Hierarchical Reinforcement Learning for Point of Interest Recommendation

Yanan Xiao[1,6] , **Lu Jiang**[2] , **Kunpeng Liu**[3] , **Yuanbo Xu**[4,7] ,
Pengyang Wang[5,8*] and **Minghao Yin**[1,6*]
[1]School of Computer Science and Information Technology, Northeast Normal University, China
[2]Department of Information Science and Technology, Dalian Maritime University, China
[3]Department of Computer Science, Portland State University, USA
[4]College of Computer Science and Technology, Jilin University, China
[5]Department of Computer and Information Science, University of Macau, China
[6]Key Laboratory of Applied Statistics of MOE, Northeast Normal University, China
[7]Mobile Intelligent Computing (MIC) Lab, Jilin University, China
[8]The State Key Laboratory of Internet of Things for Smart City, University of Macau, China
{xiaoyn117, ymh}@nenu.edu.cn, jiangl761@dlmu.edu.cn, yuanbox@jlu.edu.cn,
kunpeng@pdx.edu, pywang@um.edu.mo

Abstract

With the increasing popularity of location-based services, accurately recommending points of interest (POIs) has become a critical task. Although existing technologies are proficient in processing sequential data, they fall short when it comes to accommodating the diversity and dynamism in users' POI selections, particularly in extracting key signals from complex historical behaviors. To address this challenge, we introduced the Hierarchical Reinforcement Learning Preprocessing Framework (HRL-PRP), a framework that can be integrated into existing recommendation models to effectively optimize user profiles. The HRL-PRP framework employs a two-tiered decision-making process, where the high-level process determines the necessity of modifying profiles, and the low-level process focuses on selecting POIs within the profiles. Through evaluations of multiple real-world datasets, we have demonstrated that HRL-PRP surpasses existing state-of-the-art methods in various recommendation performance metrics.

1 Introduciton

With the popularity of location-based services (LBS), point-of-interest (POI) recommendations have become an important tool for users to navigate [Wang *et al.*, 2023b] and explore cities [Qin *et al.*, 2023]. Recommendation systems need to extract learning from users' historical location data to provide accurate recommendation services. However, given the diversity and ever-changing nature of users' points of interest, it becomes a major challenge to filter out signals with predictive value from numerous historical data. Therefore, there is an urgent need to develop a new recommendation paradigm that can deeply understand adapt to users' behavioral patterns to improve recommendation accuracy.

Early POI recommendation research focused on sequential behavioral influence, using models such as recurrent neural networks (RNN) [Xu *et al.*, 2022; Huang *et al.*, 2020], long short-term memory (LSTM) [Liu *et al.*, 2020; Luo *et al.*, 2021] and gated recurrent units (GRU) [Manotumruksa *et al.*, 2020]. With the increasingly available location-based social networks (LBSNs), research began to fuse geographic [Sun *et al.*, 2020], semantic [Wu *et al.*, 2019], temporal [Doan *et al.*, 2019], and other multidimensional information [Feng *et al.*, 2020] to improve the understanding of user behavior, but introducing additional model complexity. However, the fidelity of the model is limited by a key assumption: all historical POIs have the same influence in estimating the similarity between user preferences and target POIs. This assumption may lead to ignoring the unique contributions of different historical POIs in the prediction process. Therefore, it becomes critical to introduce an attention mechanism to distinguish the impact of historical check-ins. For example, attention-based models such as NAIS [He *et al.*, 2018] and NASR [Li *et al.*, 2017] evaluate the attention coefficient of each historical POI to determine its importance in recommending target check-ins.

Attention-based POI recommendation models, while improving performance, still face challenges such as the dilution effect caused by users' diverse check-in histories. In a user's check-in history, visits that truly reflect interest in a target POI may be obfuscated by a large number of irrelevant POIs, thus weakening the impact of key POIs. As show in the Figure 1 illustrates the recommendation results of the NAIS model, where the score of each historical POI reflects its attention factor. Key historical POIs such as "Shopping Centers", "Shoe Stores", and "Jewelry Stores", despite receiving high attention factors, are diluted in importance by other categories of POIs when all historical POI scores are aggregated. For example, POIs such as "Coffee Shops", "In-

*Corresponding author.

ternational Hotels", and "Cinema" reduce the impact of the main historical POIs. In addition, the model assigns an attention factor to each historical POI, including POIs that are not related to the predicted target, such as "gym", which may result in these random POIs being ranked higher than the actual target POIs. Therefore, even non-critical historical POIs assigned a low attention factor may have a negative impact on the predicted results. Overall, there is a need to distinguish key points of interest from non-key points of interest in a more effective manner, thereby improving the accuracy and usefulness of recommendation systems.

To address the above issues, we propose to modify user profiles by removing noisy POIs, rather than assigning attention to each POI to improve the accuracy of the model. The key challenge is that we lack explicit guidance for identifying and removing noisy POIs from historical data. To this end, we introduce a **H**ierarchical **R**einforcement **L**earning **PR**e**P**rocessing framework (HRL-PRP). HRL-PRP build profile modifiers through a hierarchical decision-making process: in high-level decision-making, a high-level agent decides whether a profile needs to be modified; in low-level decision-making, a low-level agent decides which POIs to remove. The two levels of tasks alternate in an environment where the dataset and the pre-trained base recommendation model provide feedback. Essentially, the profile modifier and the base recommendation model are trained jointly.

Our contributions are threefold:

- We introduce an auxiliary POI recommendation model consisting of a profile modifier and a basic recommendation model. The joint training of these two models effectively removes the noisy POI from user profiles, resulting in higher prediction accuracy.
- A hierarchical reinforcement learning framework has been developed that is capable of recognizing and removing irrelevant information in an unsupervised manner, thus enhancing the recommendation process without the need for explicit labeling.
- After extensive experiments on a large number of real datasets, our model proves its effectiveness. When generalized to current mainstream recommender system models, the model achieves significant improvements in key performance metrics.

2 Definitions and Problem Formulation

This section provides foundational definitions to establish the context for our study on improving POI recommendation systems using hierarchical reinforcement learning.

2.1 Definitions

User Profile. Given user u $U = \{u_1, u_2, \ldots, u_{|U|}\}$, a series of POIs $P = \{p_1, p_2, \ldots, p_{|P|}\}$. Each user profile sequence is expressed as $\mathcal{E}^u = (p_1^u, p_2^u, \ldots, p_{t_u}^u)$. Where p_t^u represents the POI visited by the specified user u at time t, $p_{t_u}^u$ denotes the last visited POI.

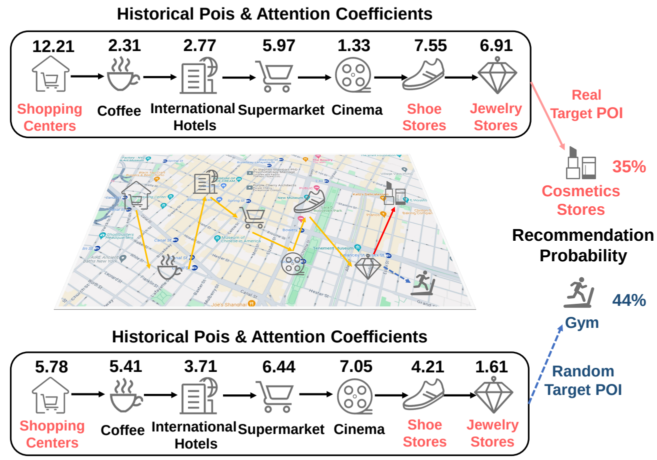


Figure 1: Example of poi recommendation motivation. The scores at the top of the history poi are mainly the attention coefficients computed by the NAIS, and the right side of the target poi is the recommendation probability predicted by the NAIS. The goal of this paper is to remove the poi that contributes the least to the prediction.

2.2 Problem Formulation

The primary objective of this research is to develop an optimization function that maximizes the probability of accurately predicting the next POI p a user will visit. This involves refining a user's original profile $\mathcal{E}^u$ to focus more effectively on the POI that are most influential for future predictions. The refined profile can be expressed as

$$\hat{\mathcal{E}}^u = \mathcal{F}(\mathcal{E}^u), \tag{1}$$

where $\mathcal{F}$ comes from the process function aimed at refining the profile, and $\hat{\mathcal{E}}^u$ represents the refined profile.

Given $\mathcal{E}^u$, our task is to predict the POI $p_{t_u+1}^u$ that the user is most likely to visit next. Mathematically, this is formulated as

$$\text{argmax}_{p \in P} \, \mathbf{P}(y = 1 | \mathcal{E}^u, p_{t_u+1}^u), \tag{2}$$

where $\mathbf{P}(y = 1 | \mathcal{E}^u, p_{t_u+1}^u)$ represents the conditional probability that the user visits POI p, $y = 1$ indicates that the prediction matches the actual next POI visited by the user. In contrast, $y = 0$ means that there is no match.

To evaluate the effectiveness of the predictive model, we use the following accuracy metric define as

$$\text{Accuracy} = \frac{1}{|U|} \sum_{u \in U} \text{argmax}_{p \in P} \, \mathbf{P}(y = 1 | \hat{\mathcal{E}}^u, p_{t_u+1}^u). \tag{3}$$

The efficacy of the function $\mathcal{F}$ is quantified by calculating the average maximum probability that the predicted POIs correspond to the POIs actually visited by users.

3 Method

In this section, we first give an overview of the proposed model, then we introduce a hierarchal reinforcement learning algorithm to revise user profiles and finally present the training process of the entire model.

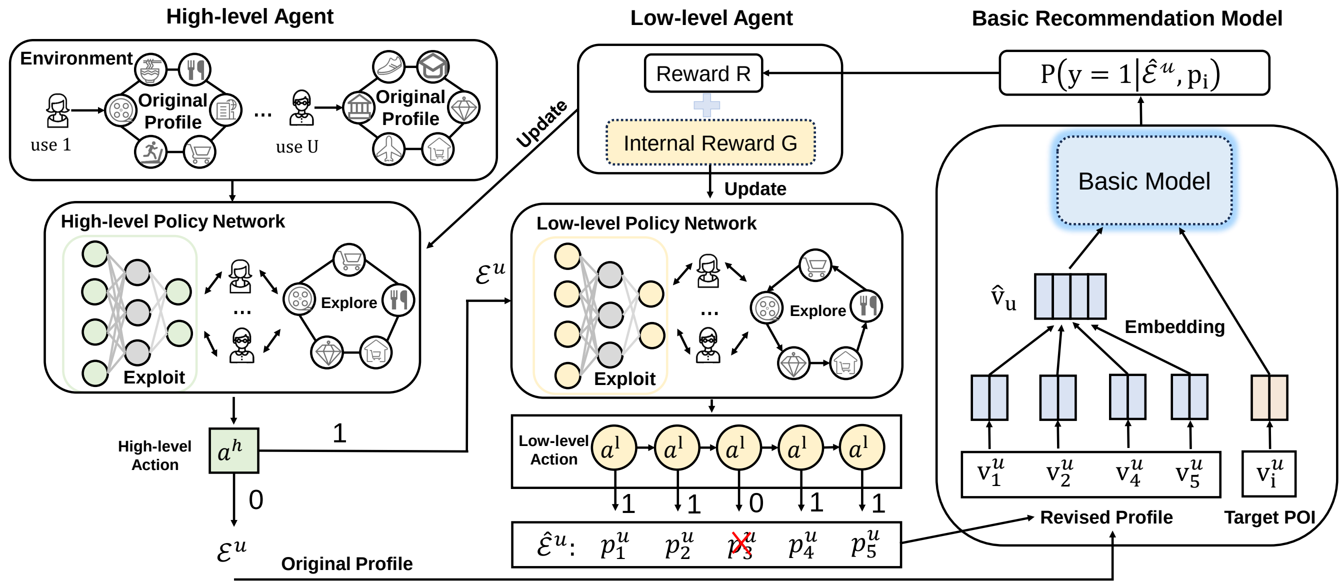


Figure 2: Hierarchical Reinforcement Learning for POI Recommendation. It is mainly composed of the High-level agent, the Low-level agent, and the Basic Recommendation Model and the basic recommendation model is exemplified by NAIS.

3.1 Overview Framework

Our model improves the fundamental recommendation system by refining user profiles. It identifies and removes "noisy" POIs—irrelevant check-ins that obscure the impact of significant POIs. This process relies on a hierarchical reinforcement learning algorithm, which divides the profile modification into sequential decision-making tasks at both high and low levels. The agent's actions, guided by a modification policy, aim to optimize the user profile summary. After each modification cycle, the agent receives feedback from the environment, comprising the dataset and an initial recommendation model, to adjust its policy. Subsequently, the basic recommendation model is retrained based on the agent's updated profile summaries. This joint training approach, which involves both the profile editor and the recommendation model, ensures more accurate POI recommendations. Figure 2 illustrates this framework.

3.2 The Basic Recommendation Model

The key element of recommendation is to accurately characterize the user's preferences based on hisher profile $\mathcal{E}^u$. The general idea is that we represent each historical POI p_t^u as a revalued low-dimensional embedding vector $\mathbf{v}_t^u$ and summarize the embeddings of all historical POIs $\mathbf{v}_1^u, \mathbf{v}_2^u, ..., \mathbf{v}_t^u$ to denote the preference of the user $u's$ $\hat{\mathbf{v}}_u$. If we also denote the target POI p_i as the embedding vector $\mathbf{v}_i$, the probability of recommending the POI p_i to user u, i.e., $\mathbf{P}(y = 1|\mathcal{E}^u, p_i)$, can be represented as

$$\mathbf{P}(y = 1|\mathcal{E}^u, p_i) = \sigma(\hat{\mathbf{v}}_u^T \mathbf{v}_i), \tag{4}$$

where $y = 1$ indicates that p_i is recommended to the user u and σ is the sigmoid function to transform the input into a probability. Then the key question is how to obtain the aggregated embedding $\hat{\mathbf{v}}_u$. One straightforward way is to average the embeddings of all the historical POIs,i.e. $\hat{\mathbf{v}}_u = \frac{1}{t_u} \sum_{t=1}^{t_u} \mathbf{p}_t^u$. However, equally treating all the POIs' contributions may impact the representation of a user's real interest in the target POI. Thus, as NAIS does, we can adopt the attention mechanism to estimate an attention coefficient a_{it}^u for each historical POI p_t^u when recommending p_i. Specifically, we parameterize the attention coefficient a_{it}^u as a function with $\mathbf{v}_t^u$ and $\mathbf{v}_i$ as input and then aggregate the embeddings according to their attentions defined as

$$\hat{\mathbf{v}}_u = \sum_{t=1}^{t_u} a_{it}^u \mathbf{p}_t^u, a_{it}^u = f(\mathbf{v}_t^u, \mathbf{v}_i), \tag{5}$$

where f can be instantiated by a multi-layer perception on the concatenation or the element-wise product of the two embeddings $\mathbf{v}_t^u$ and $\mathbf{v}_i$.

In addition to NAIS, our framework is compatible with various types of existing attention mechanisms and recommendation models. To ensure the generalizability and consistency of the study, we chose NAIS as the main mechanism for the computation of attention in our experiments. This choice not only demonstrates the flexibility of the framework, but also provides a unified benchmark for comparing different models.

3.3 Profile Reviser

As mentioned above, the purpose of the profile re-visner is to remove noisy processes that do not contribute much to the prediction. Inspired by the theory of hierarchical abstract machines [Parr and Russell, 1997], we describe the task of profile revision as a hierarchical Markov Decision Process (MDP). In general, we decompose the entire MDP task M into two classes of subtasks M^h and M^l, where M^h is a high-level abstract task in the hierarchy, and solving it solves the entire MDP M, and M^l is a low-level prototask in the hierarchy. Each kind of task is defined as a 4-tuple MDP ($\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}$), where $\mathcal{S}$ is a set of states, $\mathcal{A}$ is a set of actions, T is a transition model mapping $\mathcal{S} \times \mathcal{A} \times \mathcal{S}$ into probabilities in [0,1], and $\mathcal{R}$ is a reward function mapping $\mathcal{S} \times \mathcal{A} \times \mathcal{S}$ into real-valued rewards.

We formulate our task by a high-level task and a low-level task. Specifically, given a sequence of historical POIs $\mathcal{E}^u := (p_1^u, p_2^u, ..., p_{t_u}^u)$ of user u and target POI p, the agent performs a high-level task of one binary action to determine whether to revise the profile $\mathcal{E}^u$ or not. If it decides to revise $\mathcal{E}^u$, the agent performs a low-level task of multiple actions to determine whether to remove each historical POI $p_t^u \in \mathcal{E}^u$ or not. After the low-level task is finished, the overall task is finished. If the high-level task decides to make no revision, the low-level task will not be executed and the overall task is directly finished.

We formulate the profile reviser as two-level MDPs because some of the user profiles are discriminative and can already be correctly predicted by the basic recommendation model. We can simply keep those profiles as the original ones and only revise the ones that result in false recommendations. Out of this consideration, we design a high-level task to decide whether to revise the profile of a user or not, and a low-level task to decide which POI in the profile should be removed. We will introduce the details of how to design the state, action, and reward for the two-level tasks.

State. The high-level task takes an action according to the state of the whole profile $\mathcal{E}^u$ and the low-level task takes a sequence of actions according to the state of each POI $p_t^u \in \mathcal{E}^u$. We define different state features for the two tasks.

- **Low-level task:** When determining to remove a historical POI $p_t^u \in \mathcal{E}^u$, we define the state features s_t^l. As the cosine similarity between the embedding vectors of the current historical POI p_t^u and the target POI p_i, the element-wise product between them, and also the average of the two previous features over all the reserved historical POIs, where the embedding vector of a POI p_i can be provided by a pre-trained basic recommendation model. We also treat the user's level of interest in the POI as an additional state feature that enhances the contribution of p_t^u to p_i in addition to the similarity-based features. For simplicity, we omit the superscript u in all the notations on the state features.
- **High-level task:** When determining to revise a whole profile $\mathcal{E}^u$, we define the state features s^h as the average cosine similarity between the embedding vectors of each historical POI in $\mathcal{E}^u$ and the target POI and the average element-wise product between them. We also define an additional state feature as the probability $\mathbf{P}(y = 1|\mathcal{E}^u, p_i)$ of recommending p_i to user u by a basic recommendation model. The probability reflects the credibility of the POI p_i recommended based on the profile $\mathcal{E}^u$. The lower the probability of recommendation, the more effort should be put into revising $\mathcal{E}^u$. Note we train the profile reviser only based on the positive instances, i.e., a user profile paired with a real target POI, as negative instances with random target POIs can hardly guide the agent to select the contributing POIs to the target POI. Thus $\mathbf{P}(y = 0|\mathcal{E}^u, p_i)$ for a negative instance is not calculated.

Action and Policy. We define the high-level action $a^h \in \{0, 1\}$ as a binary value to represent whether to revise the whole profile of a user or not and define a low-level action a $a^l \in \{0, 1\}$ as a binary value to represent whether to remove the historical POI p_t^u or not. We perform a low-level action a_t^l according to the policy function defined as

$$\begin{aligned} \mathbf{H}^l &= \mathrm{ReLU}(\mathbf{W}^l s^t + \mathbf{b}^l), \\ \pi(s_t^l, a_t^l) &= P(a_t^l | s_t^l, \Theta^l) \\ &= a_t^l \sigma(\mathbf{W}_2 \mathbf{H}^l) + (1 - a_t^l)(1 - \sigma(\mathbf{W}_2 \mathbf{H}^l)), \end{aligned} \tag{6}$$

where $\mathbf{W}_1^l \in \mathbb{R}^{d_1^l \times d_2^l}$, $\mathbf{W}_2^l \in \mathbb{R}^{d_2^l \times 1}$ and $\mathbf{b}^l \in \mathbb{R}^{d_2^l}$ are the parameters to be learned with d_1^l as the number of the state features and d_2^l as the dimension of the hidden layer. Notation $\mathbf{H}_t^l$ represents the embedding of the input state. We denote $\Theta^l = \{\mathbf{W}_1^l, \mathbf{W}_2^l, \mathbf{b}^l\}$. The sigmoid function σ is used to transform the input into a probability. The high-level action is performed according to the similar policy function with different parameters $\Theta^h = \{\mathbf{W}_1^h, \mathbf{W}_2^h, \mathbf{b}^h\}$.

Reward. The reward is a signal to indicate whether the performed actions are reasonable or not. We assume that every low-level action in the low-level task has a delayed reward after the last action $a_{t_u}^l$ is performed for the last POI $p_{t_u}^l \in \mathcal{E}^u$. In other words, the immediate reward for a low-level action is zero except for the last low-level action. Thus, we define the reward for each low-level action define as

$$\mathcal{R}(a_t^l, s_t^l) = \begin{cases} \log p(\hat{\mathcal{E}}^u, p_i) - \log p(\mathcal{E}^u, p_i), & \text{if } t = t_u; \\ 0, & \text{otherwise}, \end{cases} \tag{7}$$

where $p(\mathcal{E}^u, p_i)$ is an abbreviation of $\mathbf{P}(y = 1|\mathcal{E}^u, p_i)$ and $\hat{\mathcal{E}}^u$ is the revised profile, which is a subset of $\mathcal{E}^u$. For the special case $\hat{\mathcal{E}}^u = \phi$, i.e., all the historical POIs are removed, we randomly select a POI from the original set $\mathcal{E}^u$. The reward is defined as the difference between the log-likelihood after and before the profile is revised. A positive difference indicates a positive utility gained by the revised profile.

If the high-level task chooses the revising action, it calls the low-level task and receives the same delayed reward $\mathcal{R}(a_t^l, s_t^l)$ after the last low-level action is performed. Otherwise, it keeps the original profile and obtains a zero reward as $\log p(\hat{\mathcal{E}}^u, p_i)$ is not changed.

In addition, we define an internal reward $\mathcal{G}(a_t^l, s_t^l)$ which is used only inside the low-level task to speed up its local learning and does not propagate to the high-level task. Specifically, we first calculate the average cosine similarity between each historical POI and the target POI after and before the profile is revised, and then use the difference between them as the internal reward $\mathcal{G}(a_t^l, s_t^l)$. The internal reward encourages the agent to select the most relevant POIs to the target POI. Finally, we sum $\mathcal{G}(a_t^l, s_t^l)$ and $\mathcal{R}(a_t^l, s_t^l)$ as the reward for the low-level task.

Objective Function. We aim at finding the optimal parameters of the policy function defined in Eq. 6 to maximize the

	Metrics / Method	HR@5			NDCG@5			Recall@5			F1@5			MAP@5		
		Baseline	HRL-PRP	Imp.	Baseline	HRL-PRP	Imp.	Baseline	HRL-PRP	Imp.	Baseline	HRL-PRP	Imp.	Baseline	HRL-PRP	Imp.
NYC	FPMC	0.1731	0.2152	+4.21%	0.0588	0.0854	+2.66%	0.0823	0.1341	+5.18%	0.0274	0.0649	+3.75%	0.0511	0.0828	+3.17%
	LSTM	0.3645	0.4048	+4.03%	0.2400	0.2698	+2.98%	0.3141	0.3670	+5.29%	0.1047	0.1551	+5.04%	0.2155	0.2700	+5.45%
	ST-RNN	0.1377	0.1744	+3.67%	0.0431	0.0820	+3.89%	0.0724	0.1235	+5.11%	0.0241	0.0699	+4.58%	0.0334	0.0707	+3.73%
	HST-LSTM	0.3205	0.3758	+5.53%	0.2164	0.2658	+4.94%	0.2853	0.3341	+4.88%	0.0951	0.1213	+2.62%	0.1953	0.2356	+4.03%
	SERM	0.3315	0.3618	+3.03%	0.2024	0.2360	+3.36%	0.3147	0.3452	+3.05%	0.1049	0.1475	+4.26%	0.2024	0.2548	+5.24%
	DeepMove	0.3458	0.3764	+3.06%	0.2498	0.3058	+5.60%	0.3350	0.3693	+3.43%	0.1117	0.1663	+5.46%	0.2215	0.2694	+4.79%
	LSTPM	0.3693	0.4008	+3.15%	0.2713	0.3081	+3.68%	0.3624	0.3956	+3.32%	0.1208	0.1551	+3.43%	0.2411	0.2817	+4.06%
	STAN	0.2313	0.2871	+5.58%	0.1354	0.1911	+5.57%	0.1835	0.2332	+4.97%	0.0612	0.0950	+3.38%	0.1195	0.1684	+4.89%
	CARA	0.3948	0.4440	+4.92%	0.2854	0.3206	+3.52%	0.3688	0.3997	+3.09%	0.1229	0.1679	+4.50%	0.2577	0.3070	+4.93%
	ATST-LSTM	0.4866	0.5180	+3.14%	0.3886	0.4286	+4.00%	0.4953	0.5286	+3.33%	0.1651	0.1958	+3.07%	0.3530	0.3976	+4.46%
	GeoSAN	0.4791	0.5148	+3.57%	0.3710	0.4143	+4.33%	0.5157	0.5584	+4.27%	0.1719	0.2203	+4.84%	0.3232	0.3731	+4.99%
	Metrics / Method	HR@5			NDCG@5			Recall@5			F1@5			MAP@5		
		Baseline	HRL-PRP	Imp.	Baseline	HRL-PRP	Imp.	Baseline	HRL-PRP	Imp.	Baseline	HRL-PRP	Imp.	Baseline	HRL-PRP	Imp.
TKY	FPMC	0.2721	0.3232	+5.11%	0.1574	0.1985	+4.11%	0.2156	0.2782	+6.26%	0.0719	0.1181	+4.62%	0.1381	0.1694	+3.13%
	LSTM	0.2841	0.3446	+6.05%	0.1819	0.2349	+5.30%	0.2449	0.2835	+3.86%	0.0816	0.1247	+4.31%	0.1610	0.2219	+6.09%
	ST-RNN	0.1285	0.1942	+6.57%	0.0230	0.0678	+4.48%	0.0331	0.0971	+6.40%	0.0110	0.0454	+3.44%	0.0196	0.0669	+4.73%
	HST-LSTM	0.2738	0.3335	+5.97%	0.1766	0.2284	+5.18%	0.2391	0.3051	+6.60%	0.0797	0.1054	+2.57%	0.1559	0.1791	+2.32%
	SERM	0.3128	0.3799	+6.71%	0.2040	0.2435	+3.95%	0.2740	0.3277	+5.37%	0.0913	0.1306	+3.93%	0.1807	0.2141	+3.34%
	DeepMove	0.3425	0.4069	+6.44%	0.2421	0.3078	+6.57%	0.3264	0.3647	+3.83%	0.1088	0.1437	+3.49%	0.2141	0.2536	+3.95%
	LSTPM	0.3360	0.3911	+5.51%	0.2401	0.2772	+3.71%	0.3307	0.3667	+3.60%	0.1102	0.1604	+5.02%	0.2101	0.2414	+3.13%
	STAN	0.2504	0.3174	+6.70%	0.1358	0.1674	+3.16%	0.1966	0.2635	+6.69%	0.0655	0.0915	+2.60%	0.1158	0.1475	+3.17%
	CARA	0.2130	0.2358	+2.28%	0.1051	0.1554	+5.03%	0.1657	0.2150	+4.93%	0.0552	0.1174	+6.22%	0.0852	0.1309	+4.57%
	ATST-LSTM	0.4027	0.4453	+4.26%	0.3029	0.3296	+2.67%	0.4427	0.4854	+4.27%	0.1476	0.1941	+4.65%	0.2569	0.3186	+6.17%
	GeoSAN	0.3403	0.3973	+5.70%	0.2293	0.2980	+6.87%	0.3218	0.3698	+4.80%	0.1073	0.1648	+5.75%	0.1987	0.2407	+4.20%

Table 1: The NAIS was uniformly used as a reward driver in the overall performance, and the results were averaged over five training sessions.

expected reward defined as

$$\Theta^* = \underset{\Theta}{\operatorname{argmax}} \sum_{\tau} \mathbf{P}_\Theta(\tau;\Theta)\mathcal{R}(\tau), \tag{8}$$

where Θ represents either Θ^h or Θ^l, τ is a sequence of the sampled actions and the transited states, $P_\Theta(\tau;\Theta)$ denotes the corresponding sampling probability and $\mathcal{R}(\tau)$ is the reward for the sampled sequence τ. The sampled sequence τ can be $\{s_1^l, a_1^l, s_2^l, ...s_t^l, a_t^l, s_{t+1}^l\}$ for the low-level task and $\{s^h, a^h\}$ fpr the high-level task.

Since there are too many possible action-state trajectories for the entire sequences of the two tasks, we adopt the policy gradient theorem and the Monte Carlo policy gradient methods [Sutton and Barto, 2018; Thomas and Brunskill, 2017] to sample M action-state trajectories. The index m denotes the m-th trajectory from these samples, where M is the total number of sampled trajectories. Based on these samples, we calculate the gradient of the parameters for the low-level policy function defined as

$$\begin{aligned} \nabla_\theta = \frac{1}{m}\sum_{m=1}^{M}\sum_{t=1}^{t_u} \nabla_\theta \log \pi_\theta(s_t^m, a_t^m) \\ \left(\mathcal{R}(a_t^m, s_t^m) + G(a_t^m, s_t^m)\right), \end{aligned} \tag{9}$$

where the reward $\mathcal{R}(a_t^m, s_t^m) + G(a_t^m, s_t^m)$ for each action-state pair in sequence $\tau^{(m)}$ is assigned the same value and equals to the terminal reward $\mathcal{R}(a_{t_u}^m, s_{t_u}^m) + G(a_{t_u}^m, s_{t_u}^m)$. The gradient for the high-level policy function is defined as

$$\nabla_\theta J = \frac{1}{m}\sum_{m=1}^{M} \nabla_\theta \log \pi_\theta(s^m, a^m)\mathcal{R}(a^m, s^m), \tag{10}$$

where the reward $\mathcal{R}(a^m, s^m)$ is assigned as $\mathcal{R}(a_{t_u}^m, s_{t_u}^m)$ when $a^m = 1$, or 0. We omit the superscript ^{h} and ^{l} in Eq. 8 and Eq. 9 for simplicity.

3.4 Model Training

The two models of the profile reviser and the basic recommendation model are interleaved together, and we need to train them jointly. The training process is shown in Algorithm 1, where we first pre-train the basic recommendation model based on the original dataset, then we fix the parameters of the basic recommendation model and pre-train the profile reviser to automatically revise the user profiles; finally, we jointly train the models together. Same as the settings, to have a stable update, each parameter is updated by a linear combination of its old version and the new old version, i.e. defined as

$$\Theta_{new} = \lambda\Theta_{new} + (1-\lambda)\Theta_{old}, \tag{11}$$

where $\lambda \ll 1$. The time complexity is $O(L(Nt_uM))$, where L is the number of epochs, N is the number of instances, T_U is the average number of historical courses and M is the Monto Carlo sampling time.

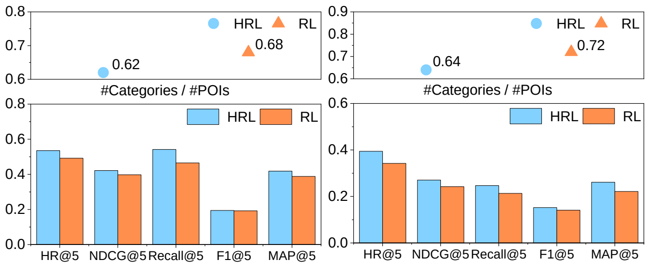
4 Experiment

We evaluate the performance of HRL-PRP in the next location prediction task. We aim to answer the following five main research questions:

- **Q1:** How does HRL-PRP perform in the next-location prediction task?
- **Q2:** What is the effectiveness of the roles of high-level and low-level agents in the decision-making process?
- **Q3:** How does reward design affect the performance of HRL-PRP?
- **Q4:** How can the necessity and rationality of the framework be analyzed through examples?
- **Q5:** How hyperparameters settings affect recommendation performance?

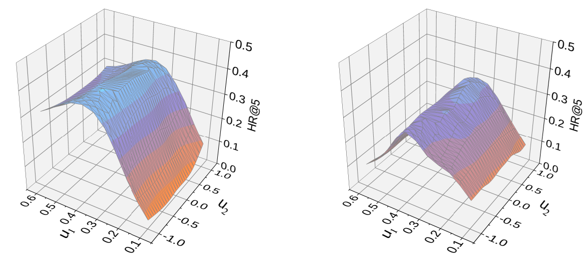
Methods	Revised profile or the learned attentions	The target POI
HRL-PRP	Cafes, Cake Stores, ~~Clothing Store~~, ~~Airports~~, ~~Train Stations~~	Baker's Store
NAIS	Cafes (13.22), Cake Stores (9.41), Clothing Store (9.55), Airports (5.42), Train Stations (7.14)	Baker's Store
HRL-PRP	Student Center, ~~Gym/Fitness Center~~, ~~Bar~~, Library	Student Apartment
NAIS	Student Center (15.20), Gym/Fitness Center (15.78), Bar (7.22), Library (12.41)	Student Apartment

Table 2: Case studies of the profiles revised by HRL-PRP and the attention coefficients learned by NAIS.



(a) NYC (b) TKY

Figure 3: An ablation study of high-level agent.



(a) NYC (b) TKY

Figure 4: An ablation study of low-level agent.

4.1 Experiment Settings

Datasets. In our study, we additionally included other datasets from Tokyo, Brightkite, Instagram, and Gowalla, which are widely used benchmarks in POI recommendation studies. Each dataset contains a series of historical POIs and a specific target POI. During the training phase, the last POI of the sequence is set as the target, and the rest constitutes the historical context. When generating negative samples, the target POI is replaced by four random POIs. In the testing phase, each check-in in the test set is considered as a target event and 99 random negative instances are paired to fully evaluate the model performance.

Baseline Methods. We compare HRL-PRP with eight base algorithms for comparison, including (1) **FPMC** [Rendle *et al.*, 2010], (2) **LSTM** [Memory, 2010]., (3) **ST-RNN** [Xu *et al.*, 2022], (4) **HST-LSTM** [Kong and Wu, 2018], (5) **SERM** [Yao *et al.*, 2017], (6) **Deepmove** [Feng *et al.*, 2018], (7) **LSTPM** [Sun *et al.*, 2020], (8) **STAN** [Luo *et al.*, 2021], (9) **CARA** [Manotumruksa *et al.*, 2020], (10) **ATST-LSTM** [Huang *et al.*, 2019], (11) **GeoSan** [Lian *et al.*, 2020].

Evaluation Metrics. Our evaluation uses the following metrics: hit rate (HR), normalized discounted cumulative gain (NDCG), recall, F1 score, and mean accuracy (MAP).

Implementaion Details. For the profile reviser, sampling time $\mathcal{M}$ is set as 3, and the learning rate is set as 0.001/0.0005 at the pre-training and joint-training stages respectively. In the policy function, the dimensions of the hidden layer d_2^l and d_2^h are both set as 8. For the basic recommender, the dimension of the POI embeddings is set to 128, the learning rate is 0.001 at both the pre-training and joint-training stages, and the size of the minibatch is 128. The delayed coefficient λ for the joint training is 0.0005.

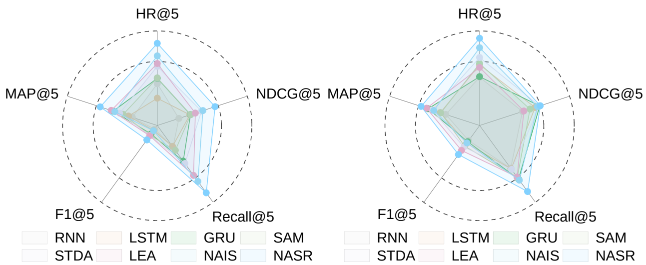
4.2 Overall Performance (Q1)

Table 1 shows the performance metrics of the HRL-PRP framework before and after combining it with the baseline recommendation model. The framework outperforms the baseline methods on key metrics, especially on sparse datasets, compared to FPMC methods. Serialization methods such as LSTM and LSTPM underperform in the distinction of historical POIs. While methods such as STAN and CARA, which fuse multidimensional information, are improved, they are affected by data noise and have different effects. Overall, existing recommender systems have limitations in handling diverse user interests. HRL-PRP more accurately reflects user preferences by effectively removing noisy POIs, thus achieving significant improvement in recommendation accuracy.

4.3 The Study of HRL-PRP Agents (Q2)

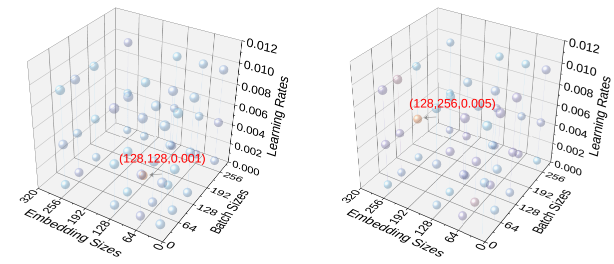
The design of High-level Agent. To demonstrate the effectiveness of high-level tasks, we compare the HRL architecture with the single-tier RL architecture (Deep Q-Learning, which directly decides whether to delete each POI mainly through low-level tasks) on several evaluation metrics. The results show that HRL outperforms single-tier RL on all metrics, highlighting the importance of high-level agents in maintaining and adjusting the diversity of user profiles. For example, the #Categories/#POIs values of HRL-modified profiles averaged 0.62 and 0.64, which were lower than those of single-tier RL at 0.68 and 0.72, suggesting that HRL produced more consistent profiles. This demonstrates the effectiveness of HRL-PRP's strategy of efficiently deciding to retain or modify profiles through high-level tasks.

The design of Low-level Agent. We compare the proposed HRL with the greedy correction algorithm. First, if $logP(y = 1|\mathcal{E}^u, p_i < \mu_1)$ decides to modify the whole contour $\mathcal{E}^u$, and if the cosine similarity between $e_t^u \in \mathcal{E}^u$ and p_i is less than μ_2, then $e_t^u \in \mathcal{E}^u$ is further deleted. $e_t^u \in \mathcal{E}^u$. In Figure 4, we tune μ_1 from 0.1 to 0.6 with an interval of 0.5, and tune μ_2 from -0.1 to 0.1 with an interval of 0.1, and get the best results for the two data when $\mu_1 = 0.4$ and $\mu_2 = 0.5$, which are 1.43% and 1.22% less than HRL-PRP. Note that the best performance is obtained when the number of remaining POIs is almost the same as HRL-PRP.



(a) NYC (b) TKY

Figure 5: An ablation study of reward design. Different attentional mechanism models include NAIS, NASR, Location Encoded Attention (LEA.), Scaled Dot Product Attention (SDTA.), Self-Attention Mechanism (SAM.), GRU, LSTM, and RNN.



(a) NYC (b) TKY

Figure 6: An Ablation study of hyperparameters embedding sizes, batch sizes, and learning rates. Sphere size corresponds to HR@5 indicator size. Blue to orange color indicates a gradual improvement in recommendation performance.

4.4 The Study of Reward (Q3)

We investigated the performance of different attention mechanisms driven by intrinsic rewards. As shown in Figure 5, among the multiple attention models examined, NAIS exhibits the best performance, while RNN and LSTM perform weakly. This performance difference mainly stems from the fact that RNN and LSTM do not sufficiently consider the episodic and non-sequential nature of user behavior in POI recommendation. This point highlights how the reward mechanism affects attention allocation in different models and how this allocation significantly affects the overall recommendation effectiveness.

4.5 The Study of Case Performance (Q4)

As shown in Table 2, the 2 cases of profiles corrected by the proposed HRL-PRP. The cases show that HRL-PRP is effective in removing spurious interest points that are not related to the target interest points. In contrast, while NAIS assigns high attention to contributing historical points of interest, some irrelevant points of interest do not receive significantly different or even higher attention than relevant points of interest, resulting in a weakened effect of truly contributing points of interest when aggregating all historical points of interest. As a result, the performance of recommendation models based on such differentiated revised profiles is improved.

4.6 The Study of Hyperparameters (Q5)

In reinforcement learning, experimental performance is extremely sensitive to parameter selection. As shown in Figure 6, we conducted an extensive learning rate analysis, scru-tinizing the effects of varying embedding dimensions andtraining batch sizes. Furthermore, the impact of differentlearning rates on joint training was also meticulously exam-ined. This analytical approach allowed us to precisely iden-tify optimal parameter configurations, thereby enhancing therobustness and reliability of our model's performance.

5 Related Work

POI Recommendation. Point of Interest (POI) recommender systems aim to recommend geographic locations, such as restaurants, museums, etc., based on a user's historical behavior and preferences. Current research focuses on the utilization of time-series data, employing a range of methods including content-based filtering [Xu *et al.*, 2017; Hu *et al.*, 2023; Wang *et al.*, 2019b; Wang *et al.*, 2019a], collaborative filtering [Yin *et al.*, 2021; Wang *et al.*, 2018; Liu *et al.*, 2018], and location-based recommendation algorithms [Chen *et al.*, 2021; Kuanr and Mohanty, 2020]. These approaches incorporate user behavioral pattern analysis, geolocation data, and in some cases social network information [Liu, 2022] to enhance the accuracy and level of personalization of recommendation results. While these approaches perform well, existing POI recommendation techniques are unable to extract key signals from a user's complex historical behavior, which limits their potential and accuracy for personalized recommendations.

Reinforcement Learning. Reinforcement learning centers on learning strategies through environmental interaction and feedback [Mnih *et al.*, 2015; Jiang *et al.*, 2023; Sanz-Cruzado *et al.*, 2019; Wang *et al.*, 2023a; Wang *et al.*, 2020]. The duality of exploration and exploitation in this learning model appropriately captures changing user preferences. Hierarchical Reinforcement Learning (HRL) extends it to a variety of comprehensive recommendation domains [Zhang *et al.*, 2019; Yu *et al.*, 2020; Xie *et al.*, 2021; Du *et al.*, 2022; Wang *et al.*, 2022; Zhang *et al.*, 2024] has a wide range of applications. Our research applies HRL to POI recommendation preprocessing by streamlining user profiles through task-independent partitioning.

6 Conclusion

This study proposes a hierarchical reinforcement learning preprocessing framework (HRL-PRP). We categorize the tasks into a two-level decision-making process: the high-level task is responsible for determining whether or not the current user's profile needs to be modified; the low-level task focuses on deciding which POIs to modify specifically. By jointly training the user profile modifier and the underlying recommendation model, we aim to improve the overall recommendation accuracy. The model simplifies the complexity of the recommendation process by effectively filtering irrelevant information while focusing on key content without explicitly supervising the signal. In the future, we plan to apply this model to other recommendation domains, to explore its potential for processing key items in users' historical data.

Acknowledgments

This work is supported by NSFC(under Grant No. 61976050, 62376048), Jilin Province Science and Technology Department Project (under Grant No. YDZJ202201ZYTS415, 20240602005RC), Jilin Education Department Project No.JJKH20231319KJ. This work is surported by the Science and Technology Development Fund (FDCT), Macau SAR (file no. 0123/2023/RIA2, 001/2024/SKL), the Start-up Research Grant of University of Macau (File no. SRG2021-00017-IOTSC).

References

[Chen *et al.*, 2021] Yi-Chung Chen, Hsi-Ho Huang, Sheng-Min Chiu, and Chiang Lee. Joint promotion partner recommendation systems using data from location-based social networks. *ISPRS International Journal of Geo-Information*, 10(2):57, 2021.

[Doan *et al.*, 2019] Khoa D Doan, Guolei Yang, and Chandan K Reddy. An attentive spatio-temporal neural model for successive point of interest recommendation. In *PAKDD*, pages 346–358, 2019.

[Du *et al.*, 2022] Qihan Du, Li Yu, Huiyuan Li, Youfang Leng, and Ningrui Ou. Denoising-oriented deep hierarchical reinforcement learning for next-basket recommendation. In *ICASSP*, pages 4093–4097, 2022.

[Feng *et al.*, 2018] Jie Feng, Yong Li, Chao Zhang, Funing Sun, Fanchao Meng, Ang Guo, and Depeng Jin. Deepmove: Predicting human mobility with attentional recurrent networks. In *WWW*, pages 1459–1468, 2018.

[Feng *et al.*, 2020] Shanshan Feng, Lucas Vinh Tran, Gao Cong, Lisi Chen, Jing Li, and Fan Li. Hme: A hyperbolic metric embedding approach for next-poi recommendation. In *SIGIR*, pages 1429–1438, 2020.

[He *et al.*, 2018] Xiangnan He, Zhankui He, Jingkuan Song, Zhenguang Liu, Yu-Gang Jiang, and Tat-Seng Chua. Nais: Neural attentive item similarity model for recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 30(12):2354–2366, 2018.

[Hu *et al.*, 2023] Xuanming Hu, Wei Fan, Kun Yi, Pengfei Wang, Yuanbo Xu, Yanjie Fu, and Pengyang Wang. Boosting urban prediction via addressing spatial-temporal distribution shift. In *2023 IEEE International Conference on Data Mining (ICDM)*, pages 160–169. IEEE, 2023.

[Huang *et al.*, 2019] Liwei Huang, Yutao Ma, Shibo Wang, and Yanbo Liu. An attention-based spatiotemporal lstm network for next poi recommendation. *IEEE Transactions on Services Computing*, 14(6):1585–1597, 2019.

[Huang *et al.*, 2020] Zhenhua Huang, Xiaolong Lin, Hai Liu, Bo Zhang, Yunwen Chen, and Yong Tang. Deep representation learning for location-based recommendation. *IEEE Transactions on Computational Social Systems*, 7(3):648–658, 2020.

[Jiang *et al.*, 2023] Lu Jiang, Kunpeng Liu, Yibin Wang, Dongjie Wang, Pengyang Wang, Yanjie Fu, and Minghao Yin. Reinforced explainable knowledge concept recommendation in moocs. *ACM Transactions on Intelligent Systems and Technology*, pages 1–20, 2023.

[Kong and Wu, 2018] Dejiang Kong and Fei Wu. A hierarchical spatial-temporal long-short term memory network for location prediction. In *IJCAI*, volume 18, 2018.

[Kuanr and Mohanty, 2020] Madhusree Kuanr and Sachi Nandan Mohanty. Location-based personalised recommendation systems for the tourists in india. *International Journal of Business Intelligence and Data Mining*, 17(3):377–392, 2020.

[Li *et al.*, 2017] Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tao Lian, and Jun Ma. Neural attentive session-based recommendation. In *CIKM*, pages 1419–1428, 2017.

[Lian *et al.*, 2020] Defu Lian, Yongji Wu, Yong Ge, Xing Xie, and Enhong Chen. Geography-aware sequential location recommendation. In *KDD*, pages 2009–2019, 2020.

[Liu *et al.*, 2018] K Liu, P Wang, J Zhang, G Liu, Yanjie Fu, and Sajal K Das. Incorporating interaction coupling among multi-view spatiotemporal contexts for mobike destination prediction. 2018.

[Liu *et al.*, 2020] Tongcun Liu, Jianxin Liao, Zhigen Wu, Yulong Wang, and Jingyu Wang. Exploiting geographical-temporal awareness attention for next point-of-interest recommendation. *Neurocomputing*, 400:227–237, 2020.

[Liu, 2022] Xiaoqiang Liu. Poi recommendation model using multi-head attention in location-based social network big data. *International Journal of Information Technologies and Systems Approach (IJITSA)*, 16(2):1–16, 2022.

[Luo *et al.*, 2021] Yingtao Luo, Qiang Liu, and Zhaocheng Liu. Stan: Spatio-temporal attention network for next location recommendation. In *The Web Conference*, pages 2177–2185, 2021.

[Manotumruksa *et al.*, 2020] Jarana Manotumruksa, Craig Macdonald, and Iadh Ounis. A contextual recurrent collaborative filtering framework for modelling sequences of venue checkins. *Information Processing & Management*, 57(6):102092, 2020.

[Memory, 2010] Long Short-Term Memory. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 2010.

[Mnih *et al.*, 2015] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, pages 529–533, 2015.

[Parr and Russell, 1997] Ronald Parr and Stuart Russell. Reinforcement learning with hierarchies of machines. *Advances in Neural Information Processing Systems*, 10, 1997.

[Qin *et al.*, 2023] Yifang Qin, Yifan Wang, Fang Sun, Wei Ju, Xuyang Hou, Zhe Wang, Jia Cheng, Jun Lei, and Ming

Zhang. Disenpoi: Disentangling sequential and geographical influence for point-of-interest recommendation. In *WSDM*, pages 508–516, 2023.

[Rendle *et al.*, 2010] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. Factorizing personalized markov chains for next-basket recommendation. In *WWW*, pages 811–820, 2010.

[Sanz-Cruzado *et al.*, 2019] Javier Sanz-Cruzado, Pablo Castells, and Esther López. A simple multi-armed nearest-neighbor bandit for interactive recommendation. In *RecSys*, pages 358–362, 2019.

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

[Sutton and Barto, 2018] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[Thomas and Brunskill, 2017] Philip S Thomas and Emma Brunskill. Policy gradient methods for reinforcement learning with function approximation and action-dependent baselines. *arXiv preprint arXiv:1706.06643*, 2017.

[Wang *et al.*, 2018] Pengyang Wang, Yanjie Fu, Jiawei Zhang, Pengfei Wang, Yu Zheng, and Charu Aggarwal. You are how you drive: Peer and temporal-aware representation learning for driving behavior analysis. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2457–2466, 2018.

[Wang *et al.*, 2019a] Pengyang Wang, Yanjie Fu, Hui Xiong, and Xiaolin Li. Adversarial substructured representation learning for mobile user profiling. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 130–138, 2019.

[Wang *et al.*, 2019b] Pengyang Wang, Xiaolin Li, Yu Zheng, Charu Aggarwal, and Yanjie Fu. Spatiotemporal representation learning for driving behavior analysis: A joint perspective of peer and temporal dependencies. *IEEE Transactions on Knowledge and Data Engineering*, 33(2):728–741, 2019.

[Wang *et al.*, 2020] Pengyang Wang, Kunpeng Liu, Lu Jiang, Xiaolin Li, and Yanjie Fu. Incremental mobile user profiling: Reinforcement learning with spatial knowledge graph for modeling event streams. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 853–861, 2020.

[Wang *et al.*, 2022] Enshu Wang, Rong Ding, Zhaoxing Yang, Haiming Jin, Chenglin Miao, Lu Su, Fan Zhang, Chunming Qiao, and Xinbing Wang. Joint charging and relocation recommendation for e-taxi drivers via multi-agent mean field hierarchical reinforcement learning. *IEEE Transactions on Mobile Computing*, pages 1274–1290, 2022.

[Wang *et al.*, 2023a] Dongjie Wang, Pengyang Wang, Yanjie Fu, Kunpeng Liu, Hui Xiong, and Charles E Hughes. Reinforced imitative graph learning for mobile user profiling. *IEEE Transactions on Knowledge and Data Engineering*, 2023.

[Wang *et al.*, 2023b] Yu Wang, An Liu, and Fan Jiang. A category-aware network for next poi recommendation with long-and short-term preferences. *IEEE Transactions on Computational Social Systems*, 2023.

[Wu *et al.*, 2019] Yuxia Wu, Ke Li, Guoshuai Zhao, and Xueming Qian. Long-and short-term preference learning for next poi recommendation. In *CIKM*, pages 2301–2304, 2019.

[Xie *et al.*, 2021] Ruobing Xie, Shaoliang Zhang, Rui Wang, Feng Xia, and Leyu Lin. Hierarchical reinforcement learning for integrated recommendation. In *AAAI*, pages 4521–4528, 2021.

[Xu *et al.*, 2017] Yi-Ning Xu, Lei Xu, Ling Huang, and Chang-Dong Wang. Social and content based collaborative filtering for point-of-interest recommendations. In *ICONIP*, pages 46–56. Springer, 2017.

[Xu *et al.*, 2022] Hengpeng Xu, Wenjian Ding, Wei Shen, Jun Wang, and Zhenglu Yang. Deep convolutional recurrent model for region recommendation with spatial and temporal contexts. *Ad Hoc Networks*, 129:102545, 2022.

[Yao *et al.*, 2017] Di Yao, Chao Zhang, Jianhui Huang, and Jingping Bi. Serm: A recurrent model for next location prediction in semantic trajectories. In *CIKM*, pages 2411–2414, 2017.

[Yin *et al.*, 2021] Minghao Yin, Yanheng Liu, Xu Zhou, and Geng Sun. A tensor decomposition based collaborative filtering algorithm for time-aware poi recommendation in lbsn. *Multimedia Tools and Applications*, 80(30):36215–36235, 2021.

[Yu *et al.*, 2020] Jifan Yu, Chenyu Wang, Gan Luo, Lei Hou, Juanzi Li, Jie Tang, Minlie Huang, and Zhiyuan Liu. Expanrl: Hierarchical reinforcement learning for course concept expansion in moocs. In *AACL-IJCNLP*, pages 770–780, 2020.

[Zhang *et al.*, 2019] Jing Zhang, Bowen Hao, Bo Chen, Cuiping Li, Hong Chen, and Jimeng Sun. Hierarchical reinforcement learning for course recommendation in moocs. In *AAAI*, pages 435–442, 2019.

[Zhang *et al.*, 2024] Zhaofan Zhang, Yanan Xiao, Lu Jiang, Dingqi Yang, Minghao Yin, and Pengyang Wang. Spatial-temporal interplay in human mobility: A hierarchical reinforcement learning approach with hypergraph representation. pages 9396–9404. AAAI Press, 2024.