Information Processing and Management 63 (2026) 104434

Contents lists available at ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/ipm

LRSA: LLM-RecSys alignment for time-specific next POI recommendation

Jinhui Zhu, Xiangfeng Luo *, Xin Yao, Xiao Wei

School of Computer Engineering and Science, Shanghai University, Shanghai, China

ARTICLE INFO

Keywords:
Point-of-interest recommendation
Large language models
Rotational alignment
Fusion prompt

ABSTRACT

Time-specific next point-of-interest (POI) recommendation aims to predict which POI a user will visit at a given time, a task challenged by the limited textual information in ID-based historical data. Although large language models (LLMs) demonstrate strong commonsense reasoning, their performance in POI recommendation remains suboptimal due to the semantic gap between textual inputs and ID-based user preferences. To address this, we propose a novel LLM-RecSys Alignment (LRSA) framework. First, the historical fact collector is designed to identify the influential trajectories efficiently. Second, the rotational alignment is proposed to align the semantics of the LLMs with the ID-based models. Finally, we design the fuse prompt to combine the user preference into the plain text prompt. Moreover, rather than directly input the fuse prompt to the LLMs, we propose the hierarchical prompt tuning to facilitate a two-stage learning process, starting from low-level plain text prompt to high-level multimodal fuse prompt for the LLMs. Experiments on three benchmark datasets (NYC, TKY, and Gowalla) demonstrate that our method achieves average improvements of 24.39% and 9.93% in Acc@1, 25.26% and 12.79% in NDCG@5, and 10.46% and 15.41% in MRR over the latest ID-based models and LLM-based models, respectively.

1. Introduction

Location-based social networks(LBSNs) services, such as Gowalla, Foursquare, and Dianping, are extensively integrated into people's daily lives (An et al., 2024; Li, Chen et al., 2024; Sun et al., 2024). Check-in trajectories are ID-based multivariate data generated by users in LBSNs services, containing limited textual information (Ding et al., 2023; Zhang, Li et al., 2023). Next point of interest(POI) recommendation predicts the POI a user will visit based on the order of visits in the current trajectory and spatiotemporal context information (Gan & Ma, 2023; Zhang, Lai et al., 2023). The time-specific next POI recommendation(TSNPR) task considers that users will visit different POIs at different times, incorporating a time factor into the next POI recommendation task to predict the POI and POI category a user will visit at a specific time in the future (Huang et al., 2024).

Previous approaches have mined and modeled check-in trajectories from two perspectives: sequential representation (Guo et al., 2020; He & McAuley, 2016; Zhao et al., 2020; Zhu et al., 2017) and graph representation (Chen et al., 2022; Qin et al., 2023; Wang et al., 2022; Yang et al., 2022). Recurrent Neural Networks (RNNs) effectively capture the temporal dependencies and contextual correlations of user behavior sequences in sequence tasks, making RNN-based variations widely used in next POI recommendation tasks. Graph-based methods establish graph structures connecting users or POIs based on attributes like time or distance, aggregating neighbor nodes using graph methods to extract user behavioral preference features and enhance understanding and representation of

* Correspondence to: School of Computer Engineering and Science, Shanghai University, Shanghai 200444, China.
E-mail address: luoxf@shu.edu.cn (X. Luo).

https://doi.org/10.1016/j.ipm.2025.104434
Received 20 August 2025; Received in revised form 30 September 2025; Accepted 2 October 2025
Available online 11 October 2025

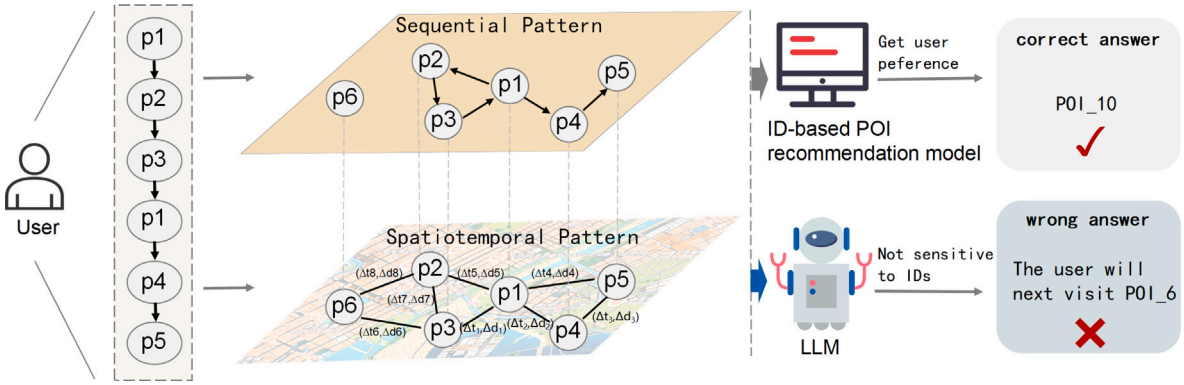

Fig. 1. Comparison between the LLMs and the ID-based POI recommendation models. Traditional ID-based POI recommendation models capture sequential and spatiotemporal patterns, summarize them to form user preferences, and complete user recommendations. The LLMs are not sensitive to specific ID tags when processing ID-based check-in trajectories, as they cannot comprehend the user preferences encoded in the POI IDs. This limitation leads to poor recommendation performance.

user interests. These approaches are all ID-based POI recommendation models, assigning unique IDs to each user and POI, converting them into trainable embedding vectors to learn user preferences embedded in these vectors for subsequent POI recommendations. However, they lack exploration of POI text features (such as categories) and fail to leverage external world semantics related to check-in trajectories.

Recently, large language models(LLMs) have demonstrated advanced performance in various natural language processing tasks (Kim et al., 2020; Lin et al., 2024; Lyu et al., 2024; Ye et al., 2022). They have introduced a new training paradigm called "text to text", where tasks are adapted to accept input data in a text format. Through pre-training on real-world data, the LLMs inherently possess broad worldly knowledge and emergent capabilities, offering significant advantages when tackling tasks.

However, the LLMs are not sensitive to specific ID tags when processing ID-based check-in trajectories, as they cannot comprehend the user preferences encoded in user IDs and POI IDs. This limitation leads to poor recommendation performance, as depicted in Fig. 1. Therefore, enabling LLMs to align their worldly knowledge with user preferences for comprehensive recommendations poses an important challenge in utilizing LLM-based methods to accomplish the TSNPR task.

To address the important challenge, we propose an LLM-RecSys Alignment (LRSA) framework to align the ID-based user preference and world knowledge. First, to reduce the cost of time and improve the efficiency in fine-tuning LLMs, the influential trajectory collector is designed to identify the influential historical trajectories efficiently. Second, the rotational alignment is proposed to align the semantics space of the LLMs with the ID-based recommendation models. Finally, we design the fusion prompt to combine the user preference as suffix with the plain POI text. Moreover, rather than directly input the fusion prompt to the LLMs, we propose the hierarchical prompt tuning to facilitate a two-stage learning process, starting from a low-level plain text prompt to a high-level multimodal fusion prompt for the LLMs. We not only familiarize the LLM with the recommendation mechanism utilizing plain text prompt, but also internalize the user preference encoded by the ID-based recommendation models with fusion prompt. The contributions of our work can be summarized as follows:

- Inspired by the influence function, the influential trajectory collector is designed to efficiently identify influential historical trajectories, thereby reducing time costs and enhancing efficiency in fine-tuning the LLMs.
- The rotational alignment is proposed to form the user preference representation that can be understood by the LLMs. It maps the LLMs' token representation and the ID-based user preference representation into the same latent space.
- We introduce a fusion prompt design method that combines the ID-based user preference representation as a suffix with the POI text. The fusion prompt contains both world knowledge and user preference representation. To the best of our knowledge, our work is the first to explore how to align the ID-based recommendation models and the LLMs to achieve the TSNPR task.
- Extensive experiments on three check-in trajectories datasets show that LRSA exhibits performance that exceeds that of prior state-of-the-art models and illustrates the ability to generalize effectively.

The remainder of this paper is organized as follows: Section 2 reviews related methods in next POI recommendation and LLMs for recommender systems. Section 3 presents the research objectives of this work. Section 4 introduces the problem formulation. Section 5 provides a detailed description of the proposed LRSA. Section 6 presents the experimental results. Section 7 discusses the theoretical and practical significance, and finally, Section 8 concludes the paper.

2. Related work

This section mainly introduces the related researches, including Next POI Recommendation and LLMs for Recommender Systems.

2.1. ID-based POI recommendation

Early methods, such as recurrent neural networks (RNNs), possess the capability to effectively process the temporal correlations and contextual relationships inherent in sequences of user behavior. Consequently, approaches utilizing variants of RNNs have been extensively employed for the purpose of next POI recommendation based on sequence representation. For instance, LSTPM (Sun et al., 2020) utilized standard LSTM and geographical extension LSTM to encode the time and spatial information of sequences, modeling users' short-term preferences, but neglected the problem of personalized item frequency (PIF). To address the PIF problem, STAN (Luo et al., 2021) employed a dual-attention structure to learn the frequency information of personalized visits to point-of-interest in the sequences. To solve the impact of users' future behaviors on decision-making, CFPRec (Zhang et al., 2022) used an LSTM-based preference encoder to predict users' future preferences.

To address the limitations of neighborhood knowledge in these early models, recently, graph-based methods have been able to improve understanding and representation of user interests by incorporating relational information in user behavior sequences. For instance, STGCAN (Wang et al., 2022) integrated temporal and spatial information to construct a temporal knowledge graph considering different users' acceptance levels for distance and time, but lacked the ability to capture fine-grained mobility features. STKGRec (Chen et al., 2022) introduced new transition relationships to build a spatiotemporal knowledge graph for learning users' fine-grained mobility features. Previous methods did not take into account the temporal correlation of POI categories. GETNext (Yang et al., 2022) captured general mobility patterns of users by constructing user trajectory flow graphs and incorporating the temporal features of POI categories. To address the lack of deep semantic information, HyperSE (Zeng et al., 2025) exploited multi-scale hypergraph correlation knowledge while jointly optimizing structural and semantic representations for trajectories.

However, these above ID-based methods lack exploration of POI text features (such as categories) and fail to leverage external real-world knowledge related to check-in trajectories. Differently, our work aims to address the shortcomings of ID-based methods by leveraging the LLMs, which contain extensive world knowledge and common-sense reasoning abilities. To overcome the gap in the latent space between the ID-based methods and the LLMs, the rotational alignment is proposed to form the user preference representation that the LLMs can understand. It maps the LLMs' token representation and the ID-based user preference representation into the same latent space.

2.2. LLM-based recommendation

The LLMs are widely used in recommender systems (Tian et al., 2024; Xu et al., 2024; Zhang et al., 2025). Although the LLMs follow a "text to text" paradigm, with their intrinsic wealth of world knowledge, many scholars are attempting to adapt the LLMs to sequence data such as user–item pairs for sequence recommendation(SR). In order for the LLMs to accurately capture the semantic representation of item text, IDGenRec (Tan et al., 2024) designed an index generator to extract key textual details from verbose metadata for SR. AutoDisenSeq (Wang et al., 2025) used the text generation capabilities of LLMs to design an optimal attention representation architecture for various scenarios with different data, but lacked the utilization of existing model knowledge. Some studies attempted to combine the traditional SR models with the LLMs to leverage their respective strengths. To address the issue of inconsistency between the item semantic space and the user behavioral space, ToolRec (Zhao et al., 2024) used the LLMs as a proxy for the user, learning and mimicking users by controlling and invoking small model tools to accomplish SR tasks, but lacked effective knowledge alignment. LLaRA (Liao et al., 2024) enhanced the LLM's SR by leveraging embedding vectors from traditional sequential recommenders, but lacked validation in cold-start scenarios.

Recently, how to utilize the LLMs to adapt to check-in trajectories has become a research focus. Different from traditional sequence data, check-in trajectories are ID-based multimodal data with minimal text information, making it challenging for the LLMs to directly understand. MRP-LLM (Wu et al., 2024) and LLMmove (Feng, Lyu et al., 2024) both innovated on prompt engineering, enabling LLMs to gradually understand user knowledge in check-in data through continuous prompting. However, they lacked deep adaptation to the target task, and an untuned LLM failed to meet the requirements of the next POI recommendation task. LLM4POI (Li, de Rijke et al., 2024) introduced, for the first time, the concept of fine-tuning LLMs to achieve the next POI recommendation task. To enable the LLMs to better adapt to the spatiotemporal information in trajectories, LLM-Next (Han et al., 2025) proposed a specific data processing method that is optimized for temporal and spatial data and fine-tunes LLMs.

However, the above LLM-based methods are not sensitive to specific ID tags when processing ID-based check-in trajectories, as they cannot comprehend the user preferences encoded in user IDs and POI IDs. Differently, our work introduces a fusion prompt design method that combines the ID-based user preference representation as a suffix with the POI text. The fusion prompt contains both world knowledge and user preference representation. During the fine-tuning stage, to enhance the LLMs' learning of the ID-based user preferences, the hierarchical prompt tuning is proposed to facilitate a two-stage learning process.

3. Research objectives

• Research Gaps and Motivation. Traditional ID-based methods lack the utilization of textual semantics and external world knowledge, which motivates the use of LLMs. Although LLMs possess rich world knowledge and strong contextual reasoning abilities, their application in POI recommendation still faces challenges. This is because check-in trajectories primarily consist of sparse, multivariate, ID-based data with limited textual information. LLMs struggle to capture fine-grained user preferences from these IDs. This often leads to suboptimal recommendation results. Existing studies often fail to bridge the gap between ID-based user preference modeling and textual world knowledge representation, thereby limiting the potential of LLMs in this area.

• Research Objectives. In response to the above challenges, this paper aims to achieve the following research objectives: (1) To construct a unified framework that combines the strengths of ID-based models (which excel at capturing user preferences) and LLMs (which excel at representing POI world knowledge) for the TSNPR task. (2) To design effective alignment mechanisms so that ID-based user preferences can be understood and utilized by LLMs. (3) To introduce efficient strategies for identifying key influential user trajectories, thereby improving the efficiency of LLM fine-tuning and recommendation effectiveness. (4) To develop adaptable prompt strategies that enable the fusion of ID-based and textual information, facilitating information transfer and collaborative modeling across multiple data modalities.

• Research Questions. Based on the above objectives, this paper addresses the following research questions: (1) How can ID-based user preference representations be effectively aligned with the world knowledge of LLMs within a unified recommendation framework? (2) In the context of POI recommendation, how can we design mechanisms to identify and utilize the most influential user trajectories to efficiently fine-tune LLMs? (3) How does the proposed framework perform in capturing user preferences and leveraging LLM knowledge compared to existing mainstream methods on standard POI datasets? (4) How does the proposed framework perform in terms of generalization and robustness in cold-start and cross-domain scenarios?

4. Problem formulation

In this section, we elaborate on the key concepts in the article and then define the studied problem.

Data Description. Each check-in record contains a unique check-in ID, user ID, timestamp, POI ID, POI category, and POI geographical coordinates (latitude and longitude). For example, in the NYC dataset, check-ins are collected from April 2012 to September 2013. Over this period, each user may have multiple check-in records.

Trajectory Construction. To analyze user movement and capture behavioral periodicity, we segment each user's check-ins into multiple trajectories based on natural daily cycles, which is a widely adopted practice in the literature. Specifically, all check-ins by a user within a single day (after sorting by time) form one trajectory. Thus, each user typically has multiple trajectories, one for each day he/she was active.

Definition 1 (*Check-In*)**.** A single check-in is denoted as $c = \langle u, v, t \rangle$, where u is the user ID, v is the POI ID, and t is the timestamp. Additional information such as category and coordinates can be associated with v.

Definition 2 (*Trajectory*)**.** Set $U = \{u_1, u_2, \cdots, u_i\}$ and $V = \{v_1, v_2, \cdots v_j\}$ are the collections of users and points of interests(POIs) respectively. For user u, let C_u denote all check-ins by u, sorted by time. For each day d, we define the user's day-specific trajectory as $S_u^{(d)} = (c_1, c_2, \ldots, c_{n_{u,d}})$, where c_i denotes the ith check-in of user u on day d, and $n_{u,d}$ is the total number of check-ins by u on day d. Let $S_u = \{S_u^{(d)} \mid d \in D_u\}$ denote all trajectories for user u, where D_u is the set of days that u was active.

Note: Trajectories for the same user are separated by different days (i.e., different values of d). This reflects the start of a new activity cycle and allows our model to capture the daily periodicity and temporal context of user behavior. Modeling multiple trajectories for each user enables the system to account for the periodicity and habitual patterns in user behaviors, and improves the learning and prediction of future movements.

Definition 3 (*Next POI Recommendation*)**.** Given a partial trajectory $S_u^{(d)} = (c_1, c_2, \ldots, c_n)$ of user u on day d, the next POI recommendation task aims to predict the POI that user u is most likely to visit next (i.e., at position $n + 1$).

Definition 4 (*Time-Specific Next POI Recommendation*)**.** Given a trajectory $S_u^{(d)}$ and a target time t^*, predict the POI v^* that user u is most likely to visit at t^*.

5. Methodology

Fig. 2 shows there are four key technical components in LRSA: The influential trajectory collector, the rotational alignment, the fusion prompt design, and the hierarchical prompt tuning. We propose the influential trajectory collector prunes to identify the influential historical trajectories efficiently, and the rotational alignment is introduced to align the text semantics of LLMs with the ID-based semantics of recommendation models, above are the important components of the fusion prompt. The fusion prompt design is conducted to integrate general knowledge and user preference into POI representations. Finally, we propose the hierarchical prompt tuning to enhance the LLMs' learning of the ID-based user preferences.

5.1. Influential trajectory collector

Fine-tuning LLMs with more data does not always lead to better results; data quality is more important than quantity (Zhou et al., 2023). Similarly, a large number of historical check-in trajectories may vary greatly in quality. Inspired by Zhou et al. (2023), we aim to select high-quality historical trajectories to reduce the fine-tuning time of LLMs and improve recommendation performance. However, Zhou et al. (2023) relied on manual experience for filtering and did not provide an automated approach.

Our approach requires computing how the removal of each historical trajectory affects the final recommendation outcome. A straightforward method would be to retrain the entire model after removing each trajectory, but this is too time-consuming. Inspired

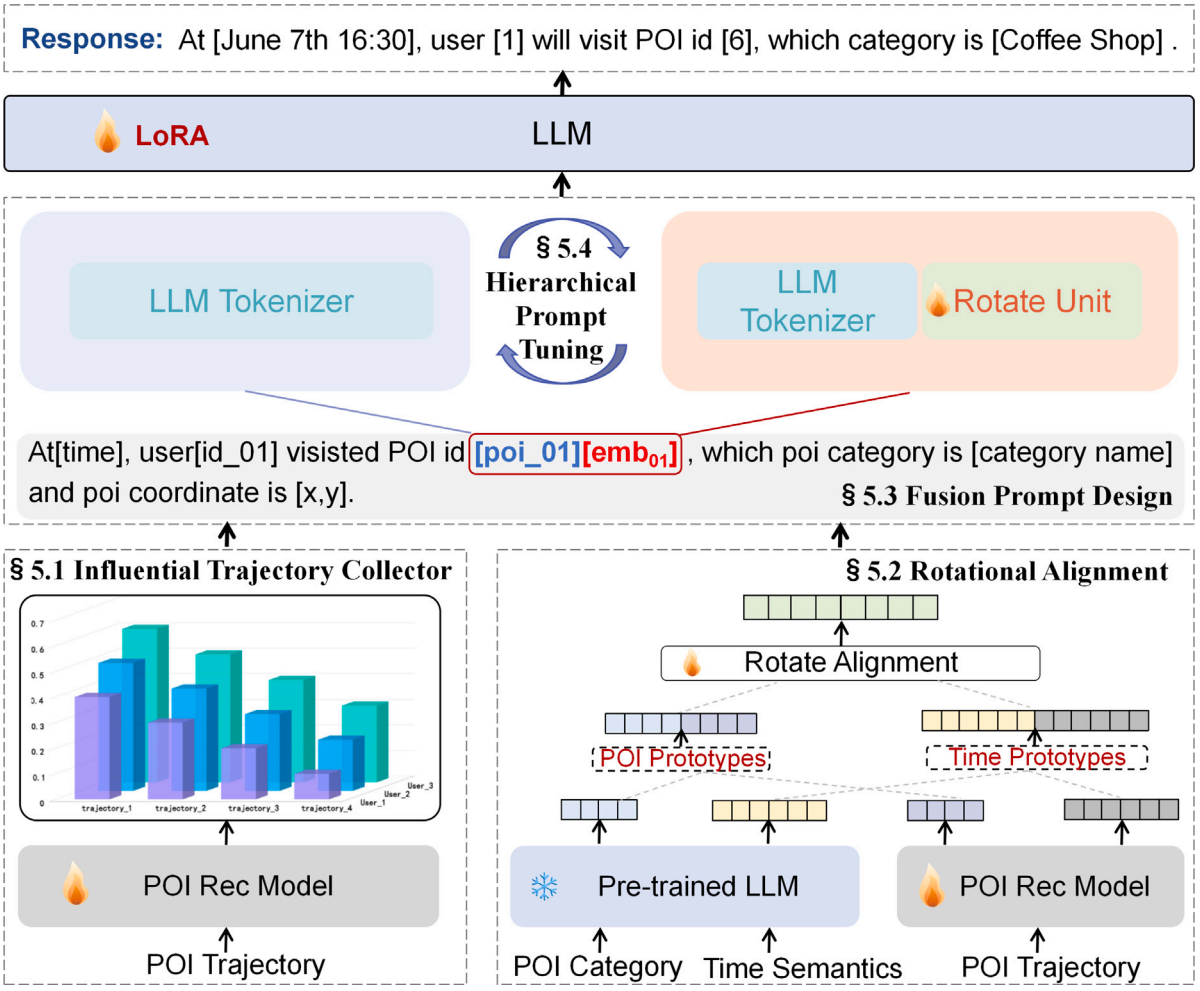

Fig. 2. An overview of the LRSA. The influential trajectory collector prunes to identify the influential historical trajectories efficiently, the rotational alignment aligns the text semantics of LLMs with the ID-based semantics of recommendation models, the fusion prompt design integrates general knowledge and user preference into POI representations, and conducts hierarchical prompt tuning to achieve LLM internal modality alignment.

by the influence function (Koh & Liang, 2017), the influential trajectory collector is proposed to efficiently approximate the change in loss for each trajectory regarding user recommendation results. Based on the change in loss, we compute an influence score, thereby collecting high-quality and influential historical trajectories to serve as candidate trajectories (as part of Section 5.3).

Let us assume the parameters of the traditional POI recommendation model are denoted by θ, the loss function is denoted by $L\left(S_u^i, \theta\right)$, and the training set is denoted as D. First, compute the optimums for $\hat{\theta}$ on the training set:

$$\hat{\theta} \stackrel{\Delta}{=} \arg\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}(S_u^i, \theta) \tag{1}$$

where n is the size of the training set. Next, introduce a disturbance η to S_u, denoted as:

$$\hat{\theta}_{\eta, S_u} \stackrel{\Delta}{=} \arg\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}\left(S_u^i, \theta\right) + \eta \mathcal{L}(S_u, \theta) \tag{2}$$

therefore, when $\eta = -\frac{1}{n}$, $\hat{\theta}_{\eta, S_u}$ is the model parameters when S_u is removed from the training set. When η is small (or when n is large), under the approximation of the first-order Taylor expansion, it is possible to obtain the difference between $\hat{\theta}$ and $\hat{\theta}_{\eta, S_u}$ without retraining the model:

$$\hat{\theta}_{\eta, S_u} - \hat{\theta} \approx -\eta H_{\hat{\theta}}^{-1} \nabla_{\theta} \mathcal{L}(S_u, \hat{\theta}) \tag{3}$$

where $H_\theta = \frac{1}{n}\sum_{i=1}^{n}\nabla^2\mathcal{L}(S_u^i,\theta)$ is the Hessian matrix of the original loss function. After obtaining the impact of S_u on the model parameters, leveraging the chain rule in differentiation allows us to determine the change in predictions for random sample $S_u^\dagger$ by the model. Similarly, the removal of training sample S_u's impact on the loss for random sample $S_u^\dagger$ can be approximated linearly as:

$$S\left(S_u, S_u^\dagger\right) = \mathcal{L}(S_u^\dagger, \hat{\theta}_{\eta,S_u}) - \mathcal{L}(S_u^\dagger,\hat{\theta}) \approx \frac{1}{n}\nabla_\theta\mathcal{L}(S_u^\dagger,\hat{\theta})^{\mathrm{T}} H_{\hat{\theta}}^{-1}\nabla_\theta\mathcal{L}(S_u,\hat{\theta}) \tag{4}$$

The influence score $I_{(S_u,D)}$ for each trajectory on the entire dataset D is calculated as follows:

$$I_{(S_u,D)} = \frac{1}{n}\sum_i \frac{1}{n}\nabla_\theta\mathcal{L}(S_u^i,\hat{\theta})^{\mathrm{T}} H_{\hat{\theta}}^{-1}\nabla_\theta\mathcal{L}(S_u,\hat{\theta}) \tag{5}$$

To further improve the efficiency of obtaining the influence scores for all samples, we utilize Hessian-Vector Products (HVP) (Koh & Liang, 2017) to rewrite the influence score $I_{(S_u,D)}$ as follows:

$$I_{(S_u,D)} = \frac{1}{n}\nabla_\theta\mathcal{L}(S_u^i,\hat{\theta})^{\top} H_{\hat{\theta}}^{-1}\left[\sum_{j=1}^{n}\frac{1}{n}\nabla_\theta\mathcal{L}(S_u^j,\hat{\theta})\right]. \tag{6}$$

Sort the trajectories in the dataset based on the influence score $I_{(S_u,D)}$ in descending order and select the top k trajectories to store in a set, denoted as H_{set}^u.

5.2. Rotational alignment

POI trajectory differs from other sequences in that it contains both ID and numerical data but lacks substantial textual information. While LLMs possess extensive world knowledge, they struggle with effectively handling ID-based data and understanding the underlying semantics. The ID-based POI recommendation models generate embeddings that capture user preferences of POIs. However, they often lack a comprehensive understanding of world knowledge during the recommendation. Therefore, the rotational alignment method is proposed to align and fuse world knowledge with user preferences.

Specifically, we introduce POI prototypes and time prototypes to align POI and time representation separately. As POI category stands as the sole textual feature in check-in datasets (see Section 6.1.1), leveraging LLMs to vectorize POI categories enriches them with ample world knowledge:

$$\left\langle emb_{cat}^i\right\rangle = LLM\_TKZ\left(txt_{cat}^i\right) \tag{7}$$

where $\left\langle emb_{cat}\right\rangle$ is the POI category embedding, and LLM_TKZ is LLM tokenizer, txt_{cat} represents the text of POI category.

We align the POI representations from the ID-based POI recommendation model with POI categories to obtain the POI prototypes:

$$\left\langle emb_{poi}^i\right\rangle = POI\_\mathrm{Rec}\,(i,\theta) \tag{8}$$

$$\left\langle pro\_emb_{poi}^i\right\rangle = Concat\left(\left\langle emb_{cat}^i\right\rangle,\quad \left\langle emb_{poi}^i\right\rangle\right) \tag{9}$$

where POI_Rec is the ID-based POI recommendation model, i and θ represent the poi and the parameters of model, $\left\langle emb_{poi}\right\rangle$ is the POI embeddings from the ID-based POI recommendation model, and $\left\langle pro\_emb_{poi}\right\rangle$ is the POI prototypes.

The ID-based POI recommendation model disperses the time into different time slots for support representations (Chen et al., 2022; Wang et al., 2022; Yin et al., 2023). It differs in granularity from the LLM's representation of time. Specifically, the time semantics is designed to enrich time with additional user-specific features. For instance, transforming the original field 'June 7th 3:30' into 'June 7th [Monday], [dawn] 3:30' enhances the representation of time semantics:

$$\left\langle emb_{ts}^i\right\rangle = LLM\_TKZ\left(txt_{ts}^i\right) \tag{10}$$

where txt_{ts} is the time semantics text, and $\left\langle emb_{ts}\right\rangle$ is the time semantics embeddings. We align the time slots from the ID-based POI recommendation model with time semantics to obtain the time prototypes:

$$\left\langle emb_{time}^i\right\rangle = POI\_\mathrm{Rec}\left(i_{time},\theta\right) \tag{11}$$

$$\left\langle pro\_emb_{time}^i\right\rangle = Concat\left(\left\langle emb_{ts}^i\right\rangle,\quad \left\langle emb_{time}^i\right\rangle\right) \tag{12}$$

where i_{time} is the poi timestamp, $\left\langle emb_{time}\right\rangle$ is the time embeddings from the ID-based POI recommendation model and $\left\langle pro\_emb_{time}\right\rangle$ is the time prototypes.

Previous methods align two representations through projection layer structured and then concatenates them, which increased network complexity and altered the original space (Jin et al., 2024). Inspired by the knowledge graph representation method RotatE (Sun et al., 2019), the rotational alignment is proposed to align and merge POI prototypes and time prototypes without altering the original space.

Specifically, define time prototype $r = \langle pro\_emb_{time} \rangle$, POI prototype $h = \langle pro\_emb_{poi} \rangle$, first, compute the complex form of the time prototype r:

$$\mathrm{Re}(\mathbf{r}) = \cos\left(\frac{\mathbf{r}}{\mathrm{embedding\_range}/\pi}\right) \tag{13}$$

$$\mathrm{Im}(\mathbf{r}) = \sin\left(\frac{\mathbf{r}}{\mathrm{embedding\_range}/\pi}\right) \tag{14}$$

where embedding_range is a scaling constant, $Re(\mathbf{r})$ and $Im(\mathbf{r})$ denote the real and imaginary parts of the time prototype vector in the complex space. Then, perform the rotation operation in the complex space:

$$\mathrm{Re}(\mathbf{s}) = \mathrm{Re}(\mathbf{h}) \cdot \mathrm{Re}(\mathbf{r}) - \mathrm{Im}(\mathbf{h}) \cdot \mathrm{Im}(\mathbf{r}) \tag{15}$$

$$\mathrm{Im}(\mathbf{s}) = \mathrm{Re}(\mathbf{h}) \cdot \mathrm{Im}(\mathbf{r}) + \mathrm{Im}(\mathbf{h}) \cdot \mathrm{Re}(\mathbf{r}) \tag{16}$$

where s is the output after rotational fusion, $Re(\mathbf{h})$ and $Im(\mathbf{h})$ represent the real and imaginary parts of the POI prototype h obtained through tensor chunking. Finally, concatenate the output:

$$\mathrm{RA}(\mathbf{h}, \mathbf{r}) = [\mathrm{Re}(\mathbf{s}), \mathrm{Im}(\mathbf{s})] \tag{17}$$

where $RA(\mathbf{h})$ is the aligned vector, which has the same dimensionality as the POI prototype h.

By treating the temporal information as a rotation vector and encoding temporal features through rotational operations in the complex space, the original spatial structure of the POI prototype h remains unchanged. This avoids the spatial distortion or dimensionality expansion that can occur with traditional methods such as concatenation or addition of temporal embeddings.

5.3. Fusion prompt design

The fusion prompt is designed to integrate world knowledge and user preference into POI representations and guide the LLMs to fine-tune. The motivation of the fusion prompt involves converting discrete IDs and numerical values into textual sentences and incorporating the rotational alignment representation (Section 5.2) within the prompt. As illustrated in Fig. 3a, the fusion prompt encompasses several key elements: (1) the check-in: detailed information of a user check-in, including time, user ID, POI ID, and POI category name. Notably, the representation of the POI ID consists of an ID index and rotational alignment representation.

(2) the current trajectory: provides information on the current trajectory, composed of multiple check-ins. The LLMs will predict the next specific time POI for the current trajectory.

(3) the candidate trajectory: the historical trajectories of the user, composed of multiple check-ins. Selecting a few valuable candidate trajectories H^u_{set} through influential trajectories (Section 5.1).

(4) the instruction: a clear description of the TSNPR task.

(5) the answer: the next check-in information of the current trajectory. It is utilized during training to fine-tune the LLMs and the LLMs are needed to predict the answer during testing.

(6) the temporal acceptance: temporal acceptance information of the user. It indicates the degree to which the user accepts the time intervals between consecutive check-ins.

(7) the periodicity: periodicity information of the user. It indicates the user's historical check-in pattern during the current time period.

To expedite the initial adaptation of the LLMs to check-in trajectory, we have devised a plain text prompt that solely differs from the fusion prompt by excluding the rotational alignment representation ***[emb]*** in the check-in, as illustrated in Fig. 3b.

In the above prompt template, users and POIs are represented using ID-based indices, for instance, (user_01, POI_3456). During input prompt template processing by the LLM tokenizer, partial tokenization occurrences may arise, like ("POI", "_", "34", "56"), potentially leading to the model failing to recognize the correct IDs. To maintain the integrity of ID tokens, we employ whole-word embedding technology (Geng et al., 2022) to treat the IDs in the template as a whole entity.

5.4. Hierarchical prompt tuning

Inspired by superordinate learning (Gagné, 1968). This approach first involves learning low-level knowledge of abstraction. Only after that does it address high-level knowledge of abstraction. The goal is to help with understanding and memory retention of high-level abstract knowledge. We define the hierarchical prompt tuning as a two-stage prompt tuning strategy: the model is first trained with low-level plain text prompts, then further tuned with high-level fusion prompts that incorporate multimodal and user preference information.

Initially, the model learns the plain text prompt at a low level of abstraction that is easy for LLMs to comprehend, enabling rapid assimilation of the concept and content of the TSNPR task. Subsequently, the model learns the fusion prompt at a high level of abstraction to enhance the LLMs' comprehension of multimodal knowledge within the fusion prompt, particularly user preferences within the ID-based POI knowledge.

(a) Fusion prompt	(b) Plain text prompt
check-in: At[time], user[id_01] visisted POI id[poi_01][emb_{01}], which poi category is [category name].	**check-in:** At[time], user[id_01] visisted POI id[poi_01], which poi category is [category name].
Input: Now, the visited trajectory of user[id_01] is [trajectory_01]. There is candidate data :[*candidate trajectory*]. Given the data, at [time], which POI id will user [user id] visit? At [time], user [user id] will visit POI id [poi id], which category is [category name] . Time acceptance of user[id_01] is []. The places visited by user[id_01] in a similar time period are[].	**Input:** Now, the visited trajectory of user[id_01] is [trajectory_01]. There is candidate data :[*candidate trajectory*]. Given the data, at [time], which POI id will user [user id] visit? At [time], user [user id] will visit POI id [poi id], which category is [category name] . Time acceptance of user[id_01] is []. The places visited by user[id_01] in a similar time period are[].

Fig. 3. Illustration of the fusion prompt and the plain text prompt. (a) The fusion prompt represents POIs with the integration of the textual token and the user preference token. (b) The plain text prompt represents POIs with a combination of textual tokens. Different font colors represent different fields: blue indicates the current trajectory, red indicates the candidate trajectory, green indicates the instruction, dark green indicates the answer, purple indicates the temporal acceptance, and orange indicates the periodicity. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Specifically, we define the plain text prompt as low-level text-based knowledge and the fusion prompt as high-level multimodal knowledge. The loss function for the low-level text-based knowledge is as follows:

$$L_{low}(x^{txt}, y^{txt}) = -\sum_{(x^{txt},y^{txt})\in P_{txt}} \sum_{t=1}^{|y^{txt}|} \log\left(P_{\Phi_0+\Delta\Phi(\Theta)}(y_t^{txt}|x^{txt}, y_{<t}^{txt})\right) \tag{18}$$

where $P_{txt} = \left\{\left(x_1^{txt}, y_1^{txt}\right) \ldots \left(x_n^{txt}, y_n^{txt}\right)\right\}$, P_{txt} is the plain text prompt, x^{txt} and y^{txt} are the instruction and the response of LLMs separately, with y_t^{txt} referring to the t-th token of y^{txt}, and $y_{<t}^{txt}$ indicating the tokens preceding y^{txt}. Φ_0 are the parameters of the LLMs, and $\Phi(\Theta)$ are the parameters of the LoRA fine-tuning. The loss function for the high-level text-based multimodal knowledge is as follows:

$$L_{high}(x^f, y^f) = -\sum_{(x^f,y^f)\in P_{fuse}} \sum_{t=1}^{|y^f|} \log\left(P_{\Phi_0+\Delta\Phi(\Theta)+\theta+\psi}(y_t^f|x^f, y_{<t}^f)\right) \tag{19}$$

where $P_{fuse} = \left\{\left(x_1^f, y_1^f\right) \ldots \left(x_n^f, y_n^f\right)\right\}$, P_{fuse} is the fusion prompt, θ are the parameters of the ID-based POI recommendation model, and ψ are the parameters of the rotational alignment.

$$L_{LRSA} = \min_{\Phi_0,\psi}\left(L_{low}\left(x^{txt}, y^{txt}\right) + L_{high}\left(x^f, y^f\right)\right) \tag{20}$$

By first learning low-level text-based knowledge, we prime the initial understanding and capabilities of the LLMs, allowing it to learn in a manner best suited for handling text and grasping the instructions and objectives of the TSNPR task. Subsequently, by learning high-level multimodal knowledge, the LLMs can build upon their existing cognitive structures to better understand user preferences within ID-based POI knowledge. This approach not only leverages user preferences encapsulated in the ID-based POI recommendation model to enhance LLMs but also equips LLMs with the capability to make TSNPR task.

6. Experiments

In this section, we conduct experiments on three publicly available check-in trajectory datasets to verify the authenticity and effectiveness of the model.

6.1. Experimental setup

6.1.1. Datasets

In order to illustrate the efficacy of our proposed model, we conducted experiments utilizing three real-world datasets: NYC, TKY, and Gowalla. The NYC and TKY datasets[1] (Yang et al., 2015) are sourced from the Foursquare platform, capturing users' POI information at locations in New York City and Tokyo. The data collection period spanned from April 2012 to September 2013. The Gowalla dataset[2] (Gong et al., 2023) is from the Gowalla platform, covering a broader geographical area and timeframe, recording

[1] https://sites.google.com/site/yangdingqi/home/foursquare-dataset.
[2] https://drive.google.com/drive/folders/1R87cldpUEMFfRODzloFRiNZohzmOdEmA.

Table 1
Dataset statistics. All per-user statistics are reported as averages computed over all users in each dataset. The 95% confidence interval reflects the variability among users.

Dataset	#User	#Loc	#Trajectory	#Avg. check-ins/ user (95% CI)	#Avg. visits to POI	#Revisit freq.	#Avg. trajectory length
NYC	687	4687	25 472	37.6 (34.3–40.9)	43.7	93.5%	8.4
TKY	2188	7807	188 390	86.2 (82.4–89.9)	162.0	99.7%	6.7
Gowalla	1820	9575	114 903	63.3 (58.1–68.5)	64.7	98.7%	5.4

Table 2
Dataset statistics of different user groups. The 20% of users with the fewest trajectories as the cold-start user subset, the 20% with the most trajectories as the active user subset, and the remaining 60% as the ordinary user subset.

Dataset	#User	#Loc	#Trajectory	#Avg. check-ins/ user (95% CI)	#Avg. visits to POI	#Revisit freq.	#Avg. trajectory length
NYC-active users	137	3002	12 356	90.2 (78.4–102.0)	49.0	99.6%	11.9
NYC-ordinary users	406	3406	11 618	28.8 (26.9–30.8)	18.4	99.3%	5.4
NYC-cold start users	144	1579	1768	12.3 (11.6–13.0)	4.3	96.4%	3.9
TKY-active users	438	6297	89 202	204.1 (192.9–215.4)	133.5	99.9%	9.4
TKY-ordinary users	1312	6675	88 438	67.5 (65.0–70.0)	59.3	99.7%	4.5
TKY-cold start users	438	3913	10 750	24.5 (23.4–25.7)	8.2	96.9%	3.0
Gowalla-active users	387	8594	69 211	190.7 (170.4–210.9)	53.5	99.9%	6.6
Gowalla-ordinary users	1069	8338	40 465	38.0 (35.9–40.1)	18.2	99.5%	3.8
Gowalla-cold start users	364	3666	5227	13.5 (12.9–14.7)	4.1	94.9%	2.9

user check-ins from February 2009 to October 2010. Check-in data is multivariate data, and each check-in entry includes user ID, POI's ID, POI's category, longitude, latitude, and timestamp.

To avoid the interference of outdated data, we sampled only the most recent 200 days. User trajectories were segmented on a daily basis. To reduce data sparsity, we included users with at least 10 check-in records and who have visited at least 10 unique POIs. To provide a detailed overview of the processed datasets, we report, for each dataset, the number of users, the number of POIs, the total number of trajectory, the average number of check-ins per user along with the 95% confidence interval, the average number of times each POI was visited, the revisiting frequency of POIs, and the average trajectory length. The processed data was divided into training, validation, and testing subsets in a ratio of 6:2:2. A summary of the statistics for these processed datasets is presented in Table 1.

6.1.2. Evaluation metrics

For a given test dataset with N samples and the set of POIs V_{set}, the metrics are defined as follows.

$$Acc@1 = \frac{1}{N}(hit@1) \tag{21}$$

$$ValidRatio = \frac{V_g}{V_{set}}, V_g \subseteq V_{set} \tag{22}$$

Acc@1 is used to evaluate the quality of the Top-1 recommendations in a recommendation system, representing the proportion of successful recommendations. The LLMs are essentially generative models. When generating recommendations, they may encounter hallucination issues where the generated recommendations are not present in the dataset. To assess the ability of models to follow instructions, we introduce the valid ratio, V_g is the generating answer of LLMs. A larger value represents better performance.

$$MRR = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{\text{rank}_i} \tag{23}$$

$$NDCG@5 = \begin{cases} \frac{1}{\log_2(rank_i+1)} & , rank_i \le 5 \\ 0 & , rank_i > 5 \end{cases} \tag{24}$$

where rank_i represents the position of POI in the recommendation list. MRR considers the position of the first relevant item returned by the recommendation system in the list of recommendations. NDCG@5 is to evaluate how well highly relevant results are ranked at the top among the first five results.

6.1.3. Baseline models

We select two types of state-of-the-art baseline models for comparison to validate the superiority of our model.

(1) We selected six representative ID-based recommendation methods:

• LightGCN (He et al., 2020): It is a state-of-the-art simplified GNN-based model for representations, where aggregation and propagation are performed without the non-linear activation and feature transformation.

• LSTPM (Sun et al., 2020): It is an LSTM-based model, which develops a nonlocal network and a geo-dilated LSTM to capture long-short-term preferences for users.

• STAN (Luo et al., 2021): It is a state-of-the-art attention-based model that combines spatiotemporal correlation to learn POI recommendations between non-adjacent locations and non-adjacent visits.

• EEDN (Wang et al., 2023): It is a graph convolution model, which exploits rich latent features between users, POIs, and interactions between users and POIs for POI recommendation.

• ROTAN (Feng, Meng et al., 2024): It is a method based on an attention mechanism, which utilizes Time2Rotation to consider temporal information in global collaborative POI transfers.

• MTNet (Huang et al., 2024): It is a method based on an attention mechanism, which proposed a mobility tree network to grasp users' dynamic preferences.

(2) We selected four representative LLM-based recommendation methods:

• Llama2 (Touvron et al., 2023): It is an open-source 7B-parameter language model by Meta, designed for general NLP tasks with improved performance and efficiency over prior versions.

• GPT3.5 (Ouyang et al., 2022): It is an advanced language model from OpenAI, capable of understanding and generating high-quality natural language across various tasks.

• Deepseek (DeepSeek-AI et al., 2025): It is an 8B-parameter language model focused on strong performance and efficient inference for a wide range of NLP applications.

• PEPLER (Li et al., 2023): It investigates explainable recommendation by integrating user and item into pre-trained Transformer models through discrete and continuous prompt learning.

• LLM4POI (Li, de Rijke et al., 2024): It transfers the check-in trajectory to text and fine-tunes the LLMs to achieve the next POI recommendation task.

6.1.4. Implementation details

We select Llama2-7B as the LLM backbone and train for 4 epochs, with a batch size of 1. We set the initial learning rate to 2e−6 and gradually increase the learning rate to 2e−5 by a cosine function scheduler. For the ID-based recommendation model, we adopt ROTAN (Feng, Meng et al., 2024), an efficient and lightweight model based on attention mechanisms. By performing hyperparameter tuning and training on three datasets, we extract the POI embeddings from the model. These embeddings encapsulate and distill user preferences. Specifically, ID-based recommendation model implementations use the Adam optimizer with a learning rate of 0.001, embedding dimension of 2048. The value of the number of candidate trajectories k is half of the number of trajectories in each dataset. For baseline models, we set hyperparameters according to the descriptions of the original papers, and reproduced their results based on the same data splits and evaluation protocols. This ensures that the comparison among all methods is fair and consistent. Our experiments are conducted on servers with Nvidia A100 GPUs.

6.2. Experimental results

Table 3 presents the comparison results of our model with other baseline models. First, across all datasets, our LRSA consistently and significantly outperforms the ID-based recommendation models and the LLM-based models in TSNPR task. Specifically, our approach achieves the highest Acc@1 metric of 0.376, 0.321 and 0.231 on NYC, TKY and Gowalla, respectively. On the NYC, TKY, and Gowalla datasets, compared with the best baseline model, our LRSA method achieves improvements of 5.6% to 12.6% on the Acc@1 metric, 8.7% to 16.2% on the NDCG@5 metric, and 4.5% to 11.9% on the MRR metric. This validates its effective alignment of the ID-based user preference and world knowledge. LRSA achieves a high ValidRatio of over 99% across all datasets, demonstrating the model's ability to follow instructions effectively when generating recommendations.

Second, as for the ID-based recommendation models, their ValidRatio is 1, and their Acc@1, NDCG@5, and MRR scores are lower than LRSA. For example, on the NYC dataset, ROTAN achieves an Acc@1 value of 0.291, which is significantly lower than LRSA's 0.376. These models make predictions solely based on user preferences without incorporating any semantic information about POI. This underscores the importance of integrating world knowledge about POI into the recommendation process.

Third, as for the LLM-based models, We can analyze them from two perspectives. On the one hand, the performance of regular LLMs (such as Llama2, GPT-3.5, and Deepseek) is relatively poor, indicating the crucial importance of adapting LLMs to the TSNPR task to enhance their performance in this domain. On the other hand, the PEPLER and LLM4POI show some improvements over regular LLM methods. However, its recommendation capabilities, as indicated by metrics like Acc@1, NDCG@5, and MRR, still fall short of LRSA. This overlooks the integration of user preference knowledge from traditional ID-based recommendation models. This emphasizes the need for a more comprehensive approach that combines the strengths of LLMs and ID-based recommendation models.

Table 3
The results of LRSA compared with ID-based recommendation methods and LLM-based methods. Bold and underlined indicate the best and the second-best performance, respectively.

Dataset		NYC				TKY				Gowalla			
	Models	Acc@1	NDCG@5	MRR	ValidRatio	Acc@1	NDCG@5	MRR	ValidRatio	Acc@1	NDCG@5	MRR	ValidRatio
ID-based	LightGCN	0.131	0.078	0.199	1.000	0.132	0.082	0.211	1.000	0.067	0.085	0.200	1.000
	LSTPM	0.192	0.253	0.195	1.000	0.189	0.247	0.219	1.000	0.107	0.153	0.192	1.000
	STAN	0.223	0.301	0.322	1.000	0.196	0.188	0.293	1.000	0.110	0.133	0.152	1.000
	EEDN	0.148	0.086	0.182	1.000	0.107	0.097	0.267	1.000	0.117	0.099	0.185	1.000
	ROTAN	0.291	0.373	0.413	1.000	0.235	0.332	0.351	1.000	0.199	0.287	<u>0.291</u>	1.000
	MTNet	0.262	0.368	0.391	1.000	0.251	0.342	<u>0.353</u>	1.000	0.153	0.255	0.229	1.000
LLM-based	Llama2	0.173	0.272	0.241	0.983	0.148	0.228	0.202	0.999	0.109	0.169	0.150	0.999
	GPT3.5	0.088	0.076	0.190	0.478	0.079	0.122	0.153	0.502	0.050	0.065	0.165	0.432
	Deepseek	0.161	0.312	0.238	0.677	0.130	0.251	0.179	0.671	0.099	0.143	0.209	0.629
	PEPLER	0.182	0.279	0.247	0.513	0.148	0.214	0.193	0.675	0.122	0.173	0.158	0.703
	LLM4POI	<u>0.334</u>	<u>0.473</u>	<u>0.429</u>	0.999	<u>0.304</u>	<u>0.364</u>	0.327	0.999	<u>0.207</u>	<u>0.289</u>	0.265	1.000
Ours	LRSA	**0.376**	**0.514**	**0.475**	0.999	**0.321**	**0.423**	**0.395**	1.000	**0.231**	**0.328**	**0.304**	1.000

Table 4
Application of ablation results on the LRSA.

Methods	NYC			TKY			Gowalla		
	Acc@1	NDCG@5	MRR	Acc@1	NDCG@5	MRR	Acc@1	NDCG@5	MRR
w/o ITC	0.359	0.467	0.462	0.310	0.368	0.312	0.219	0.298	0.276
w/o RA_pool	0.343	0.491	0.468	0.295	0.391	0.357	0.204	0.311	0.285
w/o RA_MLP	0.349	0.503	0.471	0.284	0.405	0.382	0.217	0.319	0.293
w/o FPD_text	0.363	0.487	0.471	0.312	0.377	0.325	0.220	0.305	0.298
w/o FPD_prefix	0.373	0.507	0.468	0.296	0.365	0.331	0.230	0.317	0.294
w/o FPD_interlace	0.372	0.510	0.470	0.302	0.418	0.391	0.215	0.316	0.301
w/o HPT	0.365	0.478	0.473	0.313	0.414	0.369	0.226	0.324	0.301
w/o TS	0.369	0.506	0.469	0.312	0.398	0.376	0.225	0.321	0.294
LRSA (ours)	**0.376**	**0.514**	**0.475**	**0.321**	**0.423**	**0.395**	**0.231**	**0.328**	**0.304**

6.3. Ablation study

We validate the effectiveness of each module in LRSA on three datasets: NYC, TKY, and Gowalla. "w/o ITC" refers to removing the influential trajectory collector (ITC) and replacing it with a similarity-based filtering method. "w/o RA_pool" and "w/o RA_MLP" refer to removing the rotational alignment (RA) module and replacing it with max pooling fusion and multi-layer perceptron (MLP), respectively. "w/o TS" denotes removing time semantics (TS) from RA and replacing it with the natural language datetime format. "w/o FPD_text" refers to removing the fusion prompt design (FPD) and replacing it with a pure-text format prompt. "w/o FPD_prefix" refers to removing FPD and replacing it with a vector-prefix format prompt. "w/o FPD_interlace" means removing FPD and replacing it with a vector-text interleaved format prompt. "w/o HPT" refers to removing the hierarchical prompt tuning (HPT) and using LoRA fine-tuning instead.

Table 4 shows the detailed ablation results. For the "w/o ITC" group, after removing the ITC and replacing it with the commonly used similarity-based filtering method, Acc@1 drops by 4.5%, 3.4%, and 5.2% on NYC, TKY, and Gowalla respectively, and total fine-tuning time efficiency drops by 13.5%, 12.5%, and 20.3% respectively (see Section 6.5.5). ITC selects influential trajectories, allowing LRSA to reduce the interference of noisy trajectories, thus improving recommendation accuracy, and also reducing fine-tuning time for greater efficiency.

"w/o RA_pool" and "w/o RA_MLP" are settings where RA is removed and replaced by max pooling fusion and MLP, respectively. For "w/o RA_pool", Acc@1 drops by 8.8%, 8.1%, and 11.7% on NYC, TKY, and Gowalla, while for "w/o RA_MLP", Acc@1 decreases by 7.2%, 11.5%, and 6.1% on NYC, TKY, and Gowalla, respectively. Our proposed RA module encodes time information as rotation operations via geometric transformations, expressing temporal features in complex space while preserving the original embedding space's structure. Max pooling and MLP both alter the embedding space's structure and are less capable of temporal modeling and expression, causing semantic drift when mapping to the LLM embedding space and thus reducing recommendation performance.

"w/o TS" refers to removing TS from RA, replacing it with the natural language datetime format. In the TS format, we add weekday information and time-segment semantics. For example, the original field "June 7th 3:30" becomes the time semantics text "June 7th [Monday], [dawn] 3:30". Acc@1 increases by 1.9%, 2.9%, and 2.7% on NYC, TKY, and Gowalla, respectively. This indicates that the time semantics text enhances the contextual information of the original field, making the time information more comprehensive and accurate, and assists the model in perceiving user preferences associated with POIs at check-in time.

"w/o FPD_text", "w/o FPD_prefix", and "w/o FPD_interlace" refer to using text prompts, vector-prefix prompts, and vector-text interleaved prompts, respectively. With "w/o FPD_text", Acc@1 drops by 3.5%, 2.8%, and 4.8% on NYC, TKY, and Gowalla; with "w/o FPD_prefix", Acc@1 decreases by 0.8%, 7.8%, and 0.4%; and with "w/o FPD_interlace", Acc@1 drops by 1.1%, 5.9%, and

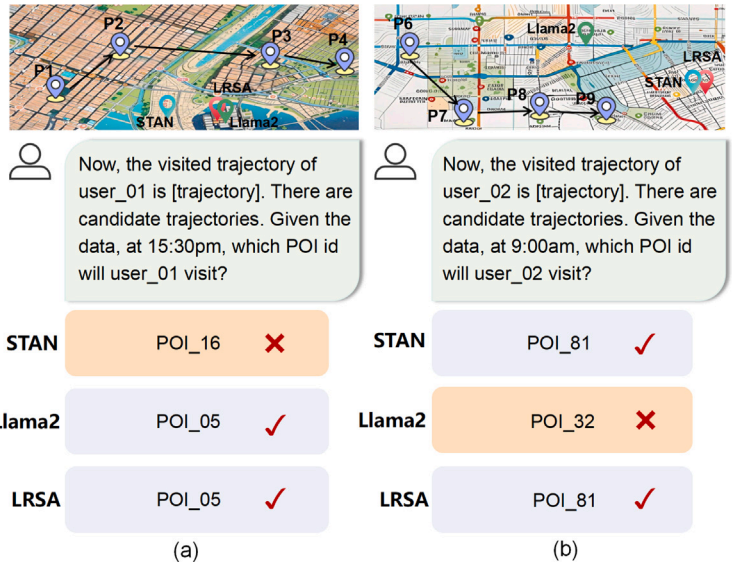

Fig. 4. Illustrative example. (a) LRSA and Llama2 correctly predict that the user will visit POI_05 park by integrating world knowledge. (b) LRSA and STAN correctly predict that the user will visit POI_16 Home by combining user preferences in historical candidate trajectories.

6.9% on the three datasets. All three are less effective than the fusion prompt used in our method. Our analysis suggests that the "w/o FPD_text" template underperforms LRSA mainly because the proposed FPD template can integrate user preference vector-modal information, achieving a stepwise complex training strategy via HPT: pre-training with pure text, then gradually introducing vectorized information, allowing the model to utilize richer context, thus improving understanding and reasoning. "w/o FPD_prefix" and "w/o FPD_interlace" underperform because prefix templates may interrupt continuous text flows, potentially limiting the model's flexibility in adapting to multimodal data. Interleaved templates may increase model comprehension and training difficulty due to complex information layout. With the FPD template used in LRSA, the user preference vector is appended to the end of the POI input—more like a "dynamic" addition—allowing the LLM to more flexibly adapt to user preference input.

"w/o HPT" removes HPT and uses LoRA fine-tuning instead. Acc@1 decreases by 2.9%, 2.5%, and 2.2% on NYC, TKY, and Gowalla, respectively. This shows that HPT, through a two-stage tuning approach, enables the LLM to start from low-level pure text prompts and gradually transition to high-level multimodal fusion prompts. This method not only familiarizes the LLM with recommendation mechanisms via text prompts but, through fusion prompt, helps the LLM internalize user preferences encoded by ID-based models, ultimately enhancing fine-tuning performance of the LLM.

The above analysis confirms that each module provides unique and irreplaceable value. While the individual gains from certain modules appear minor, their collective integration results in significant performance improvements, as shown in Table 4. Moreover, even small improvements can be valuable in challenging benchmarks, and these modules also contribute to the system's robustness and generalization.

6.4. Illustrative example

We select two examples to analyze the impact of world knowledge and user preferences in the TSNPR. As shown in Fig. 4, we choose the answers generated by the three models STAN, Llama2, and LRSA for illustrative examples. In Fig. 4(a), the last 4 check-ins of user_01 (current trajectory) are [p1, p2, p3, p4], where arrows indicate the visitation order. Here, P1 is a miscellaneous shop, P2 is an office building, P3 is a gas station, and P4 is a coffee shop. The ID-based method STAN incorrectly predicts that the user will visit POI_16 factory next solely based on user preferences in historical candidate trajectories, while LRSA and Llama2 correctly predict that the user will visit POI_05 park by integrating world knowledge. In Fig. 4(b), the last 4 check-ins of user_02 (current trajectory) are [p6, p7, p8, p9], where arrows indicate the visitation order. Here, P6 is a bank, P7 is a bookstore, P8 is a restaurant, and P9 is a church. Llama2 incorrectly predicts that the user will visit POI_32 park next by only considering world knowledge, while LRSA and the ID-based method STAN correctly predict that the user will visit POI_16 Home by combining user preferences in historical candidate trajectories. This example illustrates that the fuse of user preference and world knowledge hold importance in TSNPR task.

6.5. Model analysis

6.5.1. Cold-start user analysis

Cold-start user analysis refers to evaluating a model's recommendation performance for user groups with limited historical behavior, reflecting its ability to adapt to cold-start scenarios. We divide users in the entire dataset into three subsets: the 20%

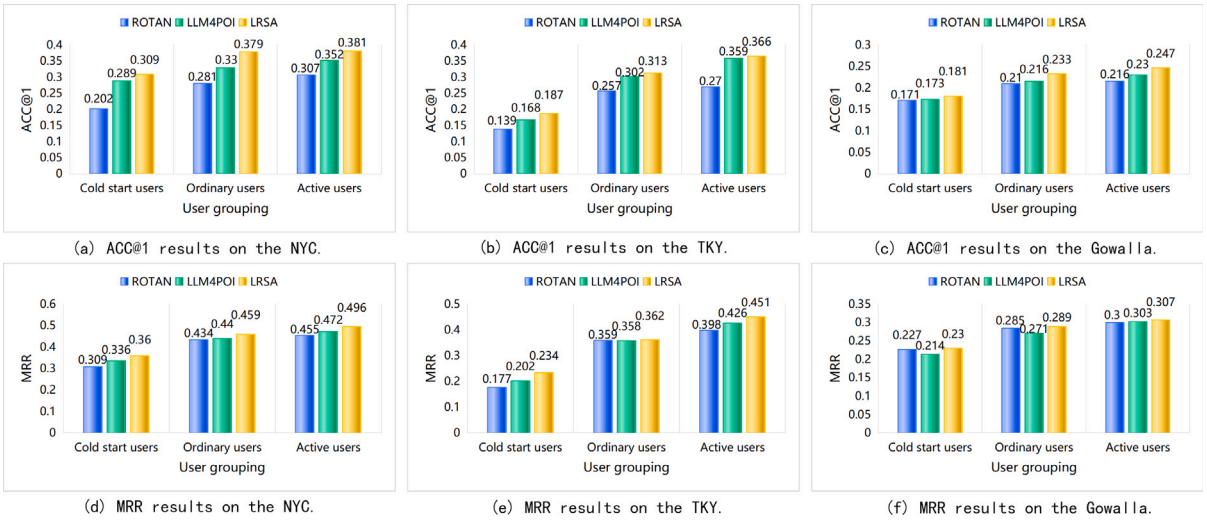

Fig. 5. Result of cold-start user analysis. The 20% of users with the fewest trajectories as the cold-start user subset, the 20% with the most trajectories as the active user subset, and the remaining 60% as the ordinary user subset.

of users with the fewest trajectories as the cold-start user subset, the 20% with the most trajectories as the active user subset, and the remaining 60% as the ordinary user subset. This three-group division aims to better assess the model's robustness across different user groups. We set up three comparison models—ROTAN, LLM4POI, and LRSA—and conduct experiments on the NYC, TKY, and Gowalla datasets, using Acc@1 and MRR as evaluation metrics. A summary of the statistics of different user groups is presented in Table 2.

The detailed results are shown in Fig. 5. In all three datasets, the cold-start user group achieves lower results than the other two groups, indicating that with fewer historical trajectories, it becomes harder to capture users' long-term preferences. However, our proposed LRSA outperforms ROTAN and LLM4POI; compared to LLM4POI, LRSA improves the Acc@1 metric by 6.5%, 10.2%, and 4.4% on the NYC, TKY, and Gowalla, respectively, and improves the MRR metric by 6.7%, 13.7%, and 7.0% on the same datasets. This is because LRSA selects influential key trajectories to extract users' long-term preferences and aligns the semantic space of the LLM, which is particularly important for reducing noise for cold-start users. In Gowalla, the ID-based method ROTAN also achieves impressive results, further demonstrating the rationality and effectiveness of incorporating ID-based user preferences. Compared to the other two methods, LRSA achieves the best performance among the cold-start, ordinary, and active user groups. This is because LRSA integrates ID-based user preferences with world knowledge, giving it better robustness and cold-start adaptability across different user groups, especially cold-start users.

6.5.2. Trajectory length analysis

The current trajectory refers to a user's most recent sequence of POI visits (e.g., the latest N check-ins), which is used as the "current trajectory" for next POI recommendation. The length of the current trajectory determines the amount of short-term preference information available: the longer the trajectory, the richer the short-term interest information the model can utilize; the shorter the trajectory, the less short-term preference can be mined. Trajectory length analysis evaluates the impact of different trajectory lengths on model performance, reflecting the model's ability to capture short-term preferences under varying conditions.

Specifically, we divide the current trajectories into three subsets by length: the top 20% longest trajectories as the long-trajectory subset, the bottom 20% shortest as the short-trajectory subset, and the remaining 60% as the middle-trajectory subset. This setup allows a more comprehensive assessment of the model's robustness to varying trajectory lengths. For each subset, we set up three comparison methods: ROTAN, LLM4POI, and LRSA. Experiments are conducted on the NYC, TKY, and Gowalla, using Acc@1 and MRR as evaluation metrics.

As shown in Fig. 6, across the different datasets, our proposed LRSA consistently achieves the best Acc@1 and MRR scores compared to LLM4POI and ROTAN, especially in the short-trajectory group where information is scarce, Acc@1 is improved over the second-best method by 11.4%, 8.5%, and 18.9% on the NYC, TKY, and Gowalla, respectively, and MRR is improved by 2.9%, 5.2%, and 9.6%. This is mainly because LRSA fully captures users' short-term preferences from the current trajectory, while also integrating long-term preferences from key influential trajectories and broad world knowledge for accurate recommendations. Within-group comparisons show that the medium-trajectory group in the NYC achieves the best results, the long-trajectory group in TKY performs best, and in Gowalla, the short- and medium-trajectory groups perform similarly and both outperform the long-trajectory group. The performance variance of LRSA across all trajectory length groups is small, reflecting its strong robustness in handling trajectories of different lengths.

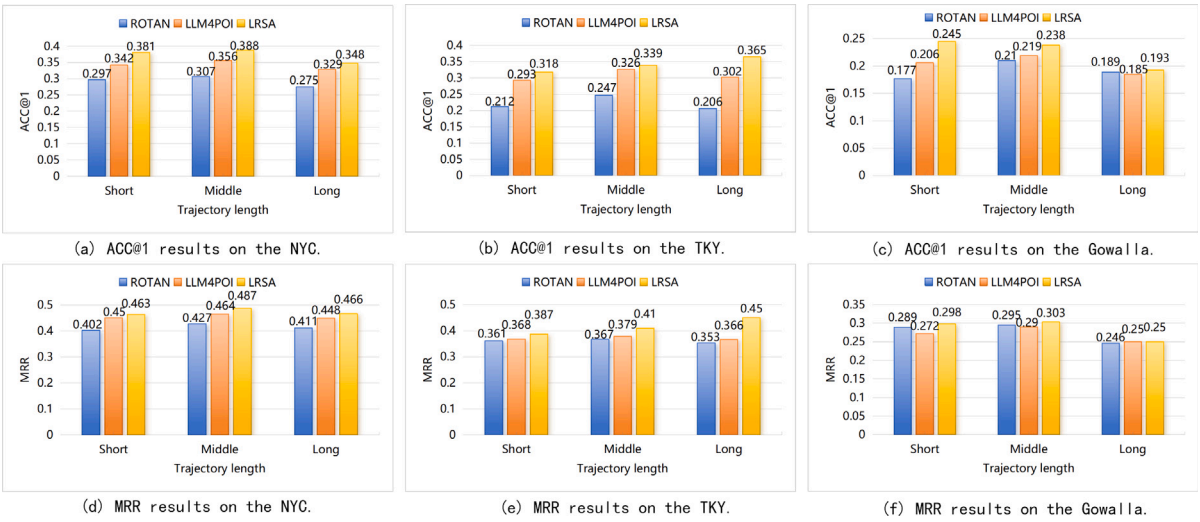

Fig. 6. Result of trajectory length analysis. The top 20% longest trajectories as the long-trajectory subset, the bottom 20% shortest as the short-trajectory subset, and the remaining 60% as the middle-trajectory subset.

Table 5
Application of unseen POI analysis results.

Unseen percent	Methods	NYC		TKY		Gowalla	
		Acc@1	MRR	Acc@1	MRR	Acc@1	MRR
50%	ROTAN	0.259	0.338	0.210	0.284	0.119	0.157
	LLM4POI	0.314	0.360	0.295	0.352	0.189	0.240
	LRSA	**0.346**	**0.388**	**0.308**	**0.367**	**0.203**	**0.264**
80%	ROTAN	0.192	0.278	0.122	0.223	0.105	0.118
	LLM4POI	0.252	0.325	0.207	0.273	0.126	0.172
	LRSA	**0.303**	**0.359**	**0.232**	**0.285**	**0.139**	**0.177**
100%	ROTAN	0.165	0.225	0.097	0.158	0.026	0.178
	LLM4POI	0.248	0.310	0.191	0.236	0.099	0.171
	LRSA	**0.269**	**0.321**	**0.207**	**0.251**	**0.125**	**0.190**

6.5.3. Unseen POI analysis

Unseen POI analysis refers to evaluating the model's generalization ability and cold-start capability when the POIs in the test set do not appear in the training set. Specifically, we set three proportions of unseen POIs in the test set: 50%, 80%, and 100%. This setup aims to comprehensively examine the model's generalization and cold-start handling across various scenarios. Each experiment includes three comparative methods: ROTAN, LLM4POI, and LRSA. Experiments are conducted on the NYC, TKY, and Gowalla datasets, with Acc@1 and MRR as evaluation metrics.

As shown in Table 5, where bold font indicates the best results, LRSA achieves the highest Acc@1 and MRR values across the 50%, 80%, and 100% groups. Specifically, in the 50% group, Acc@1 increases by 4.4% to 10.2% and MRR by 4.3% to 10.0%. In the 80% group, Acc@1 increases by 10.3% to 20.2% and MRR by 2.9% to 10.5%. In the 100% group, Acc@1 improves by 8.4% to 26.3% and MRR by 3.6% to 11.1%. These results illustrate that LRSA fully leverages the respective strengths of LLMs and traditional ID-based models: ID-based user preference information complements the LLM's world knowledge, enabling LRSA to learn general user visit patterns instead of merely memorizing ID information. This allows LRSA to handle new POIs that do not appear in the training set. Furthermore, compared with ROTAN and LLM4POI, LRSA consistently achieves the best performance, demonstrating superior generalization ability and effectiveness in addressing the cold-start problem for unseen POIs.

6.5.4. Cross-domain generalization analysis

Cross-domain transfer is the process of training a model in one domain (the source domain) and evaluating or applying it in a different, distinct domain (the target domain), to assess the model's generalization and adaptability across different data distributions or scenarios. In our experiments, we train LRSA on one dataset and then test it on the other two datasets to evaluate its generalization performance, using Acc@1 as the evaluation metric.

As shown in Fig. 7, for the NYC test set, the models trained on TKY and Gowalla achieve the highest and second-highest Acc@1 values, indicating that the model effectively learn user preferences from the TKY and Gowalla and successfully transfers them to the NYC. For the TKY test set, the model trained on TKY achieves the highest Acc@1 value, while those trained on NYC and Gowalla are

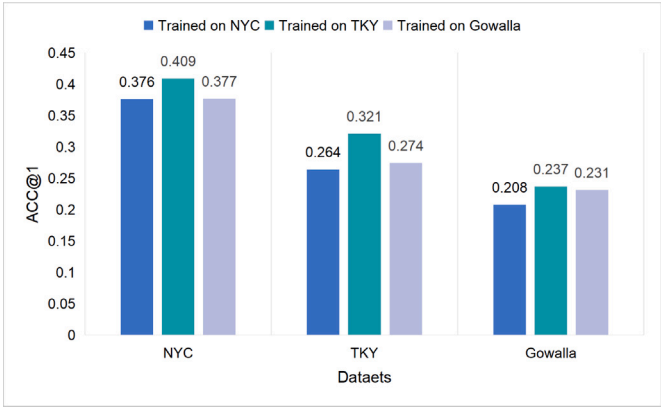

Fig. 7. Cross-domain generalization analysis in NYC, TKY, and Gowalla.

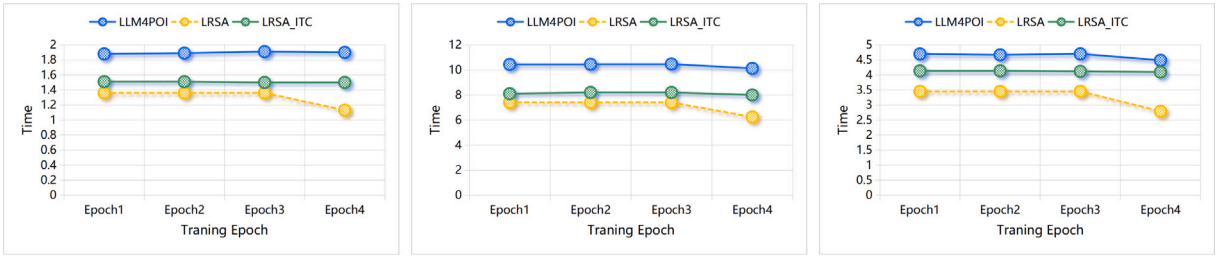

Fig. 8. Results of fine-tuning efficiency in NYC, TKY, and Gowalla.

relatively lower. This is because, compared to NYC and Gowalla, the TKY has a higher frequency of user-POI interactions and richer semantics. For the Gowalla test set, the model trained on TKY achieves the highest Acc@1 value, followed by the model trained on Gowalla, and last is the model trained on NYC. Again, this shows that the TKY, with its high user-POI interaction frequency and semantic richness, enables learned user preferences to be more transferable to other datasets, resulting in the highest Acc@1 values.

These observations essentially demonstrate that our proposed rotational alignment and fusion prompt bridge the gap between LLMs and ID-based models by aligning and integrating ID-based user preferences with the latent space of LLMs. The integrated preferences are then input to the LLM as fusion prompts, and the model is fine-tuned via hierarchical prompt tuning. As a result, the LRSA not only learns user preferences but also fully leverages its world knowledge, thereby exhibiting strong cross-domain knowledge transfer and generalization capabilities.

6.5.5. Fine-tuning efficiency

Fine-tuning efficiency refers to the ability of an LLM-based method to rapidly and efficiently learn from a new dataset within a given time frame. We conduct fine-tuning efficiency experiments on three public datasets: NYC, TKY, and Gowalla. For each dataset, we compare three methods: LLM4POI, our proposed LRSA, and LRSA_ITC (an LRSA variant where the influential trajectory collector (ITC) module is replaced with a similarity-based filtering method). LLM4POI is used as a baseline because it is the first method to perform LLM fine-tuning using check-in trajectories. LRSA_ITC serves as another baseline to verify the effectiveness of the influential trajectory collector (Section 5.1) in improving fine-tuning time efficiency.

As shown in Fig. 8(a), in 4 epochs of fine-tuning on the NYC, LRSA achieves an average fine-tuning time per epoch of 1.30 h, with a total fine-tuning time of 5.21 h. Compared to LLM4POI and LRSA_ITC, LRSA improves average per-epoch fine-tuning time efficiency by 31.6% and 13.9%, and total fine-tuning time efficiency by 31.3% and 13.5%, respectively. In Fig. 8(b), for the TKY, LRSA's average fine-tuning time per epoch over 4 epochs is 7.11 h, with a total fine-tuning time of 28.44 h. Compared to LLM4POI and LRSA_ITC, LRSA improves per-epoch efficiency by 31.4% and 12.5%, and total efficiency by 31.4% and 12.5%. In Fig. 8(c), for Gowalla, the average per-epoch fine-tuning time over 4 epochs is 3.29 h, with a total fine-tuning time of 13.14 h. Compared to LLM4POI and LRSA_ITC, LRSA improves per-epoch efficiency by 29.1% and 20.1%, and total efficiency by 29.2% and 20.3%, respectively.

Overall, compared to LLM4POI and LRSA_ITC, our proposed LRSA achieves lower total fine-tuning time as well as lower per-epoch fine-tuning time. This demonstrates that by using ITC to select key influential trajectories as candidate trajectories, LRSA effectively shortens fine-tuning duration and achieves efficient fine-tuning.

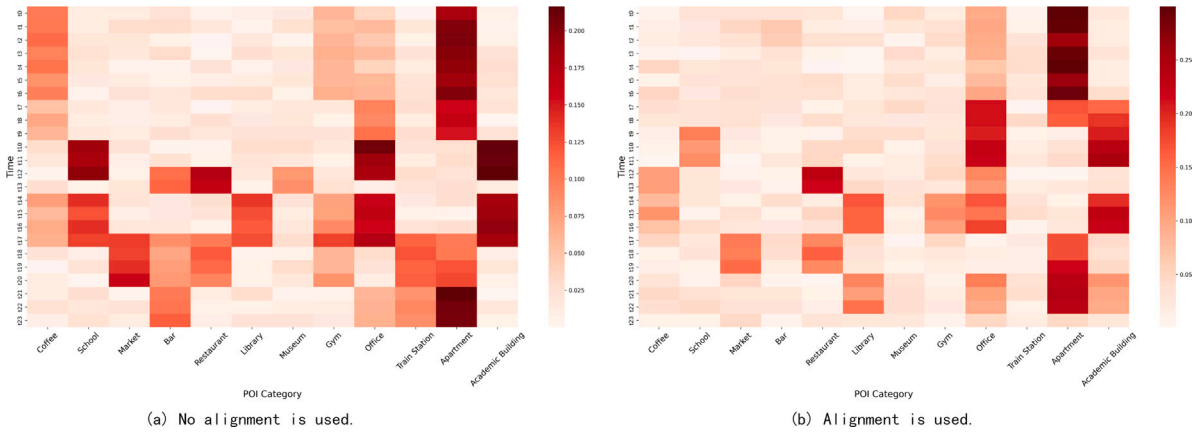

Fig. 9. Visualization of attention weights before and after alignment. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

6.5.6. Interpretability of model behavior and alignment quality

We leverage attention visualization techniques to explore the interpretability of model behavior and alignment quality. To further understand how the LRSA model learns user preferences, we visualize the attention weights from the last layer of the LLM for the same user (user_16), both without and with alignment.

When rotational alignment is not applied and the model is fine-tuned using only plain text, user_16's POI preferences across different time slots are shown in Fig. 9(a). In this case, the overall data exhibits considerable fluctuations, and the distribution of high-activity blocks across POIs and time slots appears rather scattered and even abrupt (for example, categories such as apartment, office, and train station show prominent deep-colored blocks at certain times). These POIs demonstrate extremely high activity during specific periods, but "gaps" or sharp changes emerge at other times and POIs, lacking spatio-temporal continuity.

In contrast, after applying rotational alignment, user user_16's POI preferences across time slots are shown in Fig. 9(b), where the data distribution becomes noticeably smoother. Regions of high values are more contiguous and coherent, with extreme values (such as the deep stripes for certain POIs in Fig. 9(a)) effectively suppressed. The overall pattern reflects a more reasonable spatio-temporal continuity. For instance, high activity segments for apartment from evening to night become smoother and more continuous, indicating more regular behavioral sequences. Additionally, the activity distributions for POIs such as bar, gym, and library become more uniform and distinguishable. Rotational alignment not only suppresses extreme outliers and reduces occasional noise, but also strengthens the expression of the main trends in user preferences. The increased continuity of the same POI across adjacent time slots facilitates clearer extraction and understanding of user preference characteristics.

Based on these observations, we can offer interpretable model behaviors. For example, if the restaurant is the most prominent attribute for user_16 during time slot t_{12}, we should recommend dining venues rather than academic buildings. Similarly, if apartment is the defining attribute for user user_16 during time slot t_{20}, we should recommend residential places rather than markets.

7. Theoretical and practical significance

We summarize the main theoretical and practical contributions of our work as follows:

(1) Theoretical Significance

• LLM-RecSys Alignment (LRSA) Framework: We propose the LRSA framework to align ID-based user preferences with the world knowledge captured by LLMs.

• Influential Trajectory Collector: Inspired by influence functions, an influential trajectory collector is designed to efficiently identify influential user trajectories. This component reduces computational costs and improves the efficiency of fine-tuning LLMs.

• Rotational Alignment Module: We introduce a rotational alignment mechanism to construct user preference representations understandable by LLMs. This technique maps both the LLM token representations and ID-based user preference representations into a unified latent space.

• Fusion Prompt Design: We develop a fusion prompt design method that incorporates the ID-based user preference as a suffix to the POI text, enabling the prompt to jointly encode world knowledge and user preferences.

(2) Practical Significance

• Superior Empirical Performance: Extensive experiments are conducted on multiple public datasets, including NYC, TKY, and Gowalla. The results show that LRSA achieves robust and superior performance compared to state-of-the-art methods. This demonstrates that LRSA is effective in leveraging both ID-based preference modeling and LLM-based world knowledge for the TSNPR task.

Table A.1
Runtime of the influential trajectory collector module on three datasets.

Dataset	NYC	TKY	Gowalla
Runtime (s)	39	115	81

• Bridging Traditional and LLM-based Methods: Our proposed alignment mechanism provides a practical and generalizable approach for integrating LLMs with traditional ID-based recommender systems, potentially guiding future developments in the field and promoting seamless engineering integration.

• Foundations for a Unified Framework: Our work lays the groundwork for a future unified recommendation architecture that leverages both natural language models and traditional recommendation approaches, enabling broader applicability to complex and diverse POI recommendation scenarios.

8. Conclusion

In this paper, we propose an LLM-RecSys Alignment framework, called LRSA, for the TSNPR task. Different from previous methods, we do not directly input ID-based trajectory to the LLMs. In contrast, we use the understanding ability of the ID-based recommendation models to capture the ID-based user preferences, through LRSA to align the user preferences with the semantic space of the LLMs, and we form a fusion prompt for fine-tuning the LLMs to achieve the TSNPR task. Specifically, the influential trajectory collector is designed to identify the influential historical trajectories efficiently. Then, the rotational alignment is proposed to align the semantics of the LLMs with the ID-based recommendation models. Finally, we design the fusion prompt to combine the user preference with the textual features. Moreover, rather than directly input the fusion prompt to the LLMs, we propose the hierarchical prompt tuning to facilitate a two-stage learning process, starting from a low-level plain text prompt to a high-level multimodal fusion prompt for the LLMs. Experimental results on three datasets for the TSNPR task highlight its significant performance.

CRediT authorship contribution statement

Jinhui Zhu: Writing – original draft, Software, Methodology, Data curation, Conceptualization. **Xiangfeng Luo:** Writing – review & editing, Supervision, Project administration, Funding acquisition, Conceptualization. **Xin Yao:** Writing – review & editing, Supervision, Conceptualization. **Xiao Wei:** Validation, Investigation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors thank editors and anonymous reviewers for detailed and constructive feedback capable of improving our paper. This work was supported by the National Key Research and Development Project of China (2021YFC3300602), and the Innovation Research Group Project of the National Natural Science Foundation of China (No. 62421004).

Appendix. Computational cost analysis of the influential trajectory collector

The influential trajectory collector (ITC) incurs a certain amount of computational overhead when selecting high-value trajectories. As shown in Table A.1, the ITC consumes 39 s on the NYC dataset, 115 s on the TKY dataset, and 81 s on the Gowalla dataset. The overall cost is acceptable and does not pose a significant computational burden. In fact, compared to the time required for large-scale model fine-tuning described in Section 6.5.5 (Fine-tuning efficiency), the time spent on ITC is negligible.

In practice, we refer to the commonly used approximation strategies proposed by Koh and Liang (2017), employing stochastic Hessian-Vector Product (HVP) techniques. This approach efficiently approximates the inverse Hessian-vector product via random sampling, significantly reducing computational cost. It allows us to avoid direct computation and storage of the full Hessian matrix while maintaining the effectiveness of the influence estimation, thus making influence function-based sample selection feasible and efficient in real-world settings.

Data availability

Data will be made available on request.

References

An, J., Gao, M., & Tang, J. (2024). MvStHgL: Multi-View hypergraph learning with Spatial-Temporal periodic interests for next POI recommendation. *ACM Transactions on Information Systems*, *42*(6), 1–29.

Chen, W., Wan, H., Guo, S., Huang, H., Zheng, S., Li, J., Lin, S., & Lin, Y. (2022). Building and exploiting spatial–temporal knowledge graph for next POI recommendation. *Knowledge-Based Systems*, *258*(C), http://dx.doi.org/10.1016/j.knosys.2022.109951.

DeepSeek-AI, Guo, D., Yang, D., Zhang, H., Song, J., Zhang, R., Xu, R., Zhu, Q., Ma, S., Wang, P., Bi, X., Zhang, X., Yu, X., Wu, Y., Wu, Z. F., Gou, Z., Shao, Z., Li, Z., Gao, Z., ... Zhang, Z. (2025). DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. URL https://arxiv.org/abs/2501.12948.

Ding, R., Chen, B., Xie, P., Huang, F., Li, X., Zhang, Q., & Xu, Y. (2023). MGeo: Multi-Modal geographic language model Pre-Training. In *Proceedings of the 46th international ACM SIGIR conference on research and development in information retrieval* (pp. 185–194). New York, NY, USA: Association for Computing Machinery, http://dx.doi.org/10.1145/3539618.3591728.

Feng, S., Lyu, H., Li, F., Sun, Z., & Chen, C. (2024). where to move next: Zero-shot generalization of LLMs for next POI recommendation. In *2024 IEEE conference on artificial intelligence* (pp. 1530–1535). Los Alamitos, CA, USA: IEEE Computer Society, URL https://doi.ieeecomputersociety.org/10.1109/CAI59869.2024.00277.

Feng, S., Meng, F., Chen, L., Shang, S., & Ong, Y. S. (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* (pp. 759–770). New York, NY, USA: Association for Computing Machinery, http://dx.doi.org/10.1145/3637528.3671809.

Gagné, R. M. (1968). Reviews: Ausubel, david p. Educational psychology: A cognitive view. New York: Holt, Rinehart & Winston, 1968. 685 + xviii pp. $8.95. *American Educational Research Journal*, *6*(2), 287–290. http://dx.doi.org/10.3102/00028312006002287.

Gan, M., & Ma, Y. (2023). Mapping user interest into hyper-spherical space: A novel POI recommendation method. *Information Processing & Management*, *60*(2), Article 103169. http://dx.doi.org/10.1016/j.ipm.2022.103169.

Geng, S., Liu, S., Fu, Z., Ge, Y., & Zhang, Y. (2022). Recommendation as language processing (RLP): A unified pretrain, personalized prompt & predict paradigm (P5). In *Proceedings of the 16th ACM conference on recommender systems* (pp. 299–315). New York, NY, USA: Association for Computing Machinery, http://dx.doi.org/10.1145/3523227.3546767.

Gong, L., Lin, Y., Guo, S., Lin, Y., Wang, T., Zheng, E., Zhou, Z., & Wan, H. (2023). *AAAI'23/iAAI'23/eAAI'23*, *Contrastive pre-training with adversarial perturbations for check-in sequence representation learning*. AAAI Press, http://dx.doi.org/10.1609/aaai.v37i4.25546.

Guo, Q., Sun, Z., Zhang, J., & Theng, Y.-L. (2020). An attentional recurrent neural network for personalized next location recommendation. *Proceedings of the AAAI Conference on Artificial Intelligence*, *34*(01), 83–90, URL https://ojs.aaai.org/index.php/AAAI/article/view/5337.

Han, Q., Yoshikawa, A., & Yamamura, M. (2025). Adapting large language model for Spatio-Temporal understanding in next Point-of-Interest prediction. In *ICASSP 2025 - 2025 IEEE international conference on acoustics, speech and signal processing* (pp. 1–5). http://dx.doi.org/10.1109/ICASSP49660.2025.10889866.

He, X., Deng, K., Wang, X., et al. (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* (pp. 639–648). New York, NY, USA: Association for Computing Machinery, http://dx.doi.org/10.1145/3397271.3401063.

He, R., & McAuley, J. (2016). Fusing similarity models with Markov chains for sparse sequential recommendation. In *2016 IEEE 16th international conference on data mining* (pp. 191–200). Los Alamitos, CA, USA: IEEE Computer Society, URL https://doi.ieeecomputersociety.org/10.1109/ICDM.2016.0030.

Huang, T., Pan, X., Cai, X., Zhang, Y., & Yuan, X. (2024). Learning time slot preferences via mobility tree for next POI recommendation. In *AAAI'24/iAAI'24/eAAI'24*, *Proceedings of the thirty-eighth AAAI conference on artificial intelligence and thirty-sixth conference on innovative applications of artificial intelligence and fourteenth symposium on educational advances in artificial intelligence*. AAAI Press, http://dx.doi.org/10.1609/aaai.v38i8.28697.

Jin, M., Wang, S., Ma, L., Chu, Z., et al. (2024). Time-LLM: Time series forecasting by reprogramming large language models. In *The twelfth international conference on learning representations*. URL https://openreview.net/forum?id=Unb5CVPtae.

Kim, B., Hong, T., Ko, Y., & Seo, J. (2020). Multi-Task learning for knowledge graph completion with Pre-trained language models. In D. Scott, N. Bel, & C. Zong (Eds.), *Proceedings of the 28th international conference on computational linguistics* (pp. 1737–1743). Barcelona, Spain (Online): International Committee on Computational Linguistics, URL https://aclanthology.org/2020.coling-main.153/.

Koh, P. W., & Liang, P. (2017). Understanding black-box predictions via influence functions. In *Proceedings of the 34th international conference on machine learning - volume 70* (pp. 1885–1894). JMLR.org, URL https://proceedings.mlr.press/v70/koh17a.

Li, S., Chen, W., Wang, B., Huang, C., Yu, Y., & Dong, J. (2024). MCN4Rec: Multi-level collaborative neural network for next location recommendation. *ACM Transactions on Information Systems*, *42*(4), http://dx.doi.org/10.1145/3643669.

Li, P., de Rijke, M., Xue, H., Ao, S., Song, Y., & Salim, F. D. (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* (pp. 1463–1472). New York, NY, USA: Association for Computing Machinery, http://dx.doi.org/10.1145/3626772.3657840.

Li, L., Zhang, Y., & Chen, L. (2023). Personalized prompt learning for explainable recommendation. *ACM Transactions on Information Systems*, *41*(4), http://dx.doi.org/10.1145/3580488.

Liao, J., Li, S., Yang, Z., Wu, J., Yuan, Y., Wang, X., & He, X. (2024). LLaRA: Large Language-Recommendation assistant. In *Proceedings of the 47th international ACM SIGIR conference on research and development in information retrieval* (pp. 1785–1795). New York, NY, USA: Association for Computing Machinery, http://dx.doi.org/10.1145/3626772.3657690.

Lin, X., Wang, W., Li, Y., Yang, S., Feng, F., Wei, Y., & Chua, T.-S. (2024). Data-efficient Fine-tuning for LLM-based recommendation. In *Proceedings of the 47th international ACM SIGIR conference on research and development in information retrieval* (pp. 365–374). New York, NY, USA: Association for Computing Machinery, http://dx.doi.org/10.1145/3626772.3657807.

Luo, Y., Liu, Q., & Liu, Z. (2021). STAN: Spatio-Temporal attention network for next location recommendation. In *Proceedings of the web conference 2021* (pp. 2177–2185). New York, NY, USA: Association for Computing Machinery, http://dx.doi.org/10.1145/3442381.3449998.

Lyu, H., Jiang, S., Zeng, H., Xia, Y., Wang, Q., Zhang, S., Chen, R., Leung, C., Tang, J., & Luo, J. (2024). LLM-Rec: Personalized recommendation via prompting large language models. In K. Duh, H. Gomez, & S. Bethard (Eds.), *Findings of the association for computational linguistics: NAACL 2024* (pp. 583–612). Mexico City, Mexico: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/2024.findings-naacl.39.

Ouyang, L., Wu, J., Jiang, X., et al. (2022). Training language models to follow instructions with human feedback. URL https://arxiv.org/abs/2203.02155.

Qin, Y., Wu, H., Ju, W., Luo, X., & Zhang, M. (2023). A diffusion model for POI recommendation. *ACM Transactions on Information Systems*, *42*(2), http://dx.doi.org/10.1145/3624475.

Sun, Z., Deng, Z.-H., Nie, J.-Y., & Tang, J. (2019). Rotate: Knowledge graph embedding by relational rotation in complex space. In *International conference on learning representations*. URL https://openreview.net/forum?id=HkgEQnRqYQ.

Sun, K., Li, C., & Qian, T. (2024). City Matters! A dual-target Cross-City sequential POI recommendation model. *ACM Transactions on Information Systems*, *42*(6), http://dx.doi.org/10.1145/3664284.

Sun, K., Qian, T., Chen, T., Liang, Y., Nguyen, Q. V. H., & Yin, H. (2020). Where to go next: Modeling long- and Short-Term user preferences for Point-of-Interest recommendation. *Proceedings of the AAAI Conference on Artificial Intelligence*, *34*(01), 214–221. http://dx.doi.org/10.1609/aaai.v34i01.5353.

Tan, J., Xu, S., Hua, W., Ge, Y., Li, Z., & Zhang, Y. (2024). IDGenRec: LLM-RecSys alignment with textual ID learning. In *Proceedings of the 47th international ACM SIGIR conference on research and development in information retrieval* (pp. 355–364). New York, NY, USA: Association for Computing Machinery, http://dx.doi.org/10.1145/3626772.3657821.

Tian, J., Wang, Z., Zhao, J., & Ding, Z. (2024). MMREC: LLM based Multi-Modal recommender system. In *2024 19th international workshop on semantic and social media adaptation & personalization* (pp. 105–110).

Touvron, H., Martin, L., Stone, K., et al. (2023). Llama 2: Open foundation and Fine-Tuned chat models. URL https://arxiv.org/abs/2307.09288.

Wang, X., Chen, H., Pan, Z., Zhou, Y., Guan, C., Sun, L., & Zhu, W. (2025). Automated disentangled sequential recommendation with large language models. *ACM Transactions on Information Systems*, *43*(2), http://dx.doi.org/10.1145/3675164.

Wang, X., Fukumoto, F., Cui, J., et al. (2023). EEDN: Enhanced Encoder-Decoder network with local and global context learning for POI recommendation. In *Proceedings of the 46th international ACM SIGIR conference on research and development in information retrieval* (pp. 383–392). New York, NY, USA: Association for Computing Machinery, http://dx.doi.org/10.1145/3539618.3591678.

Wang, X., Sun, G., Fang, X., Yang, J., & Wang, S. (2022). Modeling Spatio-temporal neighbourhood for personalized Point-of-interest Recommendation. In L. D. Raedt (Ed.), *Proceedings of the thirty-first international joint conference on artificial intelligence, IJCAI-22* (pp. 3530–3536). International Joint Conferences on Artificial Intelligence Organization, http://dx.doi.org/10.24963/ijcai.2022/490.

Wu, Z., Sun, Z., Wang, D., Zhang, L., Zhang, J., & Ong, Y. S. (2024). MRP-LLM: Multitask reflective large language models for privacy-preserving next POI recommendation. URL https://arxiv.org/abs/2412.07796.

Xu, S., Hua, W., & Zhang, Y. (2024). OpenP5: An open-source platform for developing, training, and evaluating LLM-based recommender systems. In *Proceedings of the 47th international ACM SIGIR conference on research and development in information retrieval* (pp. 386–394). New York, NY, USA: Association for Computing Machinery, http://dx.doi.org/10.1145/3626772.3657883.

Yang, S., Liu, J., & Zhao, K. (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* (pp. 1144–1153). New York, NY, USA: Association for Computing Machinery, http://dx.doi.org/10.1145/3477495.3531983.

Yang, D., Zhang, D., Zheng, V. W., et al. (2015). Modeling user activity preference by leveraging user spatial temporal characteristics in LBSNs. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *45*(1), 129–142. http://dx.doi.org/10.1109/TSMC.2014.2327053.

Ye, H., Zhang, N., Chen, H., & Chen, H. (2022). Generative knowledge graph construction: A review. In Y. Goldberg, Z. Kozareva, & Y. Zhang (Eds.), *Proceedings of the 2022 conference on empirical methods in natural language processing* (pp. 1–17). Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/2022.emnlp-main.1.

Yin, F., Liu, Y., Shen, Z., Chen, L., Shang, S., & Han, P. (2023). Next POI recommendation with dynamic graph and explicit dependency. In *AAAI'23/iAAI'23/eAAI'23, Proceedings of the thirty-seventh AAAI conference on artificial intelligence and thirty-fifth conference on innovative applications of artificial intelligence and thirteenth symposium on educational advances in artificial intelligence*. AAAI Press, http://dx.doi.org/10.1609/aaai.v37i4.25608.

Zeng, J., Tao, H., Tang, H., Wen, J., & Gao, M. (2025). Global and local hypergraph learning method with semantic enhancement for POI recommendation. *Information Processing & Management*, *62*(1), Article 103868. http://dx.doi.org/10.1016/j.ipm.2024.103868.

Zhang, W., Lai, X., & Wang, J. (2023). Social link inference via multiview matching network from spatiotemporal trajectories. *IEEE Transactions on Neural Networks and Learning Systems*, *34*(4), 1720–1731. http://dx.doi.org/10.1109/TNNLS.2020.2986472.

Zhang, Y., Li, Y., & Ji, W. (2023). A Trajectory-Based user movement pattern similarity measure for user identification. *IEEE Transactions on Network Science and Engineering*, *10*(6), 3834–3845. http://dx.doi.org/10.1109/TNSE.2023.3274516.

Zhang, Z., Liu, S., Liu, Z., Zhong, R., et al. (2025). LLM-Powered user simulator for recommender system. *Proceedings of the AAAI Conference on Artificial Intelligence*, *39*(12), 13339–13347. http://dx.doi.org/10.1609/aaai.v39i12.33456.

Zhang, L., Sun, Z., Wu, Z., Zhang, J., Ong, Y. S., & Qu, X. (2022). Next Point-of-Interest recommendation with inferring Multi-step future preferences. In L. D. Raedt (Ed.), *Proceedings of the thirty-first international joint conference on artificial intelligence, IJCAI-22* (pp. 3751–3757). International Joint Conferences on Artificial Intelligence Organization, http://dx.doi.org/10.24963/ijcai.2022/521.

Zhao, Y., Wu, J., Wang, X., Tang, W., Wang, D., & de Rijke, M. (2024). Let me do it for you: Towards LLM empowered recommendation via tool learning. In *Proceedings of the 47th international ACM SIGIR conference on research and development in information retrieval* (pp. 1796–1806). New York, NY, USA: Association for Computing Machinery, http://dx.doi.org/10.1145/3626772.3657828.

Zhao, K., Zhang, Y., Yin, H., Wang, J., Zheng, K., Zhou, X., & Xing, C. (2020). Discovering subsequence patterns for Next POI recommendation. In C. Bessiere (Ed.), *Proceedings of the twenty-ninth international joint conference on artificial intelligence, IJCAI-20* (pp. 3216–3222). International Joint Conferences on Artificial Intelligence Organization, http://dx.doi.org/10.24963/ijcai.2020/445.

Zhou, C., Liu, P., Xu, P., Iyer, S., Sun, J., Mao, Y., Ma, X., Efrat, A., Yu, P., Yu, L., Zhang, S., Ghosh, G., Lewis, M., Zettlemoyer, L., & Levy, O. (2023). LIMA: less is more for alignment. In *Proceedings of the 37th international conference on neural information processing systems*. Red Hook, NY, USA: Curran Associates Inc., http://dx.doi.org/10.5555/3666122.3668522.

Zhu, Y., Li, H., Liao, Y., Wang, B., Guan, Z., Liu, H., & Cai, D. (2017). What to do next: Modeling user behaviors by time-LSTM. In *Proceedings of the twenty-sixth international joint conference on artificial intelligence, IJCAI-17* (pp. 3602–3608). http://dx.doi.org/10.24963/ijcai.2017/504.