

Generative Next POI Recommendation with Semantic ID

Dongsheng Wang

University of Electronic Science and
Technology of China
Chengdu, China
202411081626@std.uestc.edu.cn

Yuxi Huang

University of Electronic Science and
Technology of China
Chengdu, China
yuxi.h.cs@gmail.com

Shen Gao*

University of Electronic Science and
Technology of China
Chengdu, China
shengao@uestc.edu.cn

Yifan Wang

University of Electronic Science and
Technology of China
Chengdu, China
202422081324@std.uestc.edu.cn

Chengrui Huang

University of Electronic Science and
Technology of China
Chengdu, China
2021080101014@std.uestc.edu.cn

Shuo Shang*

University of Electronic Science and
Technology of China
Chengdu, China
jedi.shang@gmail.com

Abstract

Point-of-interest (POI) recommendation systems aim to predict the next destinations of user based on their preferences and historical check-ins. Existing generative POI recommendation methods usually employ random numeric IDs for POIs, limiting the ability to model semantic relationships between similar locations. In this paper, we propose Generative Next POI Recommendation with Semantic ID (GNPR-SID), an LLM-based POI recommendation model with a novel semantic POI ID (SID) representation method that enhances the semantic understanding of POI modeling. There are two key components in our GNPR-SID: (1) a Semantic ID Construction module that generates semantically rich POI IDs based on semantic and collaborative features, and (2) a Generative POI Recommendation module that fine-tunes LLMs to predict the next POI using these semantic IDs. By incorporating user interaction patterns and POI semantic features into the semantic ID generation, our method improves the recommendation accuracy and generalization of the model. To construct semantically related SIDs, we propose a POI quantization method based on residual quantized variational autoencoder, which maps POIs into a discrete semantic space. We also propose a diversity loss to ensure that SIDs are uniformly distributed across the semantic space. Extensive experiments on three benchmark datasets demonstrate that GNPR-SID substantially outperforms state-of-the-art methods, achieving up to 16% improvement in recommendation accuracy¹.

CCS Concepts

- Information systems → Location based services; Recommender systems;
- Human-centered computing → Social recommendation.

^{*}Shen Gao and Shuo Shang are corresponding authors.

¹Code is available at <https://github.com/wds1996/GNPR-SID>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

KDD '25, Toronto, ON, Canada

© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 979-8-4007-1454-2/2025/08
<https://doi.org/10.1145/3711896.3736981>

Keywords

Point-of-Interest, Recommendation, Generative model, LLM

ACM Reference Format:

Dongsheng Wang, Yuxi Huang, Shen Gao, Yifan Wang, Chengrui Huang, and Shuo Shang. 2025. Generative Next POI Recommendation with Semantic ID. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2 (KDD '25), August 3–7, 2025, Toronto, ON, Canada*. ACM, New York, NY, USA, 11 pages. <https://doi.org/10.1145/3711896.3736981>

1 Introduction

Location-based social networks, such as Foursquare and Google Maps, are being used more and more frequently in modern life, with a large number of users sharing their check-in records on these platforms via mobile devices, such as once-trip or once-restaurant. Thus, point-of-interest (POI) recommendation [16, 24, 28] became a popular research task, which aims to provide location services to users based on their preferences and context information. The next POI recommendation [4, 19, 49], is a variant of POI recommendation tasks, focuses on the user's historical check-in trajectory, incorporating sequential transitions and spatial-temporal context information to predict the next visit.

Recently, with the success of Large Language Models (LLMs) in various domains [1, 3, 7], LLM-based generative recommendation has emerged as a new paradigm for next POI recommendation [9, 18, 33, 40], which directly generates the next POI IDs for a given user query. Compared to traditional POI recommendation methods, these models can be trained end-to-end, which avoids using the embedding vector for each POI and instead allows the model to directly generate the POI ID. This improves the generalization ability of model and reduces the space for storing the vectors of each POI.

Existing LLM-based POI recommendation methods [18, 40] rely predominantly on Random ID (RID), where each POI is treated as a random unique numeric representation, and user historical interactions are modeled as sequences of these RIDs. Traditional methods based on POI embedding vectors can store rich semantic information about POIs within the embedding vectors. However, existing LLM-based methods only use the randomly generated RID as input, and these random numbers do not capture any semantic information. As a result, existing LLM-based methods struggle to effectively model POI semantic information. It is also evident

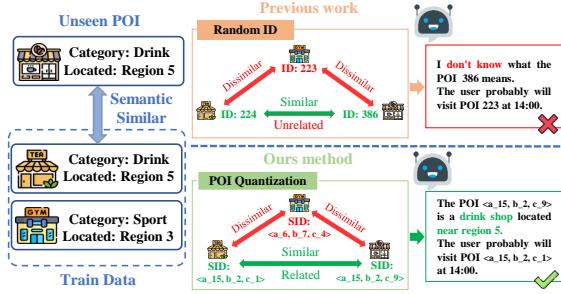


Figure 1: Comparison of random ID and semantic ID: Semantic ID can captures semantic associations between similar POIs which can facilitate LLM to model the user preference.

that these methods also exhibit poor generalization ability for POI information outside the dataset.

Therefore, designing an effective POI ID representation method is crucial for improving LLM-based POI recommendation models. A well-designed POI ID representation method should have the following characteristics: (1) The POI ID should reflect the semantic information of the POI, with similar POIs having similar IDs; (2) It should represent a vast number of POIs using a smaller ID space, without causing ID conflicts. For example, in a nearby neighborhood, there is a coffee shop and a tea shop. As adjacent stores sell similar beverages, both designed to refresh customers, these two shops should have similar IDs. We show a comparison between existing LLM-based POI recommendation methods and our proposed GNPR-SID in Figure 1. For instance, if the coffee shop's ID is $\langle a_15 \rangle \langle b_2 \rangle \langle c_9 \rangle$, a good ID for the tea shop might be $\langle a_15 \rangle \langle b_2 \rangle \langle c_1 \rangle$. This ID representation reflects the similarity between the two POIs, which not only helps the LLM better understand the semantics of the POIs but also allows for better generalization when a new bubble tea shop is added. By assigning an ID like $\langle a_15 \rangle \langle b_2 \rangle \langle c_x \rangle$ to the bubble tea shop, the system can help the LLM generalize to the newly introduced POI more effectively. However, the existing methods assign a random ID (e.g., 386) for the newly added POI which cannot reflect any semantic similarity between the other two shops (e.g., 223 and 224).

Therefore, in this paper, we propose **Generative Next POI Recommendation with Semantic ID** (GNPR-SID), which introduces an LLM-based POI recommendation model with a novel semantic POI ID representation module that enhances the semantic understanding of POI modeling. There are two main modules in GNPR-SID: (i) **Semantic ID construction module**, which can generate a semantic ID for each POI based on common POI features (e.g., location, category, etc.), where the semantic ID consists of multiple numbers and POIs with similar semantics share similar ID prefixes. (ii) **Generative POI recommendation module**, which uses the semantic POI ID as input and fine-tunes the LLM to predict the next POI by incorporating the knowledge and reasoning ability within the LLM.

Since POI features are typically represented by numerical data, which are challenging for LLMs primarily designed for text understanding, we propose a **POI semantic representation** method.

This method transforms complex numerical features into semantic representations that are more interpretable for LLMs. Furthermore, our approach not only captures the intrinsic semantic information of POIs but also integrates collaborative filtering signals and user visiting patterns into the POI semantic ID, enriching the representation with both contextual and behavioral insights. Next, we propose a **hierarchical POI quantization** method based on residual quantized variational autoencoder. This process maps POI embeddings into a discrete semantic space, where each POI is represented by a sequence of codeword indices derived from multi-layer codebooks. This quantization ensures that semantically similar POIs share similar SIDs, preserving their relationships in a structured and compact form. To achieve this, we propose the diversity loss, which promotes the distinctiveness of codewords within each codebook layer.

Finally, we use the POI SID representation as input to the LLM, and end-to-end to train the model to directly predict the next POI based on the user historical check-in data. Compared to existing LLM-based next POI recommendation methods, our approach helps the model comprehensively understand the semantics of each POI and the deeper relationships between POIs. By leveraging the rich commonsense knowledge and reasoning capabilities inherent in LLMs, our method enables a better understanding of user intent, leading to more accurate next POI recommendations. To demonstrate the effectiveness of GNPR-SID, we conduct extensive experiments on three real-world benchmark datasets: Foursquare-NYC [44], Foursquare-TKY [44], and GowallaCA [5]. Experimental results demonstrate that GNPR-SID consistently outperforms state-of-the-art approaches, with accuracy improvements reaching up to 16% on these benchmark datasets. Furthermore, to validate that the generated SIDs effectively capture the semantic similarity between POIs, we conduct semantic similarity analysis and visualization experiments on the SIDs. The experiments demonstrate that semantically similar POIs share more common SID prefixes.

Our contributions are summarized as follows:

- We first introduce the **Generative Next POI Recommendation with Semantic ID** (GNPR-SID), which fine-tunes the LLM to comprehensively understand the spatial-temporal semantics in the POI recommendation task.
- We propose the semantic encoding of POI metadata and then propose a semantic ID construction method that maps POI into semantic ID by POI quantization.
- We train an LLM to directly generate the next POI SID by incorporating the knowledge and reasoning ability of an LLM.
- Extensive experiments on three benchmark datasets demonstrate that GNPR-SID outperforms state-of-the-art methods.

2 Related Work

Early next POI recommendation methods are often framed as a sequential recommendation task, leveraging feature engineering such as personalized Markov chains [4, 14], but suffered from limited scalability and dependency on domain expertise. With the rise of deep learning, RNN-based models [17, 22, 29] were widely adopted. For example, Zhao et al. [49] propose STGN a spatio-temporal gated network, to capture personalized sequential patterns, addressing user long and short-term preferences. Driven by the flexibility of attention mechanisms, Transformer-based models [6, 32, 41, 48]

are surpassing RNNs in next POI recommendation. For example, STAN [26] proposes a spatial-temporal attention mechanism to capture spatial-temporal relevance within POI trajectories. In addition, graph neural networks gained traction for modeling user-POI interactions [20, 21, 38, 39, 45]. For example, STHGCN [43] uses a spatial-temporal hypergraph convolutional network that captures trajectory-level information by learning from both user history and collaborative trajectories.

Recently, LLMs have demonstrated strong capabilities in handling spatial-temporal data and commonsense knowledge, achieving significant success in retrieval tasks [30, 36] and recommendation tasks [27, 34, 37], making them a promising solution for next POI recommendation. LLM-Mob [35] introduce in-context learning to enhance next POI recommendation using historical and contextual trajectories. NextLocLLM [23] uses spatial coordinates to enhance its understanding of spatial relationships between locations, transforming ID prediction into coordinate prediction, and improving transferability and generalization in different datasets. LLM4POI [18] performs supervised fine-tuning of the LLM to transform the next POI recommendation task into a question-answering task, and it achieves the stats-of-the-art performance in several next POI recommendation benchmark datasets. These models rely solely on randomly initialized ID representations or a combination of random numeric IDs with text prompts for POI trajectory input, failing to capture semantic associations between POIs and demonstrating limited cross-dataset generalization capabilities. However, the semantic association between POIs is not only reflected in spatial coordinates.

To address the challenge of incorporating non-textual knowledge into LLMs, multimodal large language models (MLLMs) [2, 46] provide a perspective. OneLLM [13] maps signals from any modality into embeddings using a frozen unified encoder and an MOE-based Universal Projection Module, then inputs them into LLMs for downstream tasks. However, POI representation requires modeling temporal sequences and spatial location information, most existing MLLMs cannot effectively capture POI semantic information. Therefore, exploring the application of MLLM methods to POI recommendation is also a promising research direction worth investigating.

3 Problem Definition

Let $U = \{u_1, u_2, \dots, u_M\}$ and $P = \{p_1, p_2, \dots, p_N\}$ represent the set of M users and N POIs, respectively. Each POI is characterized by a tuple $p_i = (c, lon, lat, u_{p_i}, t_{p_i})$, where c represents the category (e.g., Home, Subway and Food Shop), lon and lat denotes the geographical coordinates (longitude and latitude), u_{p_i} refers to the set of check-in users, and t_{p_i} indicates the set of timestamps when users visited. For a user $u \in U$, we can obtain the recent visited POIs $P_u = \{p_1, p_2, \dots, p_i\}$, and the corresponding check-in timestamps for each POI $T_u = \{t_1, t_2, \dots, t_i\}$. In the next POI recommendation task, we use the recent check-in record P_u and T_u as input to predict the next POI p_{i+1} that the user will visit at time t_{i+1} .

4 Methodology

The overall framework of GNPR-SID is presented in Figure 2. The framework consists of two main modules: (1) the **semantic ID**

construction module maps each POI to a semantic ID (SID) using the codebook quantization, and (2) the **generative POI recommendation** module fine-tunes an LLM to generate the next POI which leverages the SID as the representation of POI.

4.1 Semantic ID Construction

To map the POI features to the SID, we first construct a semantic codebook, which is an autoencoder framework to encode the POI dense representation into a sparse space and then reconstruct the POI dense representation. A well-designed SID representation enhances the performance of subsequent LLM-based recommendation models by capturing semantic correlations between POIs. Specifically, an effective SID should exhibit three key characteristics: (i) **Semantic Richness**: It should encapsulate significant semantic information of the POI. (ii) **Semantic Similarity**: POIs with similar semantics should possess similar SIDs. (iii) **Uniqueness**: Distinct POIs should have unique SIDs to avoid conflicts.

4.1.1 POI Semantic Representation. The first challenge in mapping POIs to semantic IDs is effectively capturing and representing their semantic information. Typically, a POI is described by its category and discrete latitude-longitude coordinates. However, these attributes alone are insufficient to capture semantic associations between POIs, and discrete coordinates are suboptimal for model learning. To address this limitation, we propose a comprehensive semantic representation method for POI data. Specifically, we extract four key features to represent a POI: (1) basic category information, (2) location data, (3) temporal signals, and (4) user collaborative signals. This multi-faceted representation enriches the semantic understanding of POIs, enabling more accurate and meaningful mappings.

Spatial Features. To capture spatial features (e.g., longitude and latitude), we utilize Google Maps' Plus Codes² to convert latitude-longitude coordinates into region-based codes, denoted as r , which represent unique numeric region IDs. The plus code defines regions by recursively dividing areas using grids of varying sizes and adjusting the code length. Compared to directly using geographic coordinates as spatial features, these region-based codes r ensure that nearby POIs are assigned the same Plus Code, allowing the model to share information among spatially adjacent POIs. Such spatial grouping not only simplifies the representation of geographic relationships but also enhances the model to capture regional patterns and dependencies.

Temporal Signals. To extract temporal features, we partition a day into 24-time slots and map the historical visit times of each POI to these slots. Time slots with the highest visitation frequency are regarded as the potential operating hours of the POI, represented by t . The top 10 time slots with the highest visitation frequency are used in our experiments. Compared to directly using the raw timestamps, the time slots t provide a more structured and interpretable representation of temporal patterns, which can effectively capture the temporal dynamics of POI visits.

Collaborative Signals. Additionally, to model user collaborative signals, we analyze the 10 most frequent visitors for each POI

²<https://maps.google.com/pluscodes/>

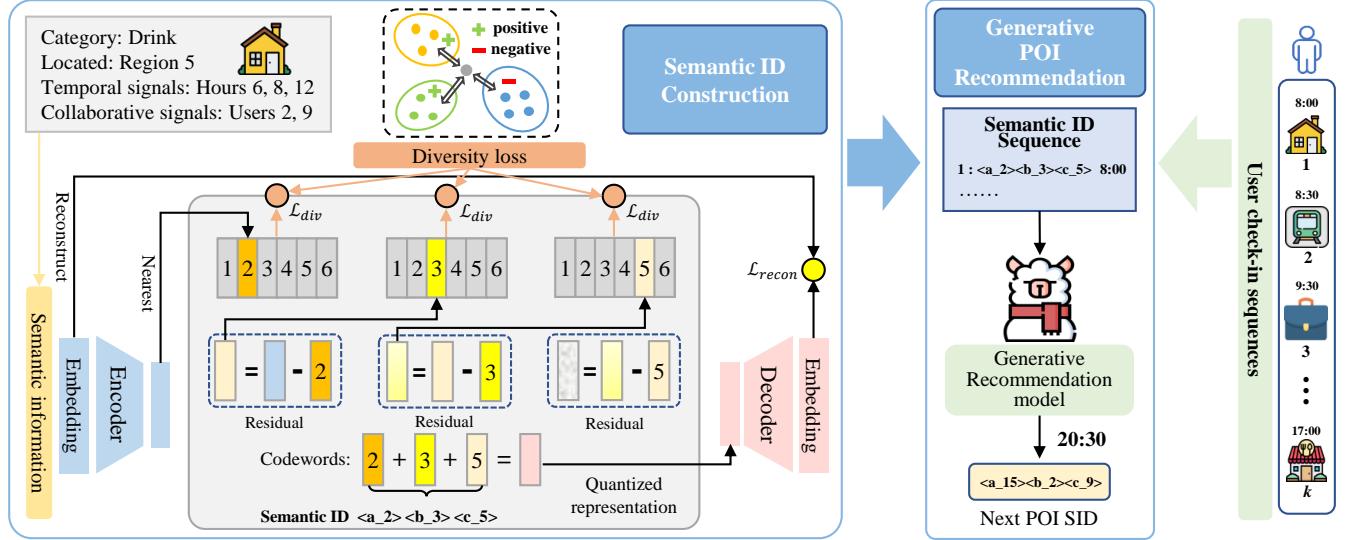


Figure 2: The overview of the Generative Next POI Recommendation with Semantic ID (GNPR-SID). It consists of two modules: (1) *Semantic POI ID Construction* module maps POI to a semantic ID (SID) using the codebook quantization and (2) *Generative POI Recommendation* module fine-tunes an LLM to use historical POI SID sequence as input and generate the next POI SID.

and analyze their co-visit patterns. This enables the model to capture latent connections between POIs through shared user behavior. The collaborative information is represented by c_u , which captures user collaboration signals based on the set of check-in users with shared interaction histories. This approach enriches the representation of POIs by incorporating both temporal dynamics and collaborative user behavior.

Finally, we construct one-hot vectors for each of the aforementioned features. The semantic representation of a POI, denoted as vector p_e , is defined as:

$$p_e = \text{concat}(c, r, t, c_u), \quad (1)$$

where c represents the POI category, r denotes the region, t corresponds to the operating time slots, and c_u captures the user collaboration signals based on shared interaction histories. Here, $\text{concat}(\cdot)$ refers to the concatenation operation. This unified representation p_e effectively integrates multiple semantic aspects of a POI, enabling a comprehensive and structured input for downstream tasks.

4.1.2 POI Quantization. After constructing the semantic vector representation p_e for each POI, the next task is to generate an ID for each POI according to the representation p_e , which can represent the correlation between semantically similar POIs. Specifically, semantically similar POIs should share as many prefix codes as possible. For example, the SID $<\text{a_15}><\text{b_2}><\text{c_1}>$ should be more similar to the POI with SID $<\text{a_15}><\text{b_2}><\text{c_9}>$ than to a POI with SID $<\text{a_15}><\text{b_12}><\text{c_2}>$. This prefix-based similarity mechanism enhances the robustness and interpretability of the SID structure, facilitating more accurate and meaningful recommendations.

Thus, we propose a novel SID quantization approach for POIs. This method ensures that the generated SIDs can represent semantic relationships among POIs, thereby improving the robustness and relevance of recommendations which can be easily adapted to the

LLM-based generative model. Specifically, we adopt the Residual Quantized Variational AutoEncoder (RQVAE) [47], a multi-level vector quantizer that leverages residual quantization, to map POI embeddings into SIDs. The RQVAE framework comprises three key modules: an encoder, a residual quantization module, and a decoder. This architecture enables hierarchical and fine-grained quantization, effectively capturing the semantic structure of POIs.

The encoder extracts a deep representation from the input POI vector, which is then used for subsequent quantization:

$$z_e = \text{Encoder}(p_e), \quad (2)$$

where p_e is the input POI vector (in Equation 1), and z_e represents the hidden representation of p_e obtained through the encoder. The encoder is implemented using fully connected layers, which effectively capture the complex semantic features of the POI. This hidden representation z_e serves as the foundation for the subsequent quantization process.

First, we define a hierarchical codebook space B consisting of L layers, where each layer contains K codeword vectors. Each codeword vector is represented in a d -dimensional space:

$$B^{(l)} = \{e_1, e_2, \dots, e_K\}, \quad (3)$$

where $B^{(l)}$ represents the codebook space of the l -th layer, and $e_k \in \mathbb{R}^{K \times d}$ denotes the k -th codeword vector in the layer. This hierarchical structure enables multi-level quantization, allowing the model to capture fine-grained semantic relationships among POIs. Residual quantization comprises L layers of quantizers (*a.k.a.*, L -layer codebook spaces). Each quantizer maps the input vector $r^{(l)}$ to the closest discrete representation (codeword vector) in its corresponding layer:

$$e_k^{(l)} = \arg \min_{e \in B^{(l)}} \|r^{(l)} - e\|, \quad (4)$$

where the $e_k^{(l)}$ is the quantized codeword at level l and the $r^{(l)}$ denotes the residual from the residual quantizer of the $(l - 1)$ -th layer. These codewords serve as cluster nodes, enabling the indexing of POIs as sequences of codewords with hierarchical information. This cluster process is repeated recursively L times to get a tuple of L codewords that represent the Semantic ID. This recursive approach approximates the input from a coarse-to-fine granularity. Note that distinct codebook spaces are used for each layer, to attain a more refined representations of indices in a coarse-to-fine manner. The input to each quantizer layer is the residual from the previous layer of quantization:

$$r^{(l)} = r^{(l-1)} - e_k^{(l-1)}. \quad (5)$$

For the first layer ($l = 0$), the input residual is initialized as $r^{(0)} = z_e$, since z_e serves as the input to the first layer. This hierarchical residual quantization process ensures fine-grained representation and preserves semantic relationships across layers. After recursive quantization for a POI, the quantized vector \hat{z}_e is obtained by summing the codeword vectors from each layer:

$$\hat{z}_e = \sum_{l=1}^L e_k^{(l)}, \quad (6)$$

where $k \in \{1, \dots, K\}$, and $e_k^{(l)}$ represents the codeword vector obtained from the l -th layer quantization. To maintain the semantic correlation between POI and SID, we employ the decoder to reconstruct POI vector z_e from the quantized vector \hat{z}_e :

$$\hat{p}_e = \text{Decoder}(\hat{z}_e), \quad (7)$$

where the decoder is also implemented by the DNN.

Based on the hierarchical codebook spaces we constructed, we generate the SID by forming a tuple of codeword indices, where each index corresponds to a codeword vector from a distinct layer of the codebook space $B^{(l)}$:

$$\text{SID} = [\text{index}(e^{(1)}), \text{index}(e^{(2)}), \dots, \text{index}(e^{(L)})], \quad (8)$$

where $e^{(l)} \in B^{(l)}$ represents the codeword vector from the l -th layer, and $\text{index}(\cdot)$ retrieves the position of the codeword vector within its respective layer. For instance, a SID of the form $\langle a_15 \rangle \langle b_2 \rangle \langle c_1 \rangle$, derived from a three-layer codebook structure, indicates that the indices $\langle a_15 \rangle$, $\langle b_2 \rangle$, and $\langle c_1 \rangle$ correspond to the codeword vectors $e_{15}^{(1)}$, $e_2^{(2)}$, and $e_1^{(3)}$ from the first, second, and third layers, respectively. This structured representation ensures that the SID captures the hierarchical semantic relationships among POIs, facilitating efficient and interpretable recommendations.

To further ensure the uniqueness of SIDs, we address any remaining cases of SID collisions by introducing an additional dimension to the SIDs. Specifically, if multiple POIs are mapped to the same SID, we append a unique identifier to each SID to distinguish them. For example, if two POIs are both mapped to SID $\langle a_15 \rangle \langle b_2 \rangle \langle c_9 \rangle$, they will be represented as $\langle a_15 \rangle \langle b_2 \rangle \langle c_9 \rangle \langle d_0 \rangle$ and $\langle a_15 \rangle \langle b_2 \rangle \langle c_9 \rangle \langle d_1 \rangle$, respectively. This approach guarantees that each POI has a distinct SID, even in cases where semantic collisions occur in the initial quantization process. By incorporating this additional dimension, we enhance the robustness and uniqueness of the SID representation, ensuring that the model can accurately differentiate between POIs with similar semantic features.

4.1.3 Model Optimization.

Reconstruction Loss. To optimize the model parameters, we employ the reconstruction loss, which quantifies the distance between the original embedding p_e and the reconstructed embedding \hat{p}_e :

$$\mathcal{L}_{\text{recon}} = \|p_e - \hat{p}_e\|^2. \quad (9)$$

Quantization Loss. Additionally, we introduce a quantization loss $\mathcal{L}_{\text{quant}}$ to evaluate the discrepancy between the residuals of the previous layer and the codeword vectors of the current layer. This loss reflects the gap between the input and its quantized representation, ensuring minimal information loss during the quantization process. The quantization loss $\mathcal{L}_{\text{quant}}$ consists of two components: (1) the quantization loss, which measures the distance between the current input vector and the codeword vector, and (2) the commitment loss, which encourages the input vector to align more closely with the codeword vector through an additional constraint term:

$$\mathcal{L}_{\text{quant}} = \sum_{l=1}^L \|\text{sg}[\mathbf{r}^{(l-1)}] - \mathbf{e}_k^{(l)}\|^2 + \beta \|\mathbf{r}^{(l-1)} - \text{sg}[\mathbf{e}_k^{(l)}]\|^2, \quad (10)$$

where sg denotes the stop-gradient operation [31], and β is a hyperparameter that controls the strength of the commitment loss. By optimizing $\mathcal{L}_{\text{quant}}$, we ensure that the quantized representation closely approximates the original input, preserving semantic fidelity throughout the quantization process.

Diversity Loss. To ensure the uniqueness of SIDs and avoid semantic collisions where multiple POIs map to the same SID, it is crucial to impose constraints during the POI quantization process. Without such constraints, relying solely on the reconstruction loss could lead to a large number of POIs being quantized into a small subset of semantic spaces, compromising the distinctiveness of SIDs. Therefore, we introduce a diversity loss to promote the diversity of codewords within the codebook. This loss encourages the model to capture distinct semantic information and ensures more efficient utilization of the discrete quantization space. We define the diversity loss from two complementary perspectives: codeword utilization constraint and intra-codeword compactness constraint. The diversity loss is defined as:

$$\mathcal{L}_{\text{div}} = \mathcal{L}_{\text{utilize}} + \sum_{i=1}^K \mathcal{L}_{\text{compactness}}(k_i), \quad (11)$$

Specifically, we aim to encourage POIs within a batch to be evenly mapped across different codeword spaces, thereby maximizing the utilization of the codebook.

$$\mathcal{L}_{\text{utilize}} = \frac{1}{N} \sum_{i=1}^K \|\text{count}_i - \frac{N}{K}\|, \quad (12)$$

where N is the total number of data, K is the number of codewords (reflected in Fig2 is the number of clusters), count_i denotes the number of vectors now existing in the i -th codeword space.

However, optimizing solely for codeword utilization may reduce semantic coherence, forcing semantically dissimilar POIs into the same cluster for distributional balance. To address this, we add a constraint enforcing semantic compactness within each codeword, ensuring vectors mapped to the same codeword are close in

embedding space.

$$\mathcal{L}_{\text{compactness}} = \frac{1}{k(k-1)} \sum_{i \neq j}^k \|e_i - e_j\|^2, \quad (13)$$

where $\|e_i - e_j\|^2$ measures the distance between codeword vectors i and j in current codeword space, k is the total number of vectors in current codeword space. This dual-constraint design maintains high codebook utilization while preserving semantic consistency. By optimizing the diversity loss, the model maximizes codebook utilization while minimizing intra-codeword variance, thereby reducing semantic collisions and enhancing the discriminative power and uniqueness of the generated SIDs.

Total Loss. To optimize the model, the total loss is defined as a weighted combination of the reconstruction loss, quantization loss, and diversity loss:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{recon}} + \mu \mathcal{L}_{\text{quant}} + \lambda \mathcal{L}_{\text{div}}, \quad (14)$$

where $\mu \in [0, 1]$ and $\lambda \in [0, 1]$ are hyperparameters that control the contributions of the quantization loss and diversity loss, respectively. This combined loss function ensures that the model not only reconstructs the original POI embeddings accurately but also maintains semantic uniqueness and diversity in the generated SIDs.

4.2 Generative POI Recommendation

In the previous steps, we outline the process of converting POI data into SIDs. Building on these SIDs, we propose a generative POI recommendation framework based on LLMs. For a given user $u \in U$, we collect their historical check-in POI sequence $P_u = \{p_1, p_2, \dots, p_i\}$ and apply the semantic ID conversion method to transform this sequence into an SID sequence $S_u = \{s_1, s_2, \dots, s_i\}$. This transformation is formalized as:

$$\mathcal{F} : P_u \rightarrow S_u, \quad (15)$$

where \mathcal{F} is the mapping function that converts POIs from their random numeric IDs to semantic IDs by our proposed method.

After learning the SID of each POI, a straightforward approach is to integrate these SIDs into the LLM vocabulary, so that LLM can fulfill the Next POI recommendation task in a generative way. However, these SIDs are still OOV tokens for LLMs, making it necessary to align language tasks with recommendation tasks, thereby bridging the gap and ensuring effective knowledge transfer. To effectively bridge this gap, inspired by [11, 42, 50], we initially construct a sequential POI prediction prompt that utilize the SID sequence of the user's historical check-in POIs in chronological order. The sequence of SIDs captures the user's evolving interests and behavioral characteristics over time.

Furthermore, considering that the temporal signals utilized in modeling the SID of a POI only reflect the frequent visitation patterns of the current POI (*i.e.*, collective temporal patterns exhibited by users), while the timestamp information in the user's historical check-in records captures behavior-specific patterns of individual users (*e.g.*, habits, long-term preferences, and personalized temporal characteristics), we propose to chronologically concatenate the SIDs of historically visited POIs with their corresponding check-in

timestamps. This integration facilitates a more comprehensive modeling of personalized user information and integrates collaborative semantics from both spatial and temporal dimensions.

Specifically, the prompt consists of two main components: (1) the instruction explicitly defines the task for the model, and (2) the input includes a user `uid` recent check-in history represented as an `SID sequence`, the associated `check-in times`, and the `current time` for prediction.

Instruction: Here is a record of a user's POI accesses, your task is based on the history to predict the POI that the user is likely to access at the specified time.

Input: The user_<uid> visited: <SID> at [time], ..., visited <SID> at [time]. When [time] user_<uid> is likely to visit:

This structured prompt enables the LLM to leverage both the semantic relationships encoded in the SIDs and the temporal patterns in the user's check-in history, facilitating accurate and context-aware next POI recommendations. In contrast to existing works [18], which transforms the next POI recommendation into question-answering pairs in their prompt, we reformulate the problem as a fill-in-the-blank task. This task design enhances the model's ability to identify and understand fine-grained temporal behaviors within specific time slots.

Since directly prompting an LLM often struggles to accurately capture the fine-grained relationships between historical check-in patterns and future POI recommendations [35], to address this limitation, we fine-tune the LLM for the next POI recommendation task. Specifically, we first create a supervised fine-tuning dataset, where we use the constructed prompt as the model input, and the SID of the POI actually visited by the user is used as the ground truth output. This approach enables the model to learn the mapping between historical check-in sequences and future POI visits, significantly enhancing the accuracy and relevance of the generated recommendations.

5 Experimental Setting

5.1 Datasets

We evaluate our approach on three real-world datasets: Foursquare-NYC [44], Foursquare-TKY [44], and Gowalla-CA [5]. Following Yan et al. [43], we preprocess each dataset by removing POIs with fewer than 10 interactions and users with fewer than 10 check-ins. The data is then sorted by time, with 80% used for training, 10% for validation, and 10% for testing. Users and POIs that do not appear in the training set are removed from the test set to ensure consistency between training and evaluation. Subsequently, the check-in records are grouped by user and ordered chronologically. For each user, the last visited POI is held out as the ground truth during evaluation, while the preceding visits are used as the input sequence. Note that in the training set, the user's visit records are also treated as historical sequences and concatenated with the test set to the specified length before being fed into the model. More detailed dataset statistics are provided in Table 1.

Table 1: Statistics of three benchmark datasets.

Dataset	NYC	TKY	CA
Categories	209	190	301
POIs	5,135	7,873	14,027
Users	1,083	2,293	6,592
Avg. Check-In	136	195	53
Fre. Check-In	68	100	10

5.2 Implementation Detail

The quantization module consists of 3 codebook layers, each containing 32 or 64 (32 for NYC and 64 for TKY and CA) codeword vectors with a dimensionality of 64. Following [47], to prevent RQ-VAE from suffering the codebook space collapse we use k-means clustering-based initialization for each codebook layers. In Equation 14, the weight of quantization loss $\mathcal{L}_{\text{quant}}$ and diversity loss \mathcal{L}_{div} is set to 1.0 and 0.25 respectively. For the generative recommendation module, we employ the LLaMA3-8B [15] as the base model and fine-tune it using LoRA [7] with rank R 16 and dropout rate 0.1. We employ a constant learning rate schedule with a learning rate of 1e-5, combined with a warm-up phase of 20 steps. The model is fine-tuned on 4 Nvidia L40 GPUs, with a batch size of 2 per GPU, gradient accumulation steps set to 8, and a sequence length of 2,048 tokens. Each input consists of the most recent 50 check-in records of the user.

5.3 Evaluation Metric & Comparison Method

Following previous work [10, 18], we employ the accuracy metric for the top-1 recommendation at the specified time.

$$\text{Acc}@1 = \frac{1}{m} \sum_{i=1}^m \mathbb{I}(y_i = \hat{y}_i), \quad (16)$$

where m is the number of test samples, y_i and \hat{y}_i represent the ground-truth label and the predict label of the i -th sample, respectively. \mathbb{I} is the indicator function, which equals 1 when the condition is true and 0 otherwise.

We compare our proposed GNPR-SID with several strong next POI recommendation baselines including the state-of-the-art LLM-based method. (1) **Traditional methods:** PRME [8], a ranking-based metric embedding method; LSTM [12], a sequential recommendation model based on RNNs; and PLSPL [41], a personalized sequential recommendation model. (2) **Transformer-based methods:** STAN [26], a spatio-temporal attention network; and GETNext [45], a Transformer-based next POI recommendation model. (3) GCN-based method: STHGCN [43], a spatio-temporal hierarchical graph convolutional network. (4) **Time-aware methods:** TPG [25], a timestamp-guided model; and ROTAN [10], a time-aware POI recommendation framework. (5) **LLM-based method:** LLM4POI [18], a state-of-the-art method based on LLMs which is the most advanced and closely related to our work. The key distinction is that LLM4POI relies on random numeric IDs, whereas our proposed GNPR-SID leverages semantic codes.

Table 2: Comparison of different models on three datasets: NYC, TKY, and CA. We present the model inputs including the POI representation approach (POI Repre.) and whether visit timestamps are utilized.

Model	Inputs		Acc@1		
	POI Repre.	Time	NYC	TKY	CA
PRME	RID	✗	0.1159	0.1052	0.0521
LSTM	RID	✗	0.1305	0.1335	0.0665
PLSPL	RID	✗	0.1917	0.1889	0.1072
STAN	RID	✗	0.2231	0.1963	0.1104
GETNext	RID	✗	0.2435	0.1829	0.1357
STHGCN	RID	✗	0.2734	0.2950	0.1730
TPG	RID	✓	0.2555	0.1420	0.1749
ROTAN	RID	✓	0.3106	0.2458	0.2199
LLM4POI	RID	✓	0.3372	0.3035	0.2065
<i>Llama3-8B</i>	RID	✓	0.2643	0.2123	0.1039
<i>Llama3-8B</i>	SID	✓	0.2885	0.2282	0.1254
GNPR-SID	SID	✓	0.3618	0.3062	0.2403

Llama3-8B denotes the performance of Llama3-8B in a zero-shot setting, evaluated using both ID and SID, respectively.

6 Experimental Result

6.1 Main Result

In this section, we compare the performance of our proposed GNPR-SID with various baselines on three benchmark datasets. The results shown in Table 2 demonstrate that GNPR-SID consistently outperforms all baselines across all datasets. Specifically, GNPR-SID achieves substantial improvements in top-1 accuracy compared to traditional model ROTAN, with relative gains of 16%, 24%, and 10% on the NYC, TKY, and CA datasets, respectively. These improvements underscore the efficacy of our proposed generative recommendation approach in the next POI recommendation task. This is attributed to the integration of the inherent commonsense knowledge and deep reasoning capabilities of LLMs into the POI recommendation task. Furthermore, we also compare GNPR-SID with the state-of-the-art LLM based method LLM4POI [18], which uses random numeric IDs to represent POIs. From the results we can find that GNPR-SID achieves 7%, 1%, and 16% of improvement compared with the SOTA method LLM4POI on the NYC, TKY, and CA datasets, respectively. This improvement demonstrate the effectiveness of our proposed semantic ID method which can capture more semantic information of POI than simply using the random numeric IDs.

6.2 Ablation Study

To verify the effectiveness of each module in our proposed GNPR-SID, we conduct experiments on the following ablation models: (1) **w/o SID:** we directly use random number IDs instead of semantic ID (in Equation 8), and keep the generative recommendation module. (2) **w/o Time:** we remove the visit time (in § 4.2) from the users check-in records and use only the SID sequence as an input to LLM. (3) **w/o \mathcal{L}_{div} :** we remove the diversity loss (in Equation 11) during semantic ID construction. (4) **w/o $\mathcal{L}_{\text{quant}}$:** we remove the

Table 3: Results of ablation models on NYC, TKY, and CA datasets with Acc@1 metric.

Ablations	NYC	TKY	CA
GNPR-SID	0.3618	0.3062	0.2403
w/o SID	0.3356 (\downarrow 7.24%)	0.2812 (\downarrow 8.18%)	0.2296 (\downarrow 4.45%)
w/o Time	0.2924 (\downarrow 19.18%)	0.2568 (\downarrow 16.13%)	0.1971 (\downarrow 17.8%)
w/o \mathcal{L}_{div}	0.3528 (\downarrow 2.48%)	0.2923 (\downarrow 4.54%)	0.2313 (\downarrow 3.72%)
w/o $\mathcal{L}_{\text{quant}}$	0.3429 (\downarrow 5.23%)	0.2900 (\downarrow 5.29%)	0.2279 (\downarrow 5.14%)
w/o $\mathcal{L}_{\text{recon}}$	—	—	—

Table 4: Statistics of the constructed SID when using different weights λ of diversity loss.

λ	NYC			TKY		
	Unique \uparrow	Collision \downarrow	Acc@1	Unique \uparrow	Collision \downarrow	Acc@1
0.00	2,916	711	0.3528	4,244	1,182	0.2923
0.25	3,259	590	0.3618	4,465	1,147	0.3062
0.50	3,292	567	0.3459	4,481	1,126	0.2898
0.75	3,313	554	0.3368	4,529	1,106	0.2865

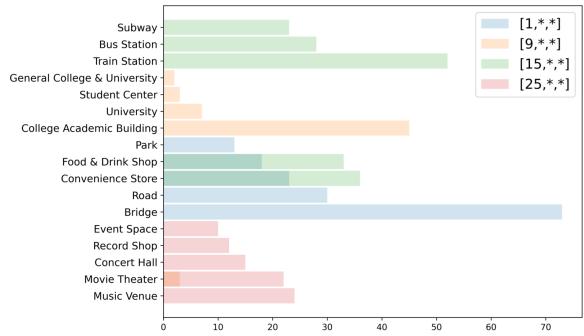
quantization loss (in Equation 10) during semantic ID construction. (5) **w/o $\mathcal{L}_{\text{recon}}$:** we remove the reconstruction loss (in Equation 9) during semantic ID construction.

Table 3 presents the Acc@1 performance of various ablation models across three datasets. In Table 3, “—” indicates that the ablation model collapsed during training and failed to converge. The results in Table 3 demonstrate that the performance of these ablation models is lower than our proposed GNPR-SID, thereby validating the effectiveness of all modules within GNPR-SID. Notably, the model fails to converge when the $\mathcal{L}_{\text{recon}}$ loss function is removed. It is intuitive that the absence of constraints on the autoencoder output leads to the codebook being entirely incapable of modeling any semantic information of POIs.

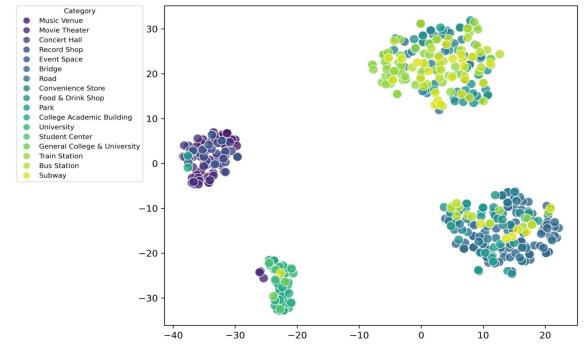
6.3 Analysis of Diversity Loss

In this section, we investigate the effectiveness of our proposed diversity loss. Table 4 shows the model performance under different weight λ of diversity loss when we combine this loss function to the total loss. Recall that, we propose the diversity loss to prevent semantic collapse, where multiple POIs are mapped to the same SID. To quantitatively evaluate the impact of the diversity loss, we propose two metrics to assess the quality of the generated SIDs. First, we measure the number of unique SIDs generated by quantization. Second, we calculate the number of colliding SIDs (*i.e.*, multiple POIs map to the same SID) generated by quantization.

Table 4 shows the model performance under different values of the hyper-parameter λ . From Table 4, it can be observed that as the weight of the diversity loss increases, the collisions of semantic ID decreases. This observation demonstrates that the diversity loss effectively encourages the model to distribute POIs more uniformly across the semantic space. However, when the weight of the diversity loss becomes excessively large, the penalty forces the model to map the POI semantic vectors as uniformly as possible within the semantic space. This can result in similar POIs being assigned



(a) Category frequency distribution for different SID prefixes.



(b) The t-SNE visualization of SID.

Figure 3: SIDs semantic relevance analysis.

to different semantic clusters, thereby negatively impacting the performance of the next POI recommendation.

6.4 Semantic Relevance Analysis of SID

To intuitively validate whether the SIDs generated by our proposed method can effectively model the semantic information of POIs, we visualize the POIs in the TKY dataset. In Figure 3a, we randomly select four SID prefixes with varying first codeword, and show the distribution of these SIDs across the corresponding POI categories. From Figure 3a we can find that different semantic spaces (*a.k.a.*, different SID prefixes) effectively capture information related to distinct semantic categories of POIs, while POIs of similar categories are quantized into the same semantic space (*a.k.a.*, the same SID prefix). For example, SIDs starting with $<\text{a}_15>$ represent transportation-related venues (*e.g.*, subways, bus stations, and train stations), while SIDs starting with $<\text{a}_25>$ correspond to entertainment venues (*e.g.*, music venues, record stores, and cinemas).

Furthermore, in Figure 3b, we present the distribution of SIDs for POIs in the TKY dataset after t-SNE dimensionality reduction. The POIs are colored according to their categories as provided in the dataset. From Figure 3b, it can be observed that POIs of different categories (*a.k.a.*, points of different colors) are well-separated, indicating significant distinguishability in their SIDs. In contrast, POIs of the same category (*a.k.a.*, points of the same color) are positioned closely. This experiment visually confirms that our proposed POI

Table 5: Comparison between GNPR-SID and LLM4POI on out-of-domain dataset.

Train Data	GNPR-SID			LLM4POI		
	NYC	TKY	CA	NYC	TKY	CA
NYC	0.3618	0.2946	0.2297	0.3373	0.2594	0.1885
TKY	0.3481	0.3062	0.2342	0.3463	0.3035	0.1960
CA	0.3536	0.2869	0.2403	0.3344	0.2600	0.2065

Table 6: Efficiency comparison on the NYC dataset.

Models	Training Time	Test Time	Total Tokens
LLM4POI	5,108s	230.36s	5,061,008
GNPR-SID	2,121s (↓ 58.48%)	110.62s (↓ 51.98%)	2,288,764 (↓ 54.78%)

quantization method can map semantically similar POIs to adjacent SID spaces, which enables the LLM to better understand the semantic relationships between POIs and achieve improved generalization performance when encountering unseen POIs during inference.

6.5 Performance on Out-of-domain Data

In this paper, we propose the semantic ID construction module, independent of LLMs, to convert POI information into SIDs. In contrast, existing methods (e.g., LLM4POI) employ random IDs, which forces the LLM to memorize the POI semantic information behind each random ID. This approach leads to generalization issues when encountering new POIs, as the model fails to comprehend the semantic meaning of the random ID for new POI. Table 5 shows the performance on out-of-domain datasets. The performance degradation of GNPR-SID on out-of-domain data is notably less pronounced compared to LLM4POI across all datasets, which further demonstrates that our proposed GNPR-SID exhibits superior generalization capabilities over LLM4POI.

Especially in the model generalizes from short-sequence datasets (CA with 53 average check-ins) to long-sequence datasets (TKY with 195 average check-ins), our model GNPR-SID outperforms LLM4POI by a significant margin. Additionally, the generalization ability of LLM4POI heavily relies on the historical sequences of similar users. It feeds these sequences into the prompt to guide the model's predictions, requiring extra resources and computation, which is clearly unsuitable for out-of-domain data. In contrast, our model requires no additional prompts and achieves better performance than LLM4POI using only the current user's historical sequence. This further highlights the superior generalization capabilities of GNPR-SID.

6.6 Analysis of Model Efficiency

In this paper, we propose the GNPR-SID that uses SID, which includes POI semantic information, as the input to the LLM. Compared to methods using RID as input, our approach does not require the descriptive text of POIs to be fed into the LLM. This also leads to a significant reduction in the number of tokens in the LLM input of GNPR-SID compared to methods that use RID. Intuitively, this makes our method faster when predicting the next POIs using the

generative model, especially when the historical check-in sequence is long. Table 6 presents a comparison of the efficiency between our SID-based GNPR-SID and the RID-based method LLM4POI. We train both models for 8 epochs on the NYC dataset. As can be seen from Table 6, since we compress all semantic information of the POI into the SID, the token usage in our GNPR-SID is lower than that of LLM4POI, resulting in a reduction of 54.78%. Additionally, our GNPR-SID shows 51.98% improvement in terms of test time. This result indicates that using the SID not only benefits the understanding of POI semantics and generates more accurate recommendations, but also improves the training and inference efficiency since we reduce the input tokens for the generative recommendation model.

7 Conclusion

In this paper, we proposed Generative Next POI Recommendation with Semantic ID (GNPR-SID), a novel LLM-based framework for next POI recommendation that uses the POI semantic ID to integrate more spatial-temporal information. By designing a semantic ID construction module that captures POI semantics and a generative recommendation module that fine-tunes LLMs, GNPR-SID significantly improves recommendation accuracy and generalizability. Our hierarchical POI quantization method ensures semantically similar POIs share similar IDs, while the diversity loss enhances codeword distinctiveness. Extensive experiments on three benchmark datasets demonstrate that GNPR-SID outperforms state-of-the-art methods, achieving accuracy improvements of up to 16%. Semantic similarity analysis further confirms that our generated SIDs effectively capture POI semantic relationships.

Acknowledgments

This work was supported by the National Key R&D Program of China 2024YFE0111800, the National Natural Science Foundation of China (62432002, 62032001, 62406061, and T2293773), and the Natural Science Foundation of Shandong Province (ZR2023QF159).

References

- [1] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609* (2023).
- [2] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023. Qwen-VL: A Versatile Vision-Language Model for Understanding, Localization, Text Reading, and Beyond. *arXiv preprint arXiv:2308.12966* (2023).
- [3] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems* 33 (2020), 1877–1901.
- [4] Chen Cheng, Haiqin Yang, Michael R Lyu, and Irwin King. 2013. Where you like to go next: Successive point-of-interest recommendation. In *Twenty-Third international joint conference on Artificial Intelligence*.
- [5] Eunjoon Cho, Seth A Myers, and Jure Leskovec. 2011. Friendship and mobility: user movement in location-based social networks. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. 1082–1090.
- [6] 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*. 473–482.
- [7] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelfen, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783* (2024).
- [8] Shanshan Feng, Xutao Li, Yifeng Zeng, Gao Cong, Yeow Meng Chee, and Quan Yuan. 2015. Personalized ranking metric embedding for next new poi recommendation. (2015).

- [9] 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)*. IEEE, 1530–1535.
- [10] Shanshan Feng, Feiyi Meng, Lisi Chen, Shuo Shang, and Yew Soon Ong. 2024. Rotan: A rotation-based temporal attention network for time-specific next poi recommendation. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 759–770.
- [11] Shijie Geng, Shuchang Liu, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. 2022. Recommendation as Language Processing (RLP): A Unified Pretrain, Personalized Prompt & Predict Paradigm (P5). In *Proceedings of the 16th ACM Conference on Recommender Systems*. 299–315.
- [12] Alex Graves and Alex Graves. 2012. Long short-term memory. *Supervised sequence labelling with recurrent neural networks* (2012), 37–45.
- [13] Jiaming Han, Kaixiong Gong, Yiyuan Zhang, Jiaqi Wang, Kaipeng Zhang, Dahua Lin, Yu Qiao, Peng Gao, and Xiangyu Yue. 2024. OneLLM: One Framework to Align All Modalities with Language. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 26584–26595.
- [14] Jing He, Xin Li, Lejian Liao, Dandan Song, and William Cheung. 2016. Inferring a personalized next point-of-interest recommendation model with latent behavior patterns. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 30.
- [15] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685* (2021).
- [16] Md Ashraful Islam, Mir Mahathir Mohammad, Sarkar Snigdha Sarathi Das, and Mohammed Eunus Ali. 2022. A survey on deep learning based Point-of-Interest (POI) recommendations. *Neurocomputing* 472 (2022), 306–325.
- [17] Dejiang Kong and Fei Wu. 2018. HST-LSTM: A hierarchical spatial-temporal long-short term memory network for location prediction.. In *Ijcai*, Vol. 18. 2341–2347.
- [18] Peibo Li, Maarten de Rijke, Hao Xue, Shuang Ao, Yang Song, and Flora D Salim. 2024. 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*. 1463–1472.
- [19] 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*. 2009–2019.
- [20] Nicholas Lim, Bryan Hooi, See-Kiong Ng, Yong Liang Goh, Renrong Weng, and Rui Tan. 2022. Hierarchical multi-task graph recurrent network for next poi recommendation. In *Proceedings of the 45th international ACM SIGIR conference on Research and development in Information Retrieval*. 1133–1143.
- [21] Nicholas Lim, Bryan Hooi, See-Kiong Ng, Xueou Wang, Yong Liang Goh, Renrong Weng, and Jagannadan Varadarajan. 2020. STP-UDGAT: Spatial-temporal-preference user dimensional graph attention network for next POI recommendation. In *Proceedings of the 29th ACM International conference on information & knowledge management*. 845–854.
- [22] Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. 2016. Predicting the next location: A recurrent model with spatial and temporal contexts. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 30.
- [23] Shuai Liu, Ning Cao, Yile Chen, Yue Jiang, and Gao Cong. 2024. nextlocclm: next location prediction using LLMs. *arXiv preprint arXiv:2410.09129* (2024).
- [24] Yiding Liu, Tuan-Anh Nguyen Pham, Gao Cong, and Quan Yuan. 2017. An experimental evaluation of point-of-interest recommendation in location-based social networks. (2017).
- [25] Yan Luo, Haoyi Duan, Ye Liu, and Fu-Lai Chung. 2023. Timestamps as Prompts for Geography-Aware Location Recommendation. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*. 1697–1706.
- [26] Yingtao Luo, Qiang Liu, and Zhaocheng Liu. 2021. Stan: Spatio-temporal attention network for next location recommendation. In *Proceedings of the web conference 2021*. 2177–2185.
- [27] Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan Hulikal Keshavan, Trung Vu, Lukasz Heldt, Lichan Hong, Yi Tay, Vinh Tran, Jonah Samost, et al. 2023. Recommender systems with generative retrieval. *Advances in Neural Information Processing Systems* 36 (2023), 10299–10315.
- [28] Pablo Sánchez and Alejandro Bellogín. 2022. Point-of-interest recommender systems based on location-based social networks: a survey from an experimental perspective. *ACM Computing Surveys (CSUR)* 54, 11s (2022), 1–37.
- [29] Ke Sun, Tieyun Qian, Tong Chen, Yili 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*, Vol. 34. 214–221.
- [30] Weiwei Sun, Lingyong Yan, Zheng Chen, Shuaiqiang Wang, Haichao Zhu, Pengjie Ren, Zhumin Chen, Dawei Yin, Maarten Rijke, and Zhaochun Ren. 2024. Learning to tokenize for generative retrieval. *Advances in Neural Information Processing Systems* 36 (2024).
- [31] Aaron Van Den Oord, Oriol Vinyals, et al. 2017. Neural discrete representation learning. *Advances in neural information processing systems* 30 (2017).
- [32] En Wang, Yiheng Jiang, Yuanbo Xu, Liang Wang, and Yongjian Yang. 2022. Spatial-temporal interval aware sequential POI recommendation. In *2022 IEEE 38th international conference on data engineering (ICDE)*. IEEE, 2086–2098.
- [33] Jiawei Wang, Renhe Jiang, Chuang Yang, Zengqing Wu, Makoto Onizuka, Ryosuke Shibasaki, Noboru Koshizuka, and Chuan Xiao. 2024. Large language models as urban residents: An llm agent framework for personal mobility generation. *arXiv preprint arXiv:2402.14744* (2024).
- [34] Wenjie Wang, Honghui Bao, Xinyu Lin, Jizhi Zhang, Yongqi Li, Fuli Feng, See-Kiong Ng, and Tat-Seng Chua. 2024. Learnable item tokenization for generative recommendation. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*. 2400–2409.
- [35] Xinglei Wang, Meng Fang, Zichao Zeng, and Tao Cheng. 2023. Where would i go next? large language models as human mobility predictors. *arXiv preprint arXiv:2308.15197* (2023).
- [36] Yujing Wang, Yingyan Hou, Haonan Wang, Ziming Miao, Shabin Wu, Qi Chen, Yuqing Xia, Chengmin Chi, Guoshuai Zhao, Zheng Liu, et al. 2022. A neural corpus indexer for document retrieval. *Advances in Neural Information Processing Systems* 35 (2022), 25600–25614.
- [37] Yidan Wang, Zhaochun Ren, Weiwei Sun, Jiyuan Yang, Zhixiang Liang, Xin Chen, Ruobing Xie, Su Yan, Xu Zhang, Pengjie Ren, et al. 2024. Enhanced generative recommendation via content and collaboration integration. *arXiv preprint arXiv:2403.18480* (2024).
- [38] Zhaoobo Wang, Yanmin Zhu, Haobing Liu, and Chunyang Wang. 2022. 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*. 1154–1163.
- [39] Zhaoobo Wang, Yanmin Zhu, Chunyang Wang, Wenze Ma, Bo Li, and Jiadi Yu. 2023. Adaptive Graph Representation Learning for Next POI Recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 393–402.
- [40] Wilson Wongso, Hao Xue, and Flora D Salim. 2024. GenUP: Generative User Profilers as In-Context Learners for Next POI Recommender Systems. *arXiv preprint arXiv:2410.20643* (2024).
- [41] Xuxia Wu, Ke Li, Guoshuai Zhao, and Xueming Qian. 2020. Personalized long-and short-term preference learning for next POI recommendation. *IEEE Transactions on Knowledge and Data Engineering* 34, 4 (2020), 1944–1957.
- [42] Hao Xue, Bhanu Prakash Voutharoja, and Flora D. Salim. 2022. Leveraging language foundation models for human mobility forecasting. In *Proceedings of the 30th International Conference on Advances in Geographic Information Systems*. 9.
- [43] 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*. 403–412.
- [44] Dingqi Yang, Daqing Zhang, Vincent W Zheng, and Zhiyong Yu. 2014. Modeling user activity preference by leveraging user spatial temporal characteristics in LBSNs. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 45, 1 (2014), 129–142.
- [45] 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*. 1144–1153.
- [46] Shukang Yin, Chaoyou Fu, Sirui Zhao, Ke Li, Xing Sun, and Chen Enhong Xu Tong. 2023. A Survey on Multimodal Large Language Models. *arXiv preprint arXiv:2306.13549* (2023).
- [47] Neil Zeghidour, Alejandro Luebs, Ahmed Omran, Jan Skoglund, and Marco Tagliasacchi. 2021. Soundstream: An end-to-end neural audio codec. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 30 (2021), 495–507.
- [48] Lu Zhang, Zhu Sun, Ziqing Wu, Jie Zhang, Yew Soon Ong, and Xinghua Qu. 2022. Next Point-of-Interest Recommendation with Inferring Multi-step Future Preferences. In *IJCAI*. 3751–3757.
- [49] Pengpeng Zhao, Anjing Luo, Yanchi Liu, Jiajie Xu, Zhixu Li, Fuzhen Zhuang, Victor S Sheng, and Xiaofang Zhou. 2020. Where to go next: A spatio-temporal gated network for next poi recommendation. *IEEE Transactions on Knowledge and Data Engineering* 34, 5 (2020), 2512–2524.
- [50] Bowen Zheng, Yupeng Hou, Hongyu Lu, Yu Chen, Wayne Xin Zhao, Ming Chen, and Ji-Rong Wen. 2024. Adapting Large Language Models by Integrating Collaborative Semantics for Recommendation. In *IEEE 40th International Conference on Data Engineering (ICDE)*. 1435–1448.

A Appendix

A.1 Data Augment

A.1.1 Multiple cropping. If the input sequence length exceeds several times the given length, the sequence is divided into multiple non-overlapping segments. If the input sequence length is greater than the given length but less than twice that length, it is randomly cropped into two overlapping segments. If the input sequence length is shorter than the given length, no cropping is performed.



Figure 4: Replace the next POI recommendation task with a fill-in-the-blank task.

A.1.2 Fill-in-the-blank. For the cropped dataset, 1 out of every 5 samples is randomly selected to replace the next POI recommendation task with a fill-in-the-blank task, where the target is the POI corresponding to a randomly chosen timestamp from the historical check-in data. The specific operation is shown in Figure 4.

A.2 Baseline

We introduce the details of the baseline methods we used in our experiments:

- PRME [8]: A metric embedding-based method, which jointly incorporates sequential transition, user preference, and geographical influence.
- LSTM [12]: A variant of the RNN model, which handles long sequence dependency and Vanishing gradient problems.
- PLSPL [41]: A RNN-based method, which learns the user’s long-term pattern by using attention mechanism and short-term preference, and short-term preference with LSTM.
- STAN [26]: A transformer-based method, which modifies the attention coefficients using temporal and spatial interval information.
- GETNext [45]: A transformer and GNN-based method, which employs a user-agnostic global trajectory flow map and a novel Graph Enhanced Transformer model.
- STHGCN [43]: Constructing a hypergraph to capture inter and intra-user relations, STHGCN proposes a hypergraph transformer and solves the cold-start problem.
- TPG [25]: A transformer-based method, which explicitly uses target times as prompts for the geography-aware location recommendation.

- ROTAN [10]: A time-aware next POI recommendation framework that incorporates target time information into the recommendation system. It introduces a novel time-aware attention mechanism based on rotation, utilizing the Time2Rotation technique. This technique represents time intervals as rotational vectors and applies rotational operations to more effectively capture the influence of time information on user behavior.
- LLM4POI [18]: The first LLM-based next POI recommendation method. By leveraging the commonsense knowledge of LLMs, the model can understand the underlying meaning of contextual information. Designing a specific prompt for POIs to preserve heterogeneous location-based social network data in its original format, hence avoiding the loss of contextual information.

A.3 Prompt format

In Section 4.2, we provided the input format for the model under normal conditions. For the ablation experiments, the inputs for W/O SID and W/O time are shown as follows.

Table 7: The input format of w/o SID

Instruction	Here is a record of a user’s POI accesses, your task is based on the history to predict the POI that the user is likely to access at the specified time.
Input	The user_<uid> visited: <RID> at [time], ..., visited <RID> at [time]. When [time] user_<uid> is likely to visit:
Example	The user_<1>, visited <3312> at 2012-05-21 23:15, ..., visited <4345> at 2013-01-29 22:21, When 2013-01-31 14:05, user_<1> is likely to visit:

Table 8: The input format of w/o time

Instruction	Here is a record of a user’s POI accesses, your task is based on the history to predict the next POI that the user is likely to access.
Input	The user_<uid> visited: <SID>, ..., visited <SID>, and in the next time user_<uid> is likely to visit:
Example	The user_<1>, visited <a_20> <b_21> <c_19> <d_1>, <a_11> <b_19> <c_6> ..., <a_11> <b_19> <c_6>, and in the next time user_<1> is likely to visit: