due to their extensive world knowledge. Nevertheless, our experiments reveal that LLMs struggle to interpret raw geographic coordinate information. To address these challenges, we propose **LaMDA**, an **L**LM-driven **M**ulti-perspective **D**ata **A**ugmentation framework that employs dual completion agents to complement user mobility behaviors. Driven by our empirical findings that natural language descriptions align more closely with real-world geographic logic, LaMDA translates geographic coordinates into text to enhance spatial reasoning. Leveraging these semantic descriptions, LaMDA constructs dual agents to complement user mobility: "Micro-Level Completion" fills short-term omissions, while "Macro-Level Completion" infers unrecorded locations based on periodic preferences. Reliability is ensured through tailored real-world point-of-interest (POI) pools and a self-verification mechanism. Finally, a collaborative dual-graph module leverages this augmented data for fine-grained preference modeling. Extensive experiments on three real-world datasets demonstrate that LaMDA significantly outperforms state-of-the-art methods[1].

1 Introduction

The exponential growth of intelligent urban infrastructure has catalyzed unprecedented demand for individualized location-aware services across multiple sectors, including tourism, social platforms, and daily activities (Lan et al., 2023; Qin et al., 2023b; Zhuang et al., 2024; Huang et al., 2024c). As a core component, next location prediction leverages historical mobility patterns to anticipate users' future destinations, serving as a critical driver for both commercial recommendation and urban planning (Wang et al., 2024; Wan et al., 2025).

Existing approaches predominantly employ deep learning architectures, such as RNNs (Li et al., 2025), GNNs (Wang et al., 2022b), and Attention Mechanisms (Luo et al., 2021), to model numerical user interaction data with spatio-temporal factors. More recently, large language models (LLMs) have emerged as a promising paradigm for next location prediction, owing to their massive world knowledge and strong semantic understanding capabilities (Bao et al., 2023; Huang et al., 2024b; Lan et al., 2025b,a; Zhang et al., 2025). This paradigm shift enables LLM-based models to transcend conventional collaborative filtering (CF) methods by leveraging textual user interactions to comprehensively understand and precisely capture users' complex, multidimensional preferences (Feng et al., 2024; Long et al., 2024; Li et al., 2024a; Wang et al., 2024; Cheng et al., 2025).

Despite the effectiveness of previous methods that utilize various complex structures, they still face critical challenges. **Firstly**, users' check-in behavior represents a subjective record of visit patterns (users can choose whether to check in, which specific locations to record, or opt not to check in at all). For instance, in real-world scenarios, check-in behaviors are easily overlooked due to activity continuity, as users' focus tends to be on the intrinsic appeal of the location rather than on documenting their visit. As shown in Figure 1(a), in commercial districts, users may visit multiple storefronts or service facilities within a short period, such as a tourist who visits a souvenir shop immediately after touring a museum. However, existing methods are built on the assumption that

*Corresponding author.
[1]Our code is available at https://github.com/LanPangxiang/LaMDA2026.

Findings of the Association for Computational Linguistics: ACL 2026, pages 5406–5428

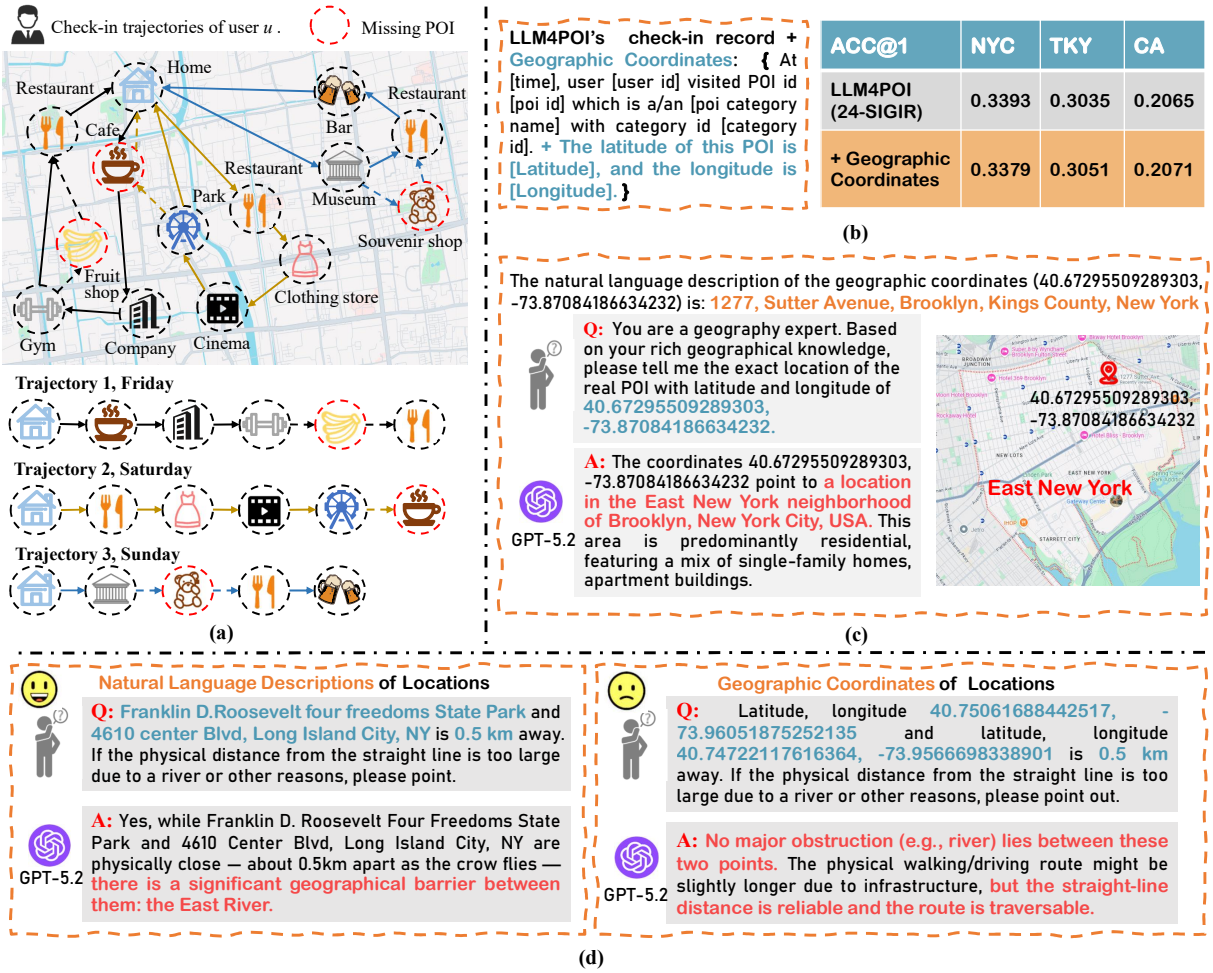

Figure 1: (a) A simple example where users may have visited a location without check-in. These locations may have appeared in other check-in trajectories, such as in trajectory 2. (b) Enhancing LLM training by incorporating geographic coordinates. (c) Evaluating whether LLMs can identify specific locations based on geographic coordinates. (d) Investigating whether LLMs can detect geographical barriers between two locations based on text or geographic coordinates.

user check-in data is complete, which leads to a biased understanding of user behavior and inaccurate capture of user preferences. Current data augmentation techniques, such as masking (Duan et al., 2023; Zeng et al., 2021), reordering (Zhuang et al., 2024; Dang et al., 2024), have effectively addressed data sparsity and improved model performance, but these approaches remain primarily numerical and lack a comprehensive understanding of the complex relationships within global check-in data. While TAU (Zhuang et al., 2024) and MPTM (Zeng et al., 2021) have preliminary exploration of missing locations, their approaches face similar limitations. **Secondly**, LLM-based approaches have proven effective in understanding location semantic information (Li et al., 2024a; Wang et al., 2024), **but our experiments reveal that LLMs struggle to interpret raw geographic coordinates in Section 2.** Furthermore, directly using LLM-generated results for location prediction introduces hallucination issues, as LLMs may generate fictitious locations that do not exist in reality.

To address these challenges, we propose an **LLM**-driven **M**ulti-perspective **D**ata **A**ugmentation framework (**LaMDA**) to complement user check-in data for the next location prediction. Specifically, we first convert geographic coordinates into natural-language descriptions of locations, thereby better aligning the model's inferences with real-world scenarios. Subsequently, to account for both short- and long-term user preferences, LaMDA employs two complementary location completion agents: (1) a Micro-Level Trajectory Completion Agent that identifies locations users likely visited but did not check into, possibly due to continuous activities; (2) a Macro-Level Intention Completion Agent that

examines relationships between users' overall visit patterns (such as periodicity) to summarize historical preferences and discover missing locations. To ensure reliability, we construct tailored real-world POI pools for each agent, anchoring LLM reasoning while mitigating hallucinations. Meanwhile, we design a self-verification mechanism that enables LLMs to provide feedback on their identified missing locations, thereby further improving the reliability of the generated locations. Finally, we constructed a collaborative dual-graph deep module that effectively utilizes semantically enhanced supplementary data to predict users' next location.

Contributions. Our main contributions can be summarized as follows:

- Our experiments reveal that LLMs struggle to interpret raw geographic coordinates. Converting these coordinates into natural language descriptions enables the model to reason more effectively within real-world scenarios.
- We propose an LLM-driven multi-perspective location completion framework (LaMDA). It employs two customized LLM agents to enrich users' check-in histories by capturing both short-term and long-term preferences. To the best of our knowledge, this is the first work to leverage LLMs for exploring missing locations in next location prediction tasks.
- LaMDA constructs tailored POI pools for each completion agent and designs a self-verification mechanism to improve the reliability of completed data. Additionally, it incorporates a dual-graph deep module to model the complex intrinsic behavior relationships.
- Extensive evaluations across three real-world datasets and two LLM-based inference settings show that LaMDA consistently outperforms state-of-the-art location prediction approaches.

2 Rethinking Spatial Comprehension of LLMs

To better explore LLMs' ability to understand spatiotemporal information, we conducted a series of detailed experiments. As shown in Figure 1(b), incorporating detailed geographic coordinate descriptions for each location in LLM4POI (Li et al., 2024a) failed to improve performance, prompting us to question: **Can LLMs properly understand geographic coordinate information?** Our subsequent investigations into LLMs' spatial reasoning capabilities revealed that **(i)** LLMs cannot precisely locate specific geographic coordinates, offering only approximate locations; **(ii)** LLMs can infer whether the travel time between two locations is consistent with their actual physical distance (more aligned with real-world scenarios) when provided natural language descriptions of locations, but they struggle to make the same judgment when the input is limited to raw geographic coordinates. Figures 1(c) and 1(d) demonstrate the validation of ChatGPT-5.2's spatial reasoning capabilities. The evaluation of spatial understanding across other advanced LLMs is presented in Appendix C.2.

3 Preliminaries

3.1 Next Location Prediction

In addressing the next location prediction problem, we define the sets of users and visited location points as $\mathcal{U} = \{u_1, u_2, \ldots, u_{|\mathcal{U}|}\}$ and $\mathcal{P} = \{p_1, p_2, \ldots, p_{|\mathcal{P}|}\}$, respectively. Each location point corresponds to a discrete real-world venue (equivalent to POI) identified by $p_i \in \mathcal{P}$ is characterized by a geographical coordinate pair and category $p_i = (lon_{p_i}, lat_{p_i}, cat_{p_i})$. For clarity and precision, we explicitly use the term POI in the Section 4 to trigger the LLMs' domain knowledge about specific venues and their functions.

Definition 1 (User Trajectory Sequence). The complete set of trajectory sequences for user $u \in \mathcal{U}$ is represented by $T_u = \{\mathcal{S}_u^1, \mathcal{S}_u^2, ..., \mathcal{S}_u^n\}$. Each trajectory sequence of the user u is denoted by $\mathcal{S}_u^i = \left\{(p_1^u, t_1^u), (p_2^u, t_2^u), \ldots, (p_{|\mathcal{S}_u^i|}^u, t_{|\mathcal{S}_u^i|}^u)\right\}$, where each tuple $(p_m^u, t_m^u), m \leq |\mathcal{S}_u^i|$ indicates that user u visited location p_m^u at timestamp t_m^u.

Definition 2 (Next Location Prediction). The objective of next location prediction is to predict the subsequent location p_{n+1}^u that user u will visit at time t_{n+1}^u, leveraging the user's historical trajectory sequences T_u.

3.2 Dual Graph Representations

To model both the sequential patterns and spatial relationships inherent in users' check-in trajectories, we construct dual location graphs.

Definition 3 (Behavior Transition Location Graph). To model the sequential patterns in users' check-in trajectories, for each user u, we construct a directed weighted graph $\mathcal{G}_u = (\mathcal{V}_u, \mathcal{E}_u)$, where $\mathcal{V}_u$ consists of locations visited by u (i.e., $\mathcal{V}_u \subseteq \mathcal{P}$), and each directed edge $e_{i \to j}^u = \langle p_i^u, p_j^u \rangle \in \mathcal{E}u$ indicates a transition from p_i^u to p_j^u in the user u's trajectory.

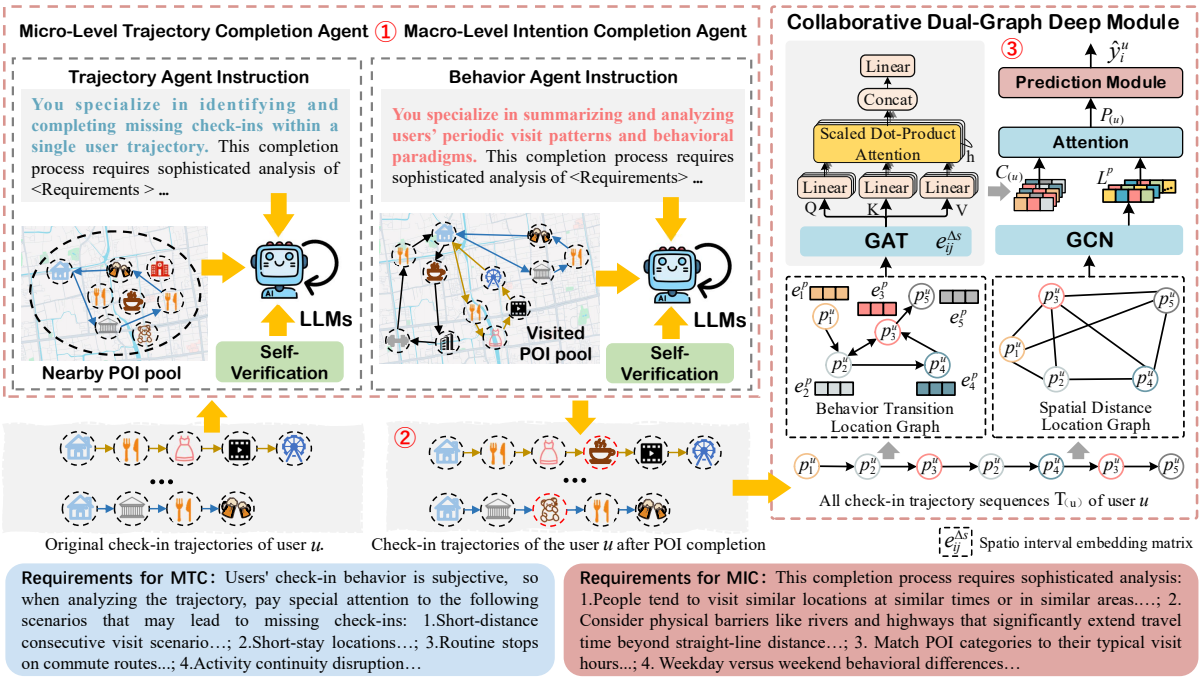

Figure 2: The overview of our proposed LaMDA. (1) The process begins with raw user check-in data, which undergoes enhancement through two specialized completion agents: micro-level trajectory completion (MTC) and macro-level intention completion (MIC). (2) These agents generate enriched user check-in records that capture detailed movement patterns and broader behavioral intentions. (3) The enhanced data is fed into the collaborative dual-graph deep module for model training.

Definition 4 (Spatial Distance Location Graph). To model global geographical relationships among locations, we construct an undirected spatial graph $\mathcal{G}_p = (\mathcal{V}_p, \mathcal{E}_p, \mathcal{W}_p)$, where $\mathcal{V}_p = \mathcal{P}$ includes all locations. An edge $e_{i \leftrightarrow j} = \langle p_i, p_j \rangle \in \mathcal{E}p$ is created between each location pair, and its weight $w^p_{i,j} \in \mathcal{W}_p$ quantifies the spatial proximity (e.g., the inverse of geographical distance) between p_i and p_j, follows GSTN (Wang et al., 2022b).

4 Methodology

This section provides a detailed description of the proposed model, LaMDA, and presents its overall architecture in Figure 2. It consists of three main components: (1) semantic location transformation, (2) two location completion agents, and (3) a collaborative dual-graph deep module.

4.1 Semantic Location Transformation

In this section, we discuss why converting latitude and longitude coordinates into textual location descriptions and how we can transform user visit data into narrative descriptions. Figure 1 and Appendix C.2 show that LLMs perform better when locations are described in text rather than as raw coordinates. They can identify geographical features that may cause the actual travel distance to exceed the straight line distance. Inspired by SeCor (Wang et al., 2024), we utilized the Geopy[2] library to generate standardized textual descriptions of real geographic locations corresponding to latitude and longitude coordinates <DetailedLocation>, using the format: "<District/County> <Street/Road> <Number> (<PoiCategoryName>)". For example, the coordinates "40.78401843692215,-73.97452399134636" are translated to the textual description "New York County West 82nd Street No.100 (American Restaurant)". We perform reverse geocoding validation on each text-converted address using the Geopy library to prevent incorrect addresses from misleading the LLM.

To enhance the ability of LLMs to understand user mobility patterns, we generated textual descriptions <Check-in descriptions> of user visit records in the format: "At <Time>, user <UserId> visited POI id <PoiId> which detailed location and category is <DetailedLocation>". This natural language temporal expression strengthens the LLMs' ability to capture time continuity. At the same time, the detailed location descriptions activate the

[2] https://geopy.readthedocs.io/en/latest/

LLMs' spatial knowledge.

4.2 Location Completion Agents

Missing behavioral data introduces bias into existing methods for capturing user preferences. Drawing on real-world human behavioral patterns, we develop LLM-powered agents for completing user check-in data by capturing dynamic short-term preferences (at the micro-level, based on check-in trajectories) and long-term preferences (at the macro-level, based on all user check-in records).

User Trajectory Filtering. We filter users with extremely high- or extremely low-frequency check-in trajectories. All filtered-out data are denoted as T_u^o. The detailed filtering procedure in Appendix A.1. The micro-level trajectory completion agent relies on T_u^s, and the macro-level intention completion agent depends on T_u^f. The purpose of data filtering is to enhance the reliability of LLM-generated locations. Before feeding into the collaborative dual-graph deep module, T_u^o is restored.

4.2.1 Micro-Level Trajectory Completion

User trajectories reflect short-term behavioral changes within current movement patterns. Our micro-level trajectory completion agent leverages behavioral preferences to enable LLMs to identify locations that users likely visited within their current trajectory but did not check-in. From a trajectory-based perspective, we can address the issue of unrecorded visits within a user's trajectory by considering the continuity of their activities.

Tailored Trajectory Agent Instruction. As illustrated in Figure 2, the micro-level trajectory completion agent is guided by a trajectory-specific agent instruction, where the *System Prompt* (highlighted in bold red) serves as the role-specific instruction. **The *Requirements* section specifies trajectory-grounded behavioral constraints for the agent.** Overall, the agent instruction consists of the *System Prompt, UserId, Trajectory id, Trajectory total, Trajectory description, Nearby POI pool*, and the *Requirements* section. The complete agent instruction $\mathcal{S}_t$ for micro-level trajectory completion is provided in Appendix A.3.

Nearby POI Pool Generation. To enhance LLMs' capability to simulate user check-ins influenced by subjective forgetfulness due to frequent short-term activity changes and to mitigate hallucinations, we generate a multi-structured candidate set of nearby locations (POI pool) for LLM reasoning. The specific generation process is outlined in Algorithm 1 and Appendix A.2. First, we create a POI-level pool $\mathcal{N}_c$ of nearby POIs around each POI visited by a user, limiting the search to within $radius = 0.1$km of the current POI, as greater distances are less consistent with user behavior influenced by attention shifts. Next, we combine the POI-level nearby POI pools within the same user trajectory into a trajectory-level nearby POI pool $\mathcal{N}_{T_u^s}$, removing duplicates during the process.

Trajectory Completion. Through carefully designed agent instruction, LLMs effectively interpret user preferences and behavioral patterns, infer potential missing user access actions that may occur between two observed access events, and produce their outputs in a standardized and consistent format. The detailed output instruction is provided in Appendix A.4.

4.2.2 Macro-Level Intention Completion

In this section, the macro-level intention completion agent focuses on capturing the periodic behavioral patterns of users and latent associations among trajectories based on holistic visit data.

Check-in History Segmentation. We first introduce a segmentation strategy for user check-in data. This approach mitigates the impact of overly long histories that may contain outdated interests while also improving inference efficiency. On the other hand, this strategy also enables us to assign a well-defined *Count id* (<Count id>) to each segment of the user's historical behavior. The detailed segmentation procedure is provided in Appendix A.5.

Tailored Behavior Agent Instruction. As illustrated in Figure 2, the macro-level intention completion agent is guided by a behavior-oriented agent instruction, where the *System Prompt* (highlighted in bold blue) serves as the role-specific instruction. **The *Requirements* section specifies behavior-level constraints derived from the user's overall activity history.** Specifically, in the second point of the *Requirements*, we ensure that the LLMs consider geographical features, such as rivers, when generating supplementary locations. This constraint prevents inconsistencies between the actual travel distance and physical proximity. Overall, the agent instruction includes the *System Prompt, UserId, Count id, Count total, All trajectory descriptions, Visited POI pool*, and the *Requirements* section. The complete agent instruction $\mathcal{S}_a$ for macro-level intention completion is provided in Appendix A.6.

Visited POI Pool Generation. The visited POI

pool aggregates all real-world locations a user has visited into a single set. LLMs focus on the user's most immediate interests as reflected by the locations in the current segment (<Count id>) while still having access to the broader context of their long-term preferences through the Visited POI Pool.

4.2.3 Self-Verification Mechanism

We incorporate a self-verification mechanism into both location completion agents. This mechanism encourages LLMs to reflect on the reasonableness of their location completions and make self-adjustments, thereby further improving the reliability of generated locations. The detailed design of the self-verification process is in Appendix A.7.

4.2.4 All Location Completion

We first insert micro-level completions into the original trajectories to obtain T_u^t, and then insert macro-level completions into T_u^t to determine consistent insertion positions and preserve behavioral coherence. Considering the insertion complexity, we restore all filtered-out data T_u^o only at the final stage. The final training dataset, enhanced from multiple perspectives, is denoted as T_u^a. Notably, each supplementary location corresponds to an actual location point with distinct latitude and longitude coordinates.

4.3 Collaborative Dual-Graph Deep Module

LaMDA does not freely generate POIs during completion. Instead, it incorporates robustness constraints by grounding the LLM output in a real POIs pool and using verification to reduce unreliable suggestions. First, we convert coordinates into standardized textual addresses and validate them with reverse geocoding to prevent incorrect location strings from misleading the model. Second, at the micro level, we use a nearby POI pool and restrict candidates to a 0.1 km radius to avoid overly large candidate sets and reduce noisy or irrelevant completions. Third, at the macro level, we use a visited POI pool and limit completion to the user's previously visited interest set, which helps avoid introducing new interests that are not supported by the data. Fourth, both agents include a self-verification step that prompts the LLM to check and revise its own completions for consistency.

Although LLMs demonstrate strong semantic reasoning capabilities, they face limitations regarding computational cost. Hence, we leverage the strengths of LLMs in data augmentation and introduce a lightweight collaborative dual-graph deep module for effective preference modeling and next location prediction.

Behavior Transition Location Graph Encoder. For a target user u, the location node embeddings from the input sequence are denoted as $E_u^{\mathcal{P}} = \{e_i^p | i = 1, 2, ..., |\mathcal{P}|\}$. We first design a Spatial Graph Attention Network (SGAT) to process the behavior transition location graph $\mathcal{G}_u$. We omit temporal modeling to avoid introducing noise because the timestamps inferred by LLMs are artificial and not grounded in real-world user behavior. To better capture relationships between non-adjacent locations, we follow the location modeling approach of STAN (Luo et al., 2021) and construct spatial interval embeddings $e_{ij}^{\Delta s}$ from T_u:

$$\begin{aligned} \text{Message}_i &= \sum_{\langle p_j, p_i \rangle \in \mathcal{E}_u} \gamma_{ij} \cdot e_j^p, \\ \gamma_{ij} &= \text{softmax}(\mathbf{W}(e_i^p \odot e_j^p + e_{ij}^{\Delta s}), \end{aligned} \tag{1}$$

where γ_{ij} represent the attention weights for the neighboring nodes of location i, $\mathbf{W}$ is a trainable matrix, $\odot$ represents the element-wise product, and softmax$(\cdot)$ is the normalization function. $\bar{E}_u^{\mathcal{P}} = \{\bar{e}_i^p | i = 1, 2, ..., n\} \in \mathbb{R}^{n \times d}$ denotes the node representations generated after the SGAT applied to the behavior transition location graph and d is hidden dimension. Finally, we employ a multi-head self-attention mechanism as the readout function to capture fine-grained and latent correlations among locations:

$$\begin{aligned} &Q = \bar{E}_u^{\mathcal{P}} \mathbf{W}_Q, \quad K = \bar{E}_u^{\mathcal{P}} \mathbf{W}_K, \quad V = \bar{E}_u^{\mathcal{P}} \mathbf{W}_V, \\ &\text{head}_i = \text{softmax}\left(\frac{QK^\top}{\sqrt{d}}\right) V, \\ &C_{(u)} = \text{Concat}(\text{head}_1, \ldots, \text{head}_h), \end{aligned} \tag{2}$$

where $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V \in \mathbb{R}^{d \times d}$ are trainable matrices, $h = 2$, $C_{(u)}$ is user's sequential representation.

Spatial Distance Location Graph Encoder. For the spatial distance location graph, we adopt Graph Convolutional Networks (GCNs) as the graph encoder, leveraging their capability to aggregate information from neighboring nodes. The update function of each layer is defined as follows:

$$l_i^{(k)} = l_i^{(k-1)} + \sum_{j \in \mathcal{N}_i} \frac{w_{i,j}^p}{\sqrt{|\mathcal{N}_i| \cdot |\mathcal{N}_j|}} W_p^{(k-2)} l_j^{(k-2)}, \tag{3}$$

where the convolution kernel at the $(k-2)^{th}$ layer, denoted as $W_p^{(k-2)}$, governs the transformation of

node features. The distance-aware coefficient $w^p_{i,j}$ serves as an adaptive weighting term, enabling the GCN to modulate the influence of neighboring locations based on their geographical proximity. After k layers of graph convolution, we apply mean pooling across all layers to integrate multi-level spatial representations:

$$l^p_i = \frac{1}{|k|} \sum_{t=0}^{|k|} l^{(t)}_i, \tag{4}$$

where the set of final spatial node representations for all locations as $L^p = \{l^p_i | i = 1, 2, ..., |\mathcal{P}|\} \in \mathbb{R}^{|\mathcal{P}| \times d}$.

Model User Location Preferences. We obtain the user's sequential representation $C_{(u)}$ and the spatial node representation L^p. By integrating these two perspectives, we employ a cross-attention mechanism to capture fine-grained representations of user location preferences $P_{(u)}$:

$$P_{(u)} = \text{Attn}\left(W^p_Q C^p_{(u)}, W^p_K L^p_{T^a_{(u)}}, W^p_V L^p_{T^a_{(u)}}\right), \tag{5}$$

where W^p_Q, W^p_K, W^p_V are trainable matrices. $\text{Attn}(\cdot)$ denotes an attention function, and $C^p_{(u)} = LayerNorm(C_{(u)})$.

Location Prediction. We leverage the target users' location preferences $P_{(u)}$ and sequential representation $C_{(u)}$ to generate the score probability $\hat{y}^u_i$ for the next location:

$$\hat{y}^u_i = \text{softmax}\left(\alpha e^p_i C^\top_{(u)} + (1-\alpha) l^p_i P^\top_u\right), \tag{6}$$

where $\alpha \in [0, 1]$ is a balance coefficient. We leverage a supervised cross-entropy loss to optimize model, loss is formulated as:

$$\mathcal{L} = -\frac{1}{|T^a_{(u)}|} \sum_{u \in T^a_{(u)}} y^u_i \log\left(\hat{y}^u_i\right) + \eta \|\Theta\|^2_2, \tag{7}$$

where η is weight for the L_2 regularization and Θ represents model trainable parameters. y^u_i is the corresponding target location.

5 Experiments

5.1 Experimental Setup[3]

Evaluation Datasets. To assess the effectiveness of our proposed LaMDA model, we conducted experiments on datasets collected from Gowalla-CA[4], Foursquare-TKY, and Foursquare-NYC[5], all datasets are widely used location-based social networking platform. More details about datasets in Appendix B.1

Metrics. We adopt two standard metrics widely used in the location prediction methods: Recall (**Recall@N**), Normalized Discounted Cumulative Gain (**NDCG@N**), where $N \in \{2, 5, 10\}$.

5.2 Experiment Results[6]

5.2.1 Main Performance Comparison

We conducted a comprehensive evaluation of LaMDA across three real-world datasets and compared its performance against ten representative baseline methods selected from diverse modeling paradigms, including graph-based, sequential, complex graph-based, and diffusion-based approaches. Table 1 shows that LaMDA consistently outperforms competing methods on key metrics such as Recall and NDCG. Notably, on the most challenging dataset, Gowalla, LaMDA achieved significant improvements over the strongest baseline, Diff-POI. Specifically, LaMDA outperformed Diff-POI by 9.9% and 8.1% on average in terms of LaMDA_D and LaMDA_Q, respectively. The superior performance of LaMDA_D, which has a substantially larger parameter count than LaMDA_Q, suggests that leveraging more advanced LLMs yields better results for location completion. LaMDA demonstrates superior performance across all evaluation metrics, confirming its effectiveness and adaptability in modeling complex user trajectory scenarios. Compared with (c1) and (c2), (c3) and (c4) improve next location prediction accuracy by leveraging richer graph structures and generative diffusion mechanisms, respectively. However, LaMDA introduces two completion agents based on semantic textual descriptions for data augmentation, yielding more substantial improvements in the model's ability to predict user mobility.

[3] **The detailed description of baselines, model size, and implementation are presented in Appendix B.2, Appendix B.3, and Appendix B.4.**

[4] https://snap.stanford.edu/data/loc-gowalla.html

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

[6] **This section includes five research questions, three of which are answered in the main text.** Details on the hyperparameter study, and visualizations of the LLM-based completion results are provided in Appendix C.

CA													
Metrics	SR-GNN	LightGCN	DGCF	LSTPM	STAN	GTR	SGRec	DRAN	STHGCN	Diff-DGMN	Diff-POI	**LaMDA_Q**	**LaMDA_D**
Recall@2	0.1531	0.1102	0.1147	0.1904	0.2195	0.2346	0.1932	0.2133	0.1791	0.2653	0.2652	**0.2918**	**0.2967**
Recall@5	0.1978	0.1348	0.1396	0.2049	0.2364	0.2747	0.2286	0.2466	0.2415	0.3091	0.3085	**0.3302**	**0.3361**
Recall@10	0.2384	0.1699	0.1721	0.2618	0.2994	0.3164	0.2718	0.3056	0.2825	0.3443	0.3470	**0.3630**	**0.3679**
NDCG@2	0.1402	0.0914	0.0995	0.1431	0.1917	0.2129	0.1573	0.2038	0.1608	0.2535	0.2540	**0.2816**	**0.2865**
NDCG@5	0.1621	0.1162	0.1319	0.1588	0.2152	0.2362	0.1662	0.2168	0.1890	0.2731	0.2733	**0.2989**	**0.3041**
NDCG@10	0.1751	0.1432	0.1561	0.1742	0.2268	0.2533	0.1704	0.2367	0.2120	0.2845	0.2858	**0.3095**	**0.3143**
TKY													
Metrics	SR-GNN	LightGCN	DGCF	LSTPM	STAN	GTR	SGRec	DRAN	STHGCN	Diff-DGMN	Diff-POI	**LaMDA_Q**	**LaMDA_D**
Recall@2	0.4600	0.3917	0.4204	0.5029	0.5105	0.5684	0.5091	0.5225	0.4879	0.5924	0.5967	**0.6108**	**0.6136**
Recall@5	0.5257	0.4473	0.4913	0.5513	0.5489	0.5822	0.5488	0.5570	0.5684	0.6269	0.6322	**0.6468**	**0.6500**
Recall@10	0.5458	0.4936	0.5310	0.5795	0.6167	0.6302	0.6173	0.6210	0.6333	0.6628	0.6660	**0.6738**	**0.6773**
NDCG@2	0.4537	0.4207	0.4299	0.4724	0.4990	0.5258	0.4715	0.5089	0.4773	0.5788	0.5831	**0.5987**	**0.6021**
NDCG@5	0.4788	0.4354	0.4376	0.4881	0.5264	0.5561	0.4905	0.5390	0.5036	0.5988	0.6037	**0.6149**	**0.6186**
NDCG@10	0.4908	0.4403	0.4558	0.4962	0.5554	0.5795	0.5112	0.5602	0.5401	0.6096	0.6146	**0.6237**	**0.6275**
NYC													
Metrics	SR-GNN	LightGCN	DGCF	LSTPM	STAN	GTR	SGRec	DRAN	STHGCN	Diff-DGMN	Diff-POI	**LaMDA_Q**	**LaMDA_D**
Recall@2	0.5052	0.3789	0.3368	0.5754	0.6027	0.6213	0.5734	0.5859	0.5642	0.6562	0.6564	**0.6725**	**0.6798**
Recall@5	0.5465	0.4363	0.4167	0.6020	0.6358	0.6459	0.6175	0.6253	0.6128	0.6688	0.6746	**0.6869**	**0.6955**
Recall@10	0.5951	0.4519	0.5653	0.6312	0.6533	0.6627	0.6424	0.6478	0.6224	0.6842	0.6880	**0.6986**	**0.7060**
NDCG@2	0.5191	0.3819	0.4847	0.5524	0.5887	0.6007	0.5490	0.5702	0.5635	0.6401	0.6497	**0.6679**	**0.6752**
NDCG@5	0.5319	0.3867	0.4922	0.5596	0.6092	0.6230	0.5559	0.5881	0.5787	0.6503	0.6579	**0.6745**	**0.6823**
NDCG@10	0.5495	0.3903	0.5041	0.5681	0.6124	0.6388	0.5613	0.5956	0.5869	0.6553	0.6623	**0.6783**	**0.6856**

Table 1: Performance comparison of all baselines on three datasets. "LaMDA_Q" and "LaMDA_D" denote the LaMDA results obtained using Qwen3-turbo (parameters approximately 72B) and DeepSeek-V3 (with 671B parameters), respectively. **(c1)** Graph-based methods (SR-GNN, LightGCN and DGCF); **(c2)** Sequence-based methods (LSTPM, STAN and GTR); **(c3)** Complex graph-based methods(SGRec, DRAN, STHGCN and LaMDA); **(c4)** Diffusion-based method (Diff-DGMN and Diff-POI).

5.2.2 Analysis on Critical Components

Due to the outstanding performance of LaMDA_D, we use LaMDA_D to verify the effectiveness of each key component, we design four model variants: **only CDGD**: This variant removes all location completion agents and only uses the raw dataset to perform the next location prediction via the collaborative dual-graph deep module (CDGD). **w/o MIC**: This variant excludes only the macro-level intention completion agent. **w/o MTC**: This variant removes only the micro-level trajectory completion agent. **w/o SV**: This variant removes the self-verification mechanism from both the macro-level intention and micro-level trajectory completion agents.

Based on the results shown in Figure 3, we draw the following conclusions: overall, LaMDA consistently outperforms all variants, demonstrating that each key component contributes positively to the model's performance. This validates our dual-perspective approach to location completion, which considers both users' overall visit patterns (macro-level) and individual trajectory behaviors

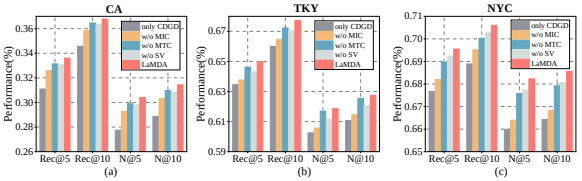

Figure 3: Performance comparison of key components in LaMDA on three datasets.

(micro-level). The performance degradation observed in the w/o SV variant confirms that the self-verification mechanism enhances the quality and reliability of location completions generated by LLMs, building upon our carefully designed completion agent instructions. While both w/o MIC and w/o MTC variants underperform compared to the LaMDA model, the w/o MTC variant shows a smaller performance decline than w/o MIC. We attribute this to the nature of the micro-level trajectory completion agent, which is designed to address short-term shifts in user interest or consecutive, session-based activities. Such scenarios occur less frequently than the routine check-in behaviors

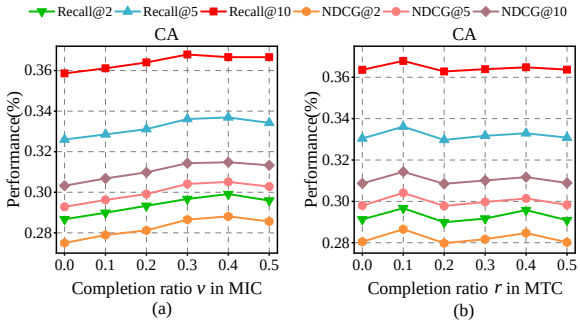

Figure 4: Performance impact of location completion ratio v in macro-level intention completion and location completion ratio r in micro-level trajectory completion.

captured by the macro-level agent.

5.2.3 Sensitivity Analysis of Completion Ratio

Location Completion Ratio v in Macro-Level Intention Completion. To evaluate the impact of the v in the macro-level intention completion (MIC) agent's model performance in LaMDA_D, we conducted experiments on the CA dataset while maintaining the completion rate (r) of the micro-level trajectory completion (MTC) at 0.1. The v represents the ratio of locations completed in MIC to the total number of locations visited by the target user. As shown in Figure 4(a), model performance improves significantly when v increases from 0 to 0.3. This improvement suggests that due to the periodic nature of user behavior, locations where users are present but do not check in are likely to be found within the pool of previously visited locations. While optimal performance is achieved at $v = 0.4$, the marginal improvement between $v = 0.3$ and $v = 0.5$ is minimal. Therefore, considering computational efficiency, we selected $v = 0.3$ for our experiments.

Location Completion ratio r in Micro-Level Trajectory Completion. We examined the effect of the r in the MTC agent on model performance (in LaMDA_D) while maintaining the MIC completion rate (v) at 0.3 using the CA dataset. The r represents the ratio of locations completed in MTC to the total number of locations in each user trajectory. Figure 4(b) illustrates that model performance shows notable improvement at $r = 0.1$, but decreases as r increases further. We attribute this decline to the fact that MTC selects from a POIs pool consisting of user-visited locations and nearby locations within a 0.1km radius. Although LLMs possess strong reasoning capabilities, excessive location completion for individual trajectories interferes with the accurate modeling of the user's core preferences.

6 Related Work

Existing works to next location prediction can be broadly grouped into several categories: sequence-based methods, graph-based methods, and LLM-based methods. The graph-based methods, such as Diff-POI (Qin et al., 2023b) and Diff-DGMN (Zuo and Zhang, 2024) utilize dual graph structures and diffusion models to capture complex mobility relationships; TAU (Zhuang et al., 2024) addresses the location completion by integrating uncertainty-guided trajectory augmentation with sequential modeling. LLM-based methods, such as SeCor (Wang et al., 2024) treats location prediction as multi-modal alignment, fusing collaborative filtering embeddings with LLM semantic representations.

For detailed related work and additional discussion, please refer to Appendix D.

7 Conclusion

In this work, we propose LaMDA, a novel LLM-driven multi-perspective data augmentation framework that enhances user check-in data by completing locations. Specifically, LaMDA incorporates a micro-level trajectory completion agent and a macro-level intention completion agent, which identify missing locations that users likely visited but did not check in, based on short-term behaviors and long-term preferences, respectively. We observe that LLMs struggle to interpret raw geographical coordinates effectively. To address this limitation, LaMDA converts geographical coordinates into descriptive text representations. We design distinct POI pools for each completion agent, along with a self-verification mechanism to enhance the accuracy of LLMs' inference. Finally, we employ a collaborative dual-graph deep module to effectively leverage semantically enhanced data. Evaluations on three real-world datasets under two LLM backbone confirm that LaMDA consistently outperforms state-of-the-art location prediction models.

Acknowledgements

This work is partially supported by the National Natural Science Foundation of China under Grant (No. 62576083, 2024-MSBA-49).

Limitations

This study specifically addresses data incompleteness (missing check-ins due to user subjectivity) rather than data uncertainty (noisy GPS signals or ambiguous check-ins). While closely related, these are distinct challenges. We recognize that real-world mobility data suffers from both issues simultaneously. Future work will aim to bridge this gap by integrating our completion framework with uncertainty modeling techniques to create a more robust representation of user mobility.

Given the fundamental paradigm differences and the prohibitive computational costs of fine-tuning LLM methods, our focus remains on a data-centric framework that balances enhanced data quality with operational efficiency. Accordingly, our primary performance comparisons are conducted against lightweight deep learning models. Future work will explore converting both completed and original data into textual inputs for LLMs and fine-tuning these models for more accurate next-location prediction, enabling direct comparison with state-of-the-art fine-tuned LLM approaches.

References

Jingmin An, Ming Gao, and Jiafu Tang. 2024. Mvsthgl: Multi-view hypergraph learning with spatial-temporal periodic interests for next poi recommendation. *ACM Trans. Inf. Syst.*, 42(6).

Keqin Bao, Jizhi Zhang, Yang Zhang, Wenjie Wang, Fuli Feng, and Xiangnan He. 2023. Tallrec: An effective and efficient tuning framework to align large language model with recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems*, pages 1007–1014.

Jinpeng Chen, Huachen Guan, Huan Li, Fan Zhang, Liwei Huang, Guangyao Pang, and Xiongnan Jin. 2024. Pacific: Enhancing sequential recommendation via preference-aware causal intervention and counterfactual data augmentation. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pages 249–258.

Yakun Chen, Xianzhi Wang, and Guandong Xu. 2023. Gatgpt: A pre-trained large language model with graph attention network for spatiotemporal imputation. *arXiv preprint arXiv:2311.14332*.

Jiawei Cheng, Jingyuan Wang, Yichuan Zhang, Jiahao Ji, Yuanshao Zhu, Zhibo Zhang, and Xiangyu Zhao. 2025. Poi-enhancer: An llm-based semantic enhancement framework for poi representation learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 11509–11517.

Yizhou Dang, Yuting Liu, Enneng Yang, Guibing Guo, Linying Jiang, Xingwei Wang, and Jianzhe Zhao. 2024. Repeated padding for sequential recommendation. In *Proceedings of the 18th ACM Conference on Recommender Systems*, pages 497–506.

Chenghua Duan, Wei Fan, Wei Zhou, Hu Liu, and Junhao Wen. 2023. Clsprec: Contrastive learning of long and short-term preferences for next poi recommendation. In *Proceedings of the 32nd acm international conference on information and knowledge management*, pages 473–482.

Shanshan Feng, Haoming Lyu, Fan Li, Zhu Sun, and Caishun Chen. 2024. Where to move next: Zero-shot generalization of llms for next poi recommendation. In *2024 IEEE Conference on Artificial Intelligence (CAI)*, pages 1530–1535. IEEE.

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

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 639–648.

Tianhao Huang, Xuan Pan, Xiangrui Cai, Ying Zhang, and Xiaojie Yuan. 2024a. Learning time slot preferences via mobility tree for next poi recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 8535–8543.

Xuanwen Huang, Kaiqiao Han, Yang Yang, Dezheng Bao, Quanjin Tao, Ziwei Chai, and Qi Zhu. 2024b. Can gnn be good adapter for llms? In *Proceedings of the ACM Web Conference 2024*, pages 893–904.

Yonghao Huang, Pengxiang Lan, Xiaokang Li, Yihao Zhang, and Kaibei Li. 2024c. Residual spatiotemporal collaborative networks for next poi recommendation. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 144–155. Springer.

Pengxiang Lan, Haoyu Xu, Enneng Yang, Yuliang Liang, Guibing Guo, Jianzhe Zhao, and Xingwei Wang. 2025a. Efficient and effective prompt tuning via prompt decomposition and compressed outer product. *arXiv preprint arXiv:2502.12200*.

Pengxiang Lan, Enneng Yang, Yuting Liu, Guibing Guo, Jianzhe Zhao, and Xingwei Wang. 2025b. Ept: Efficient prompt tuning by multi-space projection and

prompt fusion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 24366–24374.

Pengxiang Lan, Yihao Zhang, Haoran Xiang, Yuhao Wang, and Wei Zhou. 2023. Spatio-temporal position-extended and gated-deep network for next poi recommendation. In *International Conference on Database Systems for Advanced Applications*, pages 505–520. Springer.

Peibo Li, Maarten de Rijke, Hao Xue, Shuang Ao, Yang Song, and Flora D Salim. 2024a. Large language models for next point-of-interest recommendation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1463–1472.

Ranzhen Li, Yanyan Shen, and Yanmin Zhu. 2018. Next point-of-interest recommendation with temporal and multi-level context attention. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 1110–1115. IEEE.

Yang Li, Tong Chen, Yadan Luo, Hongzhi Yin, and Zi Huang. 2021a. Discovering collaborative signals for next poi recommendation with iterative seq2graph augmentation. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 1491–1497.

Zeyu Li, Wei Cheng, Haiqi Xiao, Wenchao Yu, Haifeng Chen, and Wei Wang. 2021b. You are what and where you are: Graph enhanced attention network for explainable poi recommendation. In *Proceedings of the 30th ACM International conference on information & knowledge management*, pages 3945–3954.

Zhonghang Li, Lianghao Xia, Jiabin Tang, Yong Xu, Lei Shi, Long Xia, Dawei Yin, and Chao Huang. 2024b. Urbangpt: Spatio-temporal large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 5351–5362.

Zhuoxuan Li, Jieyuan Pei, Tangwei Ye, Zhongyuan Lai, Zihan Liu, Fengyuan Xu, Qi Zhang, and Liang Hu. 2025. Gtr-mamba: Geometry-to-tangent routing mamba for hyperbolic poi recommendation. *arXiv preprint arXiv:2510.22942*.

Defu Lian, Yongji Wu, Yong Ge, Xing Xie, and Enhong Chen. 2020. Geography-aware sequential location recommendation. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2009–2019.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. 2024a. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Chenxi Liu, Kethmi Hirushini Hettige, Qianxiong Xu, Cheng Long, Shili Xiang, Gao Cong, Ziyue Li, and Rui Zhao. 2025a. St-llm+: Graph enhanced spatio-temporal large language models for traffic prediction. *IEEE Transactions on Knowledge and Data Engineering*.

Chenxi Liu, Sun Yang, Qianxiong Xu, Zhishuai Li, Cheng Long, Ziyue Li, and Rui Zhao. 2024b. Spatial-temporal large language model for traffic prediction. In *2024 25th IEEE International Conference on Mobile Data Management (MDM)*, pages 31–40. IEEE.

Zhao Liu, Wei Liu, Huajie Zhu, Jianxing Yu, Jian Yin, Wang-Chien Lee, and Shun Wang. 2025b. Geography-aware large language models for next poi recommendation. *arXiv preprint arXiv:2505.13526*.

Jing Long, Liang Qu, Guanhua Ye, Tong Chen, Quoc Viet Hung Nguyen, and Hongzhi Yin. 2024. Unleashing the power of large language models for group poi recommendations. *arXiv preprint arXiv:2411.13415*.

Zewen Long, Liang Wang, Qiang Liu, and Shu Wu. 2023. Personalized interest sustainability modeling for sequential poi recommendation. In *Proceedings of the 32nd ACM international conference on information and knowledge management*, pages 4145–4149.

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

Rohin Manvi, Samar Khanna, Gengchen Mai, Marshall Burke, David Lobell, and Stefano Ermon. 2023. Geollm: Extracting geospatial knowledge from large language models. *arXiv preprint arXiv:2310.06213*.

Yifang Qin, Yifan Wang, Fang Sun, Wei Ju, Xuyang Hou, Zhe Wang, Jia Cheng, Jun Lei, and Ming Zhang. 2023a. Disenpoi: Disentangling sequential and geographical influence for point-of-interest recommendation. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*, pages 508–516.

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

K2 Sun, Chenliang Li, and Tieyun Qian. 2024. City matters! a dual-target cross-city sequential poi recommendation model. *ACM Trans. Inf. Syst.*, 42(6).

Ke Sun, Tieyun Qian, Tong Chen, Yile Liang, Quoc Viet Hung Nguyen, and Hongzhi Yin. 2020. Where to go next: Modeling long-and short-term user preferences for point-of-interest recommendation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 214–221.

Zhenglin Wan, Anjun Gao, Xingrui Yu, Pingfu Chao, Jun Song, and Maohao Ran. 2025. Poi recommendation via multi-objective adversarial imitation learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 12676–12684.

Dongsheng Wang, Yuxi Huang, Shen Gao, Yifan Wang, Chengrui Huang, and Shuo Shang. 2025. Generative next poi recommendation with semantic id. *arXiv preprint arXiv:2506.01375*.

Shirui Wang, Bohan Xie, Ling Ding, Xiaoying Gao, Jianting Chen, and Yang Xiang. 2024. Secor: Aligning semantic and collaborative representations by large language models for next-point-of-interest recommendations. In *Proceedings of the 18th ACM Conference on Recommender Systems*, pages 1–11.

Xiang Wang, Hongye Jin, An Zhang, Xiangnan He, Tong Xu, and Tat-Seng Chua. 2020. Disentangled graph collaborative filtering. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*, pages 1001–1010.

Zhaobo Wang, Yanmin Zhu, Haobing Liu, and Chunyang Wang. 2022a. 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*, pages 1154–1163.

Zhaobo Wang, Yanmin Zhu, Qiaomei Zhang, Haobing Liu, Chunyang Wang, and Tong Liu. 2022b. Graph-enhanced spatial-temporal network for next poi recommendation. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 16(6):1–21.

Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. 2019. Session-based recommendation with graph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 346–353.

Ziqing Wu, Zhu Sun, Dongxia Wang, Lu Zhang, Jie Zhang, and Yew Soon Ong. 2024. Mrp-llm: Multitask reflective large language models for privacy-preserving next poi recommendation. *arXiv preprint arXiv:2412.07796*.

Yang Xu, Gao Cong, Lei Zhu, and Lizhen Cui. 2024. Mmpoi: A multi-modal content-aware framework for poi recommendations. In *Proceedings of the ACM Web Conference 2024*, pages 3454–3463.

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

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Song Yang, Jiamou Liu, and Kaiqi Zhao. 2022. 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*, pages 1144–1153.

Xie Yu, Jingyuan Wang, Yifan Yang, Qian Huang, and Ke Qu. 2024. Bigcity: A universal spatiotemporal model for unified trajectory and traffic state data analysis. *arXiv preprint arXiv:2412.00953*.

Jun Zeng, Yizhu Zhao, Yang Yu, Min Gao, and Wei Zhou. 2021. The missing poi completion based on bidirectional masked trajectory model. In *Proceedings of Collaborative Computing: Networking, Applications and Worksharing*, pages 229–243.

Yang Zhang, Fuli Feng, Jizhi Zhang, Keqin Bao, Qifan Wang, and Xiangnan He. 2025. Collm: Integrating collaborative embeddings into large language models for recommendation. *IEEE Transactions on Knowledge and Data Engineering*.

Yihao Zhang, Pengxiang Lan, Yuhao Wang, and Haoran Xiang. 2022. Spatio-temporal mogrifier lstm and attention network for next poi recommendation. In *2022 IEEE International Conference on Web Services (ICWS)*, pages 17–26. IEEE.

Kun Zhou, Hui Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. 2020. S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization. In *Proceedings of the 29th ACM international conference on information & knowledge management*, pages 1893–1902.

Zhuang Zhuang, Tianxin Wei, Lingbo Liu, Heng Qi, Yanming Shen, and Baocai Yin. 2024. Tau: trajectory data augmentation with uncertainty for next poi recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 22565–22573.

Jiankai Zuo and Yaying Zhang. 2024. Diff-dgmn: A diffusion-based dual graph multi-attention network for poi recommendation. *IEEE Internet of Things Journal*.

Appendix

Supplementary materials are provided in the appendix. We provide additional method details, more complete experimental settings, comprehensive results and analysis, and a more detailed discussion of related work.

- Appendix A: Method details.
- Appendix B: Complete experimental settings.
- Appendix C: Comprehensive results and analysis.
- Appendix D: Detailed discussion of related work.

A Methods Details

A.1 User Trajectory Filtering Deatlis

We filtered user visit data based on two considerations: First, we removed trajectories where the target user visited more than ten locations daily, and the refined set of user trajectories is denoted as T_u^f. These high-density trajectories indicate users with strong check-in initiatives and a low likelihood of unrecorded visits. This step filters out a small subset of abnormally active users whose trajectory completion might contradict their actual travel intentions. Second, we further eliminated trajectories with fewer than ten check-ins, yielding a final trajectory set denoted as T_u^s. While shorter trajectories might indicate less user initiative, the limited data makes preference inference during completion potentially misleading. This step filters out a small number of inactive users. The resulting set T_u^s still retains substantial original check-in data. The micro-level trajectory completion agent relies on T_u^s, while the macro-level intention completion agent depends on T_u^f, as even short user trajectories can reflect periodic behavioral preferences.

A.2 Algorithm of Nearby POI Pool

The algorithm for constructing the nearby POI pool is presented in Algorithm 1. The Nearby POI Pool is designed for scenarios where user activities change continuously within short time periods (user's local activity space). Therefore, locations in this pool must be geographically proximate to the user's current trajectory. All locations in the Nearby POI Pool correspond to real-world venues, providing the micro-level trajectory completion agent with reliable decision options. As the radius $radius$ increases, behavioral patterns become less consistent with user behavior influenced by shifts in attention. Conversely, when $radius$ is too small, the model's available choices become limited. We therefore set $radius$ to 0.1 km based on practical considerations.

A.3 Instruction of Micro-Level Trajectory Completion Agent

We custom-tailor the following agent instruction for our micro-level trajectory completion agent in Figure 5.

A.4 Output Instructions of the Agents

Standard output format instruction of the location completion agent in Figure 6. Notably, in the output format of the macro-level intention completion agent, we replace "Trajectory id: <trajectory id>" with "Count id: <count id>", and substitute "<trajectory total>" with "<count total>". The reported count total is a hyperparameter shared by the two completion agents, which we discuss in detail in Sections 5.2.3. Unlike previous data augmentation methods that rely on masking, reordering, or trimming interaction sequences—approaches can undermine the understanding of user behavior semantics (Zhou et al., 2020; Chen et al., 2024). Similarly, we avoid randomly filling in the supplementary locations generated, as this would not align with the decisions made by the LLMs after interpreting user behavioral preferences and patterns.

A.5 Check-in History Segmentation

Our macro-level intention completion agent relies on processing textual descriptions of user visit histories with LLMs. However, excessively long historical sequences can impair the model's reasoning efficiency and may contain outdated interests no longer relevant to the user's current preferences. To address this, we introduce a method to segment user check-in data. We define a hyper-parameter, $\varphi = 100$, as the maximum number of visits within a single data segment. For users with fewer than φ visited locations, their entire history is encapsulated within a single segment, assigned a unique <Count id>. For users with trajectories exceeding this length, we chronologically partition their history into multiple segments, each containing approximately φ visits and assigned its own distinct <Count id>.

Algorithm 1: Nearby POI Pool Search with Trajectory-Level

Input: A training dateset T_u^s of the target user $u \in U$

Required:, search radius r in kilometers, maximum POIs to return k, the user's latitude lat and longitude lon coordinates, specific POI categories cat, and textual location descriptions dl of the POIs; $BallTree$ creates a nearest neighbor searcher based on spherical distance; Rad converts the input latitude and longitude to radians; $\mathcal{N}_c \leftarrow \emptyset$; $\mathcal{N}_{T_u^s} \leftarrow \emptyset$.

$\mathcal{T} \leftarrow BallTree(T_u^s)$ ▷ Build ball tree spatial index;

for *each POI* $c \in T_u^s$ **do**

  $p \leftarrow (Rad(c.lat), Rad(c.lon))$

  (dis, ind) $\leftarrow (\mathcal{T}, p, r/6371)$ ▷ Convert km to radians

  $\mathcal{Z} \leftarrow zip$ (dis[0], ind[0])

  $\Gamma \leftarrow sort(\mathcal{N}[1:k])$ ▷ Select top-k nearest POIs

  for *each (dis, ind)* $\in \Gamma$ **do**

    poi_info $\leftarrow c[ind] + c.dl$

    $\mathcal{N}_c \leftarrow \mathcal{N}_c \cup$ poi_info

$\mathcal{N}_c$ is the nearby POI pool for each POI visited by users. **for** $t = 0$ *in* T_u^s **do**

  for $i = 0$ *in* $\mathcal{N}_c$ **do**

    if $\mathcal{N}_c(t)$[i] $\notin \mathcal{N}_{\mathcal{C}_t(u)}$:

    $\mathcal{N}_{T_u^s} \leftarrow \mathcal{N}_{T_u^s} \cup \mathcal{N}_c(t)$[i]

Output: $\mathcal{N}_{T_u^s}$: nearby POI pool for each trajectory of the target user.

Agent Instruction $\mathcal{S}_l$: Micro-Level Trajectory Completion

System Prompt: You are a location data analysis expert specializing in identifying and completing missing check-ins within a single user trajectory. I will provide you with a description of a user trajectory and a nearby POI pool constructed from POIs within 0.1 km of the POIs historically visited by the user. Your task is to analyze the user's behavioral patterns within this trajectory and identify potential missing check-in POIs, offering reasonable completions.

UserId*: <UserId>*

Trajectory id*: <Trajectory id>*

Trajectory total*: The total length of the user's trajectory <Trajectory id> is <Trajectory total>.*

Trajectory description*: The following data is a trajectory <Trajectory id> of user <UserId>: <Check-in descriptions>*

Nearby POI pool*: The POI pool within 0.1km around these visited POIs in <Trajectory id> is [<PoiId> <DetailedLocation>]*

Requirements*:*

Users' check-in behavior is subjective, so when analyzing the trajectory, pay special attention to the following scenarios that may lead to missing check-ins:

1.Short-Distance Consecutive Visit Scenario: After checking in at a previous POI, the user immediately proceeds to another nearby location but does not check-in.

2.Short-Stay Locations: The user visits certain locations briefly (e.g., convenience stores, coffee shops, fast-food restaurants), where check-ins are easily overlooked.

3.Routine Stops on Commute Routes: The user has regular stops along their daily commuting route but does not consistently check in at these locations.

4.Activity Continuity Disruption: The user's activities lack logical coherence, such as checking in directly from a residential location to a workplace and potentially missing intermediate locations like transportation hubs.

Figure 5: Agent instruction for the micro-level trajectory completion agent.

Instruction $\mathcal{S}_t$: Standard Output Format of the POI Completion Agent

Response: *Please organize your answer in a JSON format, following this structure: {"UserId": <UserId>, "Trajectory id": <trajectory id>, "POI completion":["Insert position": Between check-in at [time1](POI id <PoiId>) and [time2](POI id <PoiId>), "PoiId": <PoiId>]. You must suggest POIs for user that are at least 10% the count total <trajectory total>.*

Figure 6: Standard output format instruction of the location completion agent.

Agent Instruction $\mathcal{S}_a$: Macro-Level Intention Completion

System Prompt: You are an expert in Point-of-Interest (POI) recommendation, you specialize in summarizing and analyzing users' periodic visit patterns and behavioral paradigms. Since users' check-in behaviors are subjective, there exist instances where users forget to check in at certain locations they have visited, resulting in incomplete data. Your task is to leverage the existing All trajectory description of users and the Visited POI pool constructed from the POIs they have visited, to infer these potential POIs and fill in the gaps in their visit records.
UserId: *<UserId>*
Count id: *<Count id>*
Count total: *The total length of the user's all check-in data is <check-in total>.*
All trajectory description: *The following data is all check-in data of user <UserId>: <Check-in descriptions>*
Visited POI pool: *The POI pool contains POIs that the user has visited: [<PoiId> <DetailedLocation>]*
Requirements:
This completion process requires sophisticated analysis of
1. Consider the specific geographic locations and categories of the POIs that the user has visited in the All trajectory description to capture the user's historical preferences. People tend to visit similar locations at similar times or in similar areas.
2.Consider physical barriers like rivers and highways that significantly extend travel time beyond straight-line distance when determining the likelihood of a user visiting certain POIs.
3.Match POI categories to their typical visit hours (e.g., nightclubs at night, breakfast spots in the morning) when imputing missing check-ins.
4.Weekday versus weekend behavioral differences should be preserved when imputing missing visits, as users often follow distinct patterns based on day type.

Figure 7: Agent instruction for the macro-level intention completion agent.

Self-verification Mechanism

After making each completion suggestion, verify:
1. Does the sequence formed by the suggested POIs and the check-ins at both ends constitute a logically coherent and contextually consistent activity flow?
2. Is there sufficient time between check-ins for this additional visit?
3. Does this suggestion align with the user's visit patterns?
Only when all responses to the evaluation questions are affirmative can the suggested POI be deemed a valid completion. If any check fails, reconsider your suggestion.

Figure 8: The internal self-verification mechanism of the two location completion agents.

Dataset	#User	#Locations	#Check-ins	#Avg.Visit	#Density
CA	10,162	24,237	456,820	44.95	0.15%
TKY	2,293	15,177	494,807	215.79	1.42%
NYC	1,083	9,989	179,468	165.71	1.66%

Table 2: Statistics of the evaluation datasets.

A.6 Instruction of Macro-Level Intention Completion Agent

We custom-tailor the following agent instruction for our micro-level trajectory completion agent in Figure 7. "All trajectory description" is generated based on T_u^f.

A.7 Self-Verification Mechanism

The internal self verification mechanism of the two location completion agents in Figure 8.

B Experimental Details

B.1 Datasets Details

All datasets are widely used location-based social networking platform. Gowalla-CA is collected from February 2009 and October 2010 and covers a broader geographical area, including locations across both California and Nevada. Foursquare-TKY and Foursquare-NYC are collected from April 2012 to February 2013. Table 2 summarizes key statistics, including the total number of users, location points, and check-in records, along with the average number of locations visited per user and the sparsity of each dataset. Following the preprocessing protocols adopted in prior studies (Qin et al., 2023b; Zuo and Zhang, 2024; Yan et al., 2023), we removed locations with fewer than five user check-ins and excluded users with fewer than five check-in records. After filtering, each dataset was split chronologically into training, validation, and test sets using an 8:1:1 ratio.

B.2 Baselines Details

To comprehensively evaluate the effectiveness of our proposed method, LaMDA, we compare it against a diverse set of baselines from four distinct perspectives: (c1) widely adopted graph-based techniques used in the location prediction; (c2) sequential-based location prediction methods that incorporate spatio-temporal information; (c3) complex graph-based location prediction methods; and (c4) diffusion-based location prediction approaches.

- (c_1) SR-GNN (Wu et al., 2019): This method transforms session sequences into graph-structured representations and leverages gated graph neural networks to capture complex item transitions, enabling accurate session-level preference modeling.
- (c_1) LightGCN (He et al., 2020): It is one of the most representative graph-based collaborative filtering methods that simplifies traditional GCNs by removing feature transformation and nonlinear activation for recommendation.
- (c_1) DGCF (Wang et al., 2020): This model disentangles user-item representations into multiple intent-specific subspaces by constructing and refining intent-aware interaction graphs, effectively capturing the diversity of user intents.
- (c_2) LSTPM (Sun et al., 2020): It integrates a context-aware nonlocal module with a geo-dilated recurrent structure to capture users' global behavioral patterns and localized spatial dynamics across location sequences.
- (c_2) STAN (Luo et al., 2021): This model introduces a spatiotemporal bi-attention mechanism that captures fine-grained dependencies among non-adjacent and non-consecutive check-ins.
- (c_2) GTR (Li et al., 2025): This model maps user sequences into hyperbolic space and captures user preferences through the Mamba-based State Space Model block.
- (c_3) SGRec (Li et al., 2021a): A location prediction model that transforms sequential location data into graph-structured representations by integrating cross-sequence collaborative signals and category-level transitions.
- (c_3) DRAN (Wang et al., 2022a): This model disentangles location representations into transition-based and distance-based components via dual relational graphs, enabling fine-grained modeling of user preferences through aspect-aware attention mechanisms.
- (c_3) STHGCN (Yan et al., 2023): It builds a spatio-temporal hypergraph based on user check-ins and trajectory-level associations and employs a hypergraph transformer to encode intra- and inter-user trajectory dependencies jointly.

- (c_4) Diff-POI (Qin et al., 2023b): This method encodes user check-in behaviors into dual graphs and employs a diffusion-based generative framework to capture future location preferences.
- (c_4) Diff-DGMN (Zuo and Zhang, 2024): This method constructs a dual-view location graph to model personalized mobility patterns and leverages a diffusion-based preference sampler to bridge the representation–interest gap.

Note that we do not use TAU (Zhuang et al., 2024) as a baseline because it does not have public source code. We also exclude comparisons with LLM4POI(Li et al., 2024a) and its derivatives (Liu et al., 2025b), as they reformulate the location prediction task as a question-answering (QA) problem. Their objective is to generate a single best recommendation rather than a diverse top-k list, with a focus on optimizing single-point prediction accuracy. Additionally, our objective is to leverage LLMs for data augmentation while maintaining lightweight deep learning models for location prediction, enabling efficient training and inference. Fine-tuning LLM-based methods entails fundamentally different computational costs. Therefore, we focus on baselines with comparable training and inference mechanisms.

B.3 Models Size

In our preliminary study (Figure 1), we utilized ChatGPT-5.2[7] to establish the upper bound of LLM spatial reasoning capabilities. The failure of such a state-of-the-art model to interpret raw coordinates confirms that this limitation is intrinsic to the coordinate representation modality rather than a deficiency of specific model architectures.

To demonstrate the effectiveness of LaMDA across different architectures and model scales, and to account for inference cost considerations, we employ two large language models as agent backbones: Deepseek-V3 (Liu et al., 2024a), which contains 671 billion parameters, and Qwen3-Turbo (Yang et al., 2025), a smaller model with fewer than 72 billion parameters. Moreover, we conduct experiments with five state-of-the-art LLMs (Claude Sonnet 4.5[8], Grok 4.1[9], Qwen3-max (Yang et al., 2025), Deepseek-R1 (Guo et al., 2025), Gemini2.5 Pro[10]) to examine their understanding of raw geographic coordinates versus natural language location representations.

[7] https://chatgpt.com
[8] https://claude.ai/new
[9] https://grok.com
[10] https://gemini.google.com

B.4 Implementation Details

All experiments were conducted using an NVIDIA A6000 GPU with 48GB of memory in a PyTorch environment. During the location completion phase, each trajectory was augmented with locations amounting to 10% of its original length for micro-level trajectory completion. For macro-level intention completion, locations were supplemented for each target user, comprising 30% of all available check-in data. In the training phase, the hidden dimension was set to 64 and the learning rate to 0.01. All models were optimized using the Adam optimizer. Following prior work (Wang et al., 2022a; Zuo and Zhang, 2024; Qin et al., 2023b), we set the weight decay parameter η for L2 regularization uniformly to 0.01. The threshold for constructing the spatial distance location graph was set to 1 km, and the dimensionality of the spatial interval embedding matrix was set to 256. Based on Deepseek-V3 inference, we set the values of α to 0.6, 0.5, and 0.9 for the CA, TKY, and NYC datasets, respectively. Under Qwen3-Turbo inference, α is set to 0.6, 0.6, and 0.3 for the CA, TKY, and NYC datasets, respectively. We set the values of dp to 0.4, 0.3, and 0.3 for the CA, TKY, and NYC datasets.

C Results Details

This section describes our experimental design and presents extensive evaluations conducted on three practical datasets to answer eight key research questions:

RQ1: How does our proposed LaMDA model perform compared to state-of-the-art location prediction methods? (in Sections 5.2.1)

RQ2: How is the individual contribution of each key component within the LaMDA framework to overall model performance? (in Sections 5.2.2)

RQ3: How does the location completion ratio affect model performance across the two proposed agents: micro-level trajectory completion and macro-level intention completion? (in Sections 5.2.3)

RQ4: To what extent is the LaMDA model sensitive to hyperparameter variations?

RQ5: How is the capability of existing mainstream LLMs in understanding geographic information?

Da.	Metric	Recall@5				Recall@10				NDCG@5			
	h,dp	0.1	0.2	0.3	0.4	0.1	0.2	0.3	0.4	0.1	0.2	0.3	0.4
		Deepseek-V3											
	32	0.2931	0.3020	0.3053	0.3012	0.3273	0.3370	0.3395	0.3350	0.2590	0.2691	0.2717	0.2686
	64	0.3097	0.3280	0.3331	**0.3361**	0.3431	0.3602	0.3646	**0.3679**	0.2753	0.2946	0.3013	**0.3041**
	96	0.3148	0.3267	0.3381	0.3407	0.3483	0.3589	0.3703	0.3712	0.2805	0.2925	0.3072	0.3102
CA	128	0.3073	0.3265	0.3383	0.3431	0.3421	0.3579	0.3704	0.3734	0.2739	0.2936	0.3059	0.3124
		Qwen3-Turbo											
	32	0.2768	0.2910	0.3006	0.2924	0.3136	0.3260	0.3346	0.3256	0.2418	0.2569	0.2677	0.2603
	64	0.3065	0.3154	0.3280	**0.3302**	0.3416	0.3492	0.3622	**0.3630**	0.2715	0.2818	0.2960	**0.2979**
	96	0.3075	0.3230	0.3337	0.3376	0.3408	0.3547	0.3660	0.3698	0.2732	0.2902	0.3020	0.3061
	128	0.3086	0.3254	0.3355	0.3390	0.3431	0.3591	0.3673	0.3704	0.2720	0.2909	0.3043	0.3087
	h,dp	0.1	0.2	0.3	0.4	0.1	0.2	0.3	0.4	0.1	0.2	0.3	0.4
		Deepseek-V3											
	32	0.6123	0.6172	0.6142	0.6090	0.6443	0.6476	0.6415	0.6382	0.5754	0.5828	0.5820	0.5760
	64	0.6265	0.6434	**0.6500**	0.6431	0.6572	0.6731	**0.6773**	0.6703	0.5900	0.6098	**0.6186**	0.6118
	96	0.6335	0.6494	0.6585	0.6574	0.6643	0.6770	0.6845	0.6825	0.5969	0.6153	0.6275	0.6279
TKY	128	0.6351	0.6504	0.6569	0.6596	0.6657	0.6781	0.6835	0.6844	0.5992	0.6166	0.6258	0.6294
		Qwen3-Turbo											
	32	0.6082	0.6093	0.6101	0.6052	0.6406	0.6404	0.6398	0.6338	0.5711	0.5738	0.5768	0.5711
	64	0.6201	0.6389	**0.6468**	0.6413	0.6522	0.6681	**0.6738**	0.6679	0.5802	0.6046	**0.6149**	0.6102
	96	0.6273	0.6442	0.6543	0.6510	0.6575	0.6726	0.6803	0.6775	0.5901	0.6093	0.6217	0.6217
	128	0.6261	0.6434	0.6528	0.6528	0.6590	0.6719	0.6795	0.6789	0.5889	0.6092	0.6214	0.6229
	h,dp	0.1	0.2	0.3	0.4	0.1	0.2	0.3	0.4	0.1	0.2	0.3	0.4
		Deepseek-V3											
	32	0.6622	0.6672	0.6706	0.6686	0.6765	0.6820	0.6847	0.6814	0.6403	0.6479	0.6546	0.6537
	64	0.6756	0.6868	**0.6955**	0.6946	0.6905	0.6997	**0.7060**	0.7055	0.6555	0.6714	**0.6823**	0.6812
	96	0.6884	0.6867	0.7016	0.6994	0.7038	0.7007	0.7113	0.7089	0.6699	0.6725	0.6877	0.6867
NYC	128	0.6907	0.6935	0.6995	0.6993	0.7049	0.7052	0.7112	0.7091	0.6733	0.6786	0.6850	0.6873
		Qwen3-Turbo											
	32	0.6406	0.6588	0.6688	0.6644	0.6624	0.6732	0.6822	0.6782	0.6169	0.6383	0.6537	0.6492
	64	0.6657	0.6797	**0.6869**	0.6852	0.6821	0.6943	**0.6987**	0.6979	0.6447	0.6599	**0.6722**	0.6720
	96	0.6714	0.6873	0.6933	0.6922	0.6888	0.7012	0.7050	0.7039	0.6492	0.6676	0.6767	0.6788
	128	0.6796	0.6866	0.6910	0.6944	0.6956	0.7006	0.7037	0.7051	0.6618	0.6696	0.6762	0.6799

Table 3: Impact of the hidden size h and dropout rate dp on three dataset. With $h = 64$, the best results across the three datasets are highlighted in bold. "Da." is Datasets.

C.1 Hyper-parameters Analysis (RQ4)

Sensitivity Analysis of Hidden Dimensions and Dropout Rate. We investigate the impact of two key hyperparameters, hidden dimension ($h \in \{32, 64, 96, 128\}$) and dropout rate ($dp \in \{0.1, 0.2, 0.3, 0.4\}$), on the performance of our proposed LaMDA model. As shown in Table 3, the performance of LaMDA consistently improves across all three datasets as the hidden dimension h increases. For a fair and direct comparison with the baseline models (which also utilize a hidden size of 64) and to maintain computational efficiency, we set $h = 64$ for all experiments. This allows us to demonstrate the advantages of our LaMDA architecture.

The model achieved optimal performance on the CA dataset with a dropout rate of $dp = 0.4$. In contrast, the best results were obtained with $dp = 0.3$ for the NYC and TKY datasets. We attribute this difference to the datasets' characteristics. The CA dataset is considerably sparser than the NYC and TKY datasets. Consequently, a higher dropout rate on the CA dataset can effectively mitigate overfitting to the limited training data. Conversely, the NYC and TKY datasets contain more abundant training data, making higher dropout rates counterproductive as they discard valuable information and reduce model performance.

Analysis of check-in History Segmentation. To

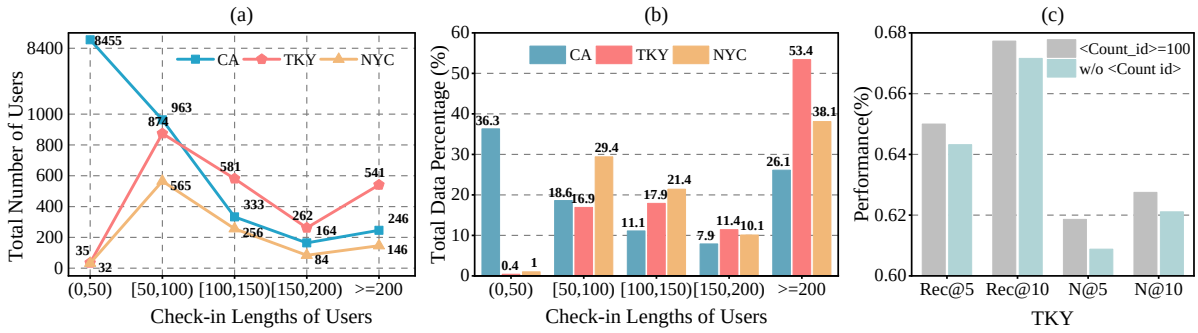

Figure 9: (a) Distribution of the total number of users across different check-in lengths. (b) Distribution of the total data proportion across different check-in lengths. (c) Performance comparison between using Check-in History Segmentation and not using it.

better understand user behavior across different datasets, we analyzed the distribution of check-in lengths within CA, TKY, and NYC training sets. As illustrated in Figure 9(a), users with extensive check-in histories (i.e., a check-in count ≥ 200) represent a small fraction of the total user base in each dataset. However, Figure 9(b) reveals that these few users contribute a disproportionately large volume of the total check-in data. This disparity presents two significant challenges for the LLM during inference: the computational burden of processing long-form text and the potential adverse effects of long-term temporal spans, during which user interests may evolve or shift. We conducted a comparative performance analysis to validate our check-in history segmentation strategy. Due to inference costs, we focused on the TKY dataset. We compared a model using segmented check-in histories with a maximum length of $\varphi = 100$ against a baseline model without segmentation (w/o <Count id>). The results, shown in Figure 9(c), demonstrate the superior performance of the segmented approach. By partitioning the data, the LLM can concentrate on the user's recent behavioral patterns within the current segment. Simultaneously, it retains the ability to access the broader context of their long-term preferences via the comprehensive visited POI pool, leading to more accurate and efficient intention completion.

Balance Coefficient α Sensitivity. We evaluate the impact of the balance coefficient α on LaMDA performance across three datasets using two location completion agents powered by deepseek-v3 and Qwen3-turbo inference models. As shown in Table 4, the choice of LLM in each completion agent influences the optimal α value. Overall, LaMDA's performance declines significantly when $\alpha \leq 0.2$. This finding has two important implications: first, insufficient understanding of user behavior sequences substantially degrades model performance; second, our location completion agents generate location that form logically coherent and contextually consistent behavioral patterns with users' original check-in data. When $\alpha \geq 0.2$, LaMDA performance stabilizes and remains consistent.

C.2 Analysis of Geographic Information Understanding in Mainstream LLMs (RQ5)

We evaluated the geographic information comprehension capabilities of several mainstream LLMs (Claude Sonnet 4.5, Grok 4.1, Qwen3-max, Deepseek-R1, Gemini2.5 Pro). Our assessment focused on two key aspects: **first**, whether these models can accurately convert latitude and longitude coordinates into specific geographic locations, and **second**, which format of geographic information representation enables the models to produce reasoning that most closely aligns with real-world scenarios. Specifically, the evaluation results for ChatGPT-5.2 are presented in Figures 1(c) and 1(d). For models that generated excessively lengthy outputs, we extracted only the essential reasoning components for analysis.

C.2.1 Analysis of Specific Geographic Coordinate Information

LLMs' capability in understanding specific geographic coordinate information. As illustrated in Figure 10, we provide LLMs with a geographic coordinate-based template to evaluate their ability to identify POI locations from precise geographic coordinates. The results from these leading LLMs demonstrate significant limitations in interpreting

Dataset	LLMs	α	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
CA	Deepseek -V3	Recall@2	0.2910	0.2961	0.2968	**0.2971**	0.2949	0.2967	0.2975	0.2916	0.2969
		Recall@10	0.3607	0.3648	0.3666	0.3678	0.3664	**0.3679**	0.3653	0.3640	0.3666
		NDCG@2	0.2805	0.2853	0.2858	0.2864	0.2844	0.2865	**0.2873**	0.2806	0.2862
		NDCG@10	0.3076	0.3122	0.3133	0.3140	0.3123	**0.3143**	0.3138	0.3089	0.3136
	Qwen3 -Turbo	Recall@2	0.2865	0.2905	0.2909	0.2904	0.2895	0.2918	0.2906	**0.2929**	0.2918
		Recall@10	0.3561	0.3594	0.3616	0.3599	0.3601	**0.3630**	0.3604	0.3618	0.3626
		NDCG@2	0.2759	0.2799	0.2811	0.2800	0.2793	0.2816	0.2800	**0.2823**	0.2806
		NDCG@10	0.3031	0.3068	0.3088	0.3074	0.3070	**0.3095**	0.3073	0.3092	0.3083
TKY	Deepseek -V3	Recall@2	0.6090	0.6083	0.6124	0.6134	**0.6136**	0.6130	0.6129	0.6119	0.6133
		Recall@10	0.6723	0.6732	0.6751	0.6758	0.6773	0.6760	0.6764	**0.6780**	0.6771
		NDCG@2	0.5983	0.5959	0.6010	0.6018	**0.6021**	0.6013	0.6005	0.6006	0.6015
		NDCG@10	0.6235	0.6217	0.6259	0.6267	**0.6275**	0.6264	0.6264	0.6265	0.6274
	Qwen3 -Turbo	Recall@2	0.6043	0.6024	0.6051	0.6067	0.6063	**0.6108**	0.6025	0.6093	0.6082
		Recall@10	0.6678	0.6684	0.6683	0.6709	**0.6742**	0.6738	0.6686	0.6728	0.6724
		NDCG@2	0.5931	0.5914	0.5930	0.5949	0.5949	**0.5987**	0.5909	0.5983	0.5973
		NDCG@10	0.6184	0.6176	0.6181	0.6204	0.6220	**0.6237**	0.6172	0.6236	0.6230
NYC	Deepseek -V3	Recall@2	0.6617	0.6687	0.6687	0.6762	0.6764	0.6770	0.6771	0.6744	**0.6798**
		Recall@10	0.6935	0.6965	0.7000	0.7052	0.7041	0.7031	0.7059	0.7041	**0.7060**
		NDCG@2	0.6555	0.6633	0.6625	0.6703	0.6709	0.6712	0.6714	0.6697	**0.6752**
		NDCG@10	0.6682	0.6744	0.6751	0.6818	0.6819	0.6818	0.6829	0.6814	**0.6856**
	Qwen3 -Turbo	Recall@2	0.6660	0.6698	**0.6725**	0.6721	0.6713	0.6714	0.6710	0.6716	0.6714
		Recall@10	0.6960	0.6986	0.6986	0.6984	**0.6994**	0.6993	0.6979	0.6990	0.6980
		NDCG@2	0.6612	0.6653	**0.6679**	0.6671	0.6667	0.6663	0.6656	0.6670	0.6669
		NDCG@10	0.6731	0.6767	**0.6783**	0.6774	0.6779	0.6774	0.6764	0.6779	0.6776

Table 4: Impact of the balance coefficient α on three dataset.

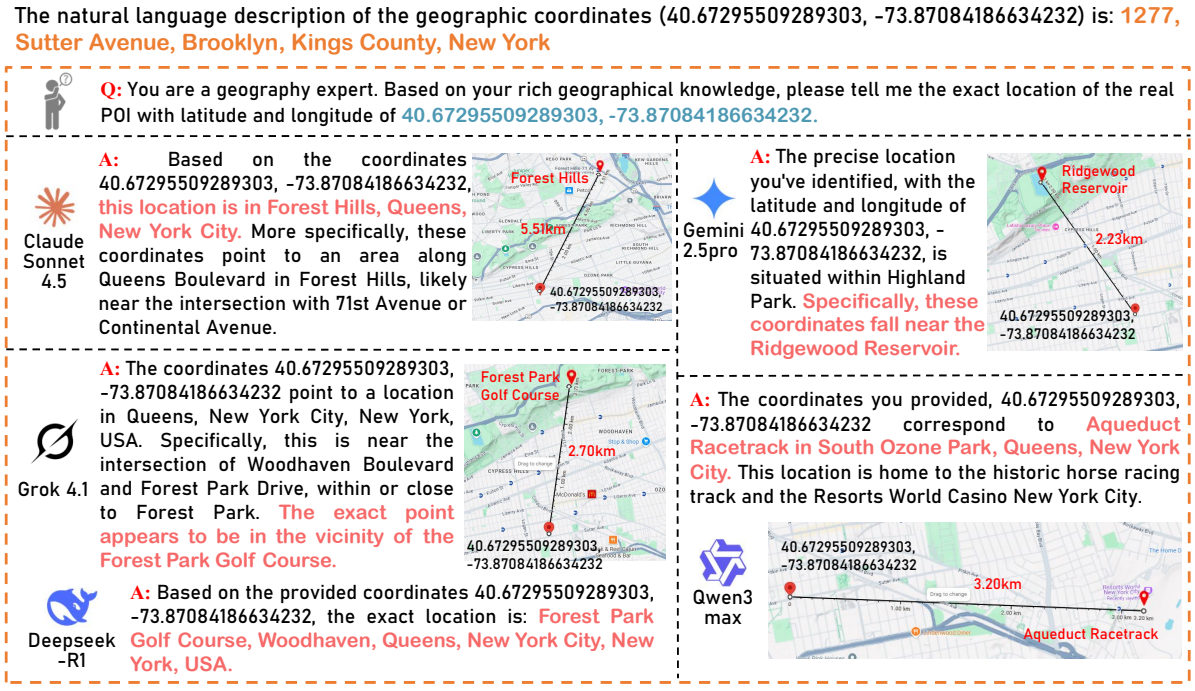

Figure 10: Verification of whether various LLMs can identify specific locations based on geographic coordinates.

Geographic Coordinates of Locations

Q: Latitude, longitude 40.75061688442517, - 73.96051875252135 and latitude, longitude 40.74722117616364, -73.9566698338901 is 0.5 km away. If the physical distance from the straight line is too large due to a river or other reasons, please point out.

Claude Sonnet 4.5

A: There are no rivers, major highways, or other significant physical barriers between these locations that would cause a substantial detour. The main "obstacle" is simply Manhattan's street grid system, which requires following streets rather than walking in a perfectly straight line, but this creates only a minor increase in distance.

Grok 4.1

A: The straight-line distance (0.46 km) aligns with the street grid in Manhattan, where walking or driving distance is typically close to the straight-line distance due to the grid layout. There is no significant river or major obstacle between these points, as they are both on the same side of Manhattan, west of the East River. Walking distance, following streets (e.g., along 1st Ave or 2nd Ave), is approximately 0.5–0.6 km, depending on the exact route, which is still close to the straight-line distance. No major detours are required due to rivers, parks, or other barriers in this area.

Qwen3 max

A: The coordinates provided are in New York City. Using a map or a geographic information system (GIS), we can see that: 1. The first point (40.75061688442517, -73.96051875252135) is near the intersection of West 8th Street and 6th Avenue. 2. The second point (40.74722117616364, -73.9566698338901) is near the intersection of West 4th Street and MacDougal Street. There are no major rivers or significant obstacles between these two points. The area is primarily urban with streets and buildings, but no large natural barriers. Therefore, the physical distance from the straight line is not significantly affected by any rivers or other reasons.

Deepseek -R1

A: No river or major waterbody exists between these points. They are both in central Manhattan: Point A: Near the Empire State Building (5th Ave & 30th St). Point B: Near Madison Square Park (Park Ave S & 28th St). The area is entirely urban (streets/buildings). The closest rivers (Hudson to the west, East River to the east) are >1.5 km away and do not intersect the straight-line path. The straight-line distance is 0.5 km with no river or physical barrier between the points.

Gemini 2.5pro

A: The calculated straight-line distance, also known as the great-circle distance, between the two points you provided is indeed approximately 0.5 kilometers. Upon examining the map, there are no significant physical barriers such as a river that would dramatically increase the walking or driving distance compared to the direct line. The area is a dense urban environment with a standard street grid. The path between the two locations is straightforward via local roads.

Figure 11: Comparative analysis of LLMs' output results of geographic coordinates of locations.

Natural Language Descriptions of Locations

Q: Franklin D.Roosevelt four freedoms State Park and 4610 center Blvd, Long Island City, NY is 0.5 km away. If the physical distance from the straight line is too large due to a river or other reasons, please point.

Claude Sonnet 4.5

A: While 0.5 km (approximately 0.3 miles) might be the straight-line distance between the two locations, there is indeed a significant physical barrier: the East River separates Roosevelt Island from Long Island City, Queens. you're absolutely correct to point out this issue. The 0.5 km straight-line distance is misleading because the East River creates a significant barrier that makes the actual travel route much longer and requires water transportation or a detour through Manhattan.

Grok 4.1

A: The straight-line distance between Franklin D. Roosevelt Four Freedoms State Park on Roosevelt Island and 4610 Center Blvd, Long Island City, NY, is approximately 0.5 km (0.3 miles). However, the East River separates Roosevelt Island from Long Island City, making direct travel impossible without using a bridge, tram, or ferry.

Qwen3 max

A: The straight-line distance between these two points is approximately 0.5 km (or 0.31 miles), as you mentioned. However, the physical distance to travel between these two locations can be significantly longer due to the East River, which separates Roosevelt Island from Long Island City. You would need to use a bridge or take public transportation, such as the F train, to cross the river. The actual travel distance and time will be greater than the straight-line distance.

Deepseek -R1

A: The East River lies between these two points. Roosevelt Island and Long Island City (Queens) are separated by the river, with no direct land connection at these locations. The physical distance is much larger than the straight-line 0.5 km due to the East River barrier. Travel requires a detour over bridges or a ferry, making the actual journey longer and more complex.

Gemini 2.5pro

A: You are correct! The straight-line distance between Franklin D. Roosevelt Four Freedoms State Park and 4610 Center Blvd is indeed very short. However, the two locations are separated by the East River. This means you cannot travel in a straight line between them. To get from one to the other, you would need to take a bridge, which makes the actual travel distance much longer than 0.5 km.

Figure 12: Comparative analysis of LLMs' output results of natural language descriptions of locations.

latitude and longitude information. Specifically, Claude Sonnet 4.5's predictions deviated from the actual coordinates by approximately 6 kilometers in straight-line distance, while Gemini 2.5 Pro and ChatGPT-5.2 could only provide approximate district-level localization. However, user mobility patterns exhibit spatial clustering characteristics, as individuals tend to visit locations in proximity to their current position. Consequently, providing only raw geographic coordinates to LLMs fails to capture the exact spatial relationships underlying user check-in behaviors. This approach can subsequently mislead the models' reasoning and predictive judgments. This finding explains why incorporating specific geographic coordinates into the LLM4POI (Li et al., 2024a) input, as shown in Figure 1(b), resulted in negligible performance improvements.

C.2.2 Analysis of Geographic Location Information

What format of geographic location information enables LLMs' reasoning to better align with real-world scenarios? We further investigated whether the reasoning processes of leading LLMs align with physical reality by asking them to assess the distance between two locations represented by either geographic coordinates or natural language descriptions. The geographic coordinates 40.75061688442517, -73.96051875252135 correspond to Franklin D. Roosevelt Four Freedoms State Park, while coordinates 40.74722117616364, -73.9566698338901 represent 610 Center Blvd, Long Island City, NY. Notably, these two locations are separated by the East River, creating a significant physical barrier that substantially increases the actual travel distance beyond the straight-line distance, with access only possible via bridges or ferry crossings.

As demonstrated in Figure 11, when provided with geographic coordinates, LLMs often fail to recognize the physical barriers between locations, resulting in unrealistic assessments that are inconsistent with real-world scenarios. We attribute this limitation to the models' inability to accurately interpret latitude and longitude information, as evidenced in Figure 10. As illustrated in Figure 12, we observed a remarkable improvement when converting coordinate information into natural language descriptions: the models demonstrated accurate location understanding and realistic spatial reasoning, successfully identifying physical barriers between locations. This approach enhances the accuracy of LLMs in understanding user check-in behaviors and performing location completion tasks.

D Detailed Related Works

D.1 CF for Next Location Prediction

Next location prediction aims to suggest relevant locations by modeling users' mobility patterns and preferences. Within the scope of traditional collaborative filtering, this section categorizes the primary methodologies into two principal types: sequence-based methods and graph-based methods.

D.1.1 Sequence-based methods

Sequence-based location prediction methods treat user check-in data as sequential patterns, incorporating temporal and spatial context to capture user visit preferences and behavioral dynamics(Zhang et al., 2022; Long et al., 2023; Huang et al., 2024c,a). TMCA (Li et al., 2018) proposed an encoder-decoder architecture based on LSTM that simultaneously captures temporal dependencies in check-in sequences and hierarchically weighs contextual factors at micro and macro levels for enhanced next location prediction. LSTPM (Sun et al., 2020) combines nonlocal networks for capturing long-term user preferences with geo-dilated recurrent neural networks to model short-term preferences. ST-PEGD (Lan et al., 2023) integrates a gated-deep network with a position-extended algorithm, employing binary gates to capture long-term dependencies and enhancing RNN interactions to model robust short-term spatio-temporal preferences. GeoSAN (Lian et al., 2020) integrates a self-attention-based geography encoder to represent GPS points as hierarchical quad key grids, capturing spatial proximity to enhance location prediction accuracy. DCSPR (Sun et al., 2024) integrates intra-city region and function mining to capture each city's unique geographical and cultural characteristics while employing a dual-target transfer strategy to share relevant functional relations across cities, enhancing sequential location prediction through a region- and function-aware neural network.

D.1.2 Graph-based methods

Graph-based location prediction methods construct various graph structures from user check-in data to model the complex relationships and dependencies between user-visiting behaviors(Yang et al., 2022; Yan et al., 2023; Fu et al., 2024; An et al.,

2024). GEAPR (Li et al., 2021b) integrates a graph-enhanced attention network to deliver interpretable location predictions by combining structural context, neighbor impact, user attributes, and geolocation influence. GSTN (Wang et al., 2022b) integrates a novel graph-based spatial dependency (GSD) module to capture complex geographical influences through graph embedding alongside a Time-LSTM network to model user-specific temporal dependencies, effectively combining spatial and temporal factors. DisenPOI (Qin et al., 2023a) introduces a dual-graph framework for POI recommendation, distinctly separating sequential and geographical influences through tailored GNNs and self-supervised contrastive learning, enhancing recommendation accuracy and interpretability. MMPOI (Xu et al., 2024) integrates multi-modal content (textual and visual) with user check-in sequences, constructing a multi-modal trajectory flow graph (MTFG) and a geographic trajectory flow graph (GTFG) to capture semantic and geographical patterns.

D.2 LLMs in Spatial-temporal Tasks

LLMs have shown significant potential in analyzing spatio-temporal data by processing rich textual representations of location and time(Yu et al., 2024; Liu et al., 2024b). Their ability to interpret contextual and semantic information enables effective modeling of user movement patterns and temporal trends. However, limitations in handling precise geographic coordinates highlight the need for innovative approaches to integrate spatial reasoning with semantic understanding. GATGPT (Chen et al., 2023) leverages the LLMs' robust temporal pattern recognition capabilities to integrate pre-trained LLMs with a graph attention mechanism, addressing spatiotemporal imputation. GeoLLM (Manvi et al., 2023) extracts geospatial knowledge from LLMs using OpenStreetMap-enhanced prompts for population and economic prediction tasks. UrbanGPT (Li et al., 2024b) integrates a spatio-temporal dependency encoder with an instruction-tuning paradigm, enabling LLMs to effectively capture complex temporal and spatial interdependencies for spatio-temporal prediction tasks. ST-LLM+ (Liu et al., 2025a) integrates graph-enhanced attention within LLMs and combines a proximity-based adjacency matrix with a partially frozen graph attention (PFGA) mechanism to capture complex spatio-temporal dependencies in traffic networks.

D.3 LLMs for Next Location Prediction

LLM-based approaches enhance location prediction accuracy by understanding location semantics and user behavior contexts (Wu et al., 2024; Liu et al., 2025b; Wang et al., 2025). LLMmove (Feng et al., 2024) introduces a prompting-based framework that leverages LLMs to integrate long-term spatial preferences, current movement patterns, geographical proximity, and sequential transitions for zero-shot location prediction. LLM4POI (Li et al., 2024a) transforms the next location prediction into a QA task using LLMs, converting heterogeneous LBSN data into structured prompts to leverage commonsense knowledge while preserving contextual information. LLMGPR (Long et al., 2024) integrates semantic-enhanced location tokens with QLORA-based adapters to capture group dynamics, employing an aggregation mechanism for individual member fusion and self-supervised sequence purpose prediction to address group-level data sparsity. POI-Enhancer (Cheng et al., 2025) employs specialized prompts to extract semantic knowledge from LLMs and integrates this information into location representations through dual feature alignment and cross-attention fusion.

Although these influential location prediction approaches have demonstrated outstanding performance, they universally assume user check-in behavior to be comprehensive, neglecting the inherently subjective nature of user check-in decisions.