Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence (IJCAI-22)

Next Point-of-Interest Recommendation with Inferring Multi-step Future Preferences

Lu Zhang[1], **Zhu Sun**[2,3*], **Ziqing Wu**[1], **Jie Zhang**[1], **Yew Soon Ong**[3,1] and **Xinghua Qu**[4]
[1]Nanyang Technological University, Singapore
[2]A*STAR Institute of High Performance Computing, Singapore
[3]A*STAR Centre for Frontier AI Research, Singapore
[4]Bytedance AI Lab, Singapore
sunzhuntu@gmail.com

Abstract

Existing studies on next point-of-interest (POI) recommendation mainly attempt to learn user preference from the past and current sequential behaviors. They, however, completely ignore the impact of future behaviors on the decision-making, thus hindering the quality of user preference learning. Intuitively, users' next POI visits may also be affected by their multi-step future behaviors, as users may often have activity planning in mind. To fill this gap, we propose a novel **C**ontext-aware **F**uture **P**reference inference **Rec**ommender (CFPRec) to help infer user future preference in a self-ensembling manner. In particular, it delicately derives multi-step future preferences from the learned past preference thanks to the periodic property of users' daily check-ins, so as to implicitly mimic user's activity planning before her next visit. The inferred future preferences are then seamlessly integrated with the current preference for more expressive user preference learning. Extensive experiments on three datasets demonstrate the superiority of CFPRec against state-of-the-arts.

1 Introduction

In recent years, the large volume of check-in records provided by location-based services, e.g., Foursquare, greatly promotes the researches on next point-of-interest (POI) recommendation [Liao *et al.*, 2018; Zhao *et al.*, 2019; Zhao *et al.*, 2020; Sun *et al.*, 2021; Chen *et al.*, 2021]. These methods aim at modeling sequential check-in behaviors and predicting time-sensitive POI visits, which emphasize that learning user preference from past and current check-in behaviors is a fundamental problem in boosting the next POI prediction.

With the assumption that users' next movements are highly correlated with their latest check-in behaviors, many next POI recommenders focus on modeling users' current check-ins (i.e., short-term behaviors). For instance, the pioneer work FPMC-LR [Cheng *et al.*, 2013] models two successive check-ins with localized region constraint incorporated. Afterwards, the studies derived from metric embedding, e.g.,

[*]Corresponding Author.

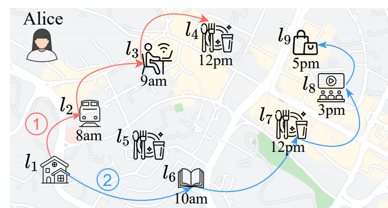

Figure 1: An example of Alice's two daily trajectories.

PRME [Feng *et al.*, 2015], and Word2Vec, e.g., HCT [Zhang *et al.*, 2019], aim to model the sequential check-ins and learn transition regularities by capturing the local co-occurrence of POIs. Later, recurrent neural networks (RNN) and its variants, such as ST-RNN [Liu *et al.*, 2016] and ARNN [Guo *et al.*, 2020], have gained much attention to capture users' current preferences by modeling the whole sequential information rather than the pairs of POIs within a trajectory as in the above studies. Meanwhile, a few methods fusing both past (i.e., long-term behaviors) and current sequential behaviors have achieved state-of-the-art performance in the next POI prediction [Wu *et al.*, 2020; Sun *et al.*, 2020].

Despite the success of the above models, we argue that they are not sufficient to learn accurate user preference, as they completely ignore the potential impact of future behaviors. Intuitively, a user's next movement may be affected by the multi-step future behaviors in addition to past and current behaviors, as they may have activity planning in mind. For illustration, Fig. 1 presents Alice's two daily trajectories. We attempt to make a recommendation for her at lunchtime. In trajectory 1, Alice went to a restaurant l_4 close to her office l_3, which implies that her next movement is highly influenced by her recent visits (e.g., office). While in trajectory 2, we observe that Alice checked in at a library l_6 and then she went to a distant restaurant l_7 rather than a nearby restaurant l_5. Afterwards, she went to a cinema at l_8 and shopping at l_9 after leaving l_7. That is to say, she most likely has an activity planning ahead of the check-in at a restaurant.

The above example emphasizes the potential impact and indicates promising improvements of future behaviors bringing in for next POI prediction. This is further verified by the most recent effort [Shao *et al.*, 2021], which leverages users' past and pre-specified future courses to achieve outstanding performance in multiple consecutive course recommendations. Nevertheless, in the next POI recommendation scenario, it is always impossible to acquire users' explicit multi-step future behaviors, for example, as in Alice's trajectory ②, such activities, going to a cinema at l_8 and shopping at l_9, are regarded as her future behaviors and always unavailable when recommending a restaurant for her at l_7. As a result, incorporating such future behaviors is non-trivial and still an open question in next POI recommendation.

To this end, we propose a novel **C**ontext-aware **F**uture **P**reference inference **Rec**ommender (CFPRec) to help infer and integrate user future preference in a self-ensembling manner, which consists of past and current preference encoders as well as future preference extractor. In particular, we employ a Transformer layer and an LSTM layer to model users' past and current preferences, respectively. Meanwhile, we devise three auxiliary objectives to supervise the past preference learning. The periodic property of users' daily check-in behaviors [Liu *et al.*, 2016; Feng *et al.*, 2018] motivates our design of the extractor for implicit future preference inference via a self-ensembling way, i.e., the multi-step future preferences are derived from the learned past preference with a two-layer attention aggregation, which mitigates the challenge of the absence of explicit future behaviors. As such, CFPRec encodes both left and right contexts (i.e., past/current and future preferences) for the next POI prediction.

Our main contributions lie in three-fold. **(1)** To the best of our knowledge, we are the first to investigate the effect of multi-step future preferences in next POI recommendation. **(2)** The proposed CFPRec jointly models users' past, current and future sequential behaviors to gain more expressive preference representation. Particularly, a future preference extractor is devised to infer the implicit future preference from the learned past preference in a self-ensembling manner. **(3)** The experimental results on three real-world datasets show that our CFPRec outperforms state-of-the-arts with average lifts of 16.4% and 21.6% on HR and NDCG, respectively.

2 Related Work

A line of research work on next POI recommendation assumes that users' next check-in behaviors are highly influenced by their recent visits. Hence, a class of Markov-based methods [He *et al.*, 2016; Zhao *et al.*, 2016] is introduced to learn the transition probability between successive check-ins. Due to the success of the RNN in modeling sequential data, many approaches are derived from RNN and its variants (e.g., LSTM [Hochreiter and Schmidhuber, 1997]), which are able to characterize users' current preferences (i.e., short-term preferences) by modeling the whole sequential behaviors. ST-RNN [Liu *et al.*, 2016] and SERM [Yao *et al.*, 2017] adopt RNN to model spatiotemporal regularities. ATST-LSTM [Huang *et al.*, 2019] and ARNN [Guo *et al.*, 2020] extend the LSTM with an attention mechanism to capture the relevant behaviors within trajectories. To overcome the long-term dependency problem of the RNN based methods, STAN [Luo *et al.*, 2021] employs the bi-attention model to learn correlations between non-consecutive check-in behaviors for next POI prediction.

Another line of studies exploits both past and current sequential behaviors, which has shown great benefits in next POI recommendation. For example, DeepMove [Feng *et al.*, 2018] applies a GRU unit to model the current sequential information and leverages the attention module to capture the most related context from the past trajectories. Similarly, PLSPL [Wu *et al.*, 2020] learns a user's location- and category-level preference from current trajectory via two parallel LSTM units and extracts the general preference from a past long-range sequence by using attention mechanism. STGN [Zhao *et al.*, 2019] equips the LSTM with time and distance gates to model users' long-term (i.e., past) and short-term (i.e., current) interests simultaneously. Although such RNN/LSTM based methods have shown the powerful ability in modeling sequential trajectories, they cannot capture the geographical correlations among non-consecutive POIs due to the nature of the standard RNN. As a result, LSTPM [Sun *et al.*, 2020] is proposed to address such a limitation by employing a geo-dilated RNN.

In fact, users' next movements might be influenced by their future POI visits in addition to past and current behaviors. However, the absence of explicit future behaviors brings challenges to learn future preferences, which hinders the effectiveness of next POI recommenders in capturing accurate user preference. A few efforts on sequential item recommendation tend to augment the recommendation quality by fusing future signals. Specifically, GRec [Yuan *et al.*, 2020] utilizes a gap-filling mechanism to model past and future contexts of a target item in the model training. A more recent study, FAT [Lu *et al.*, 2021], proposes to infer potential future preferences from other similar users. Considering the special characteristics of the next POI recommendation scenario, the periodic property of users' daily check-in behaviors motivates our design of the extractor for future preference inference from the learned past preference. Eventually, we achieve to model users' past, current and future preferences for more accurate next POI prediction in a self-ensembling fashion.

3 The Proposed Method

Fig. 2 illustrates the framework of CFPRec, which consists of three modules: (1) past preference encoder, which models users' past long-range trajectories to capture the context-aware correlations between non-consecutive check-in behaviors; (2) current preference encoder, which learns users' current preference by modeling the latest sequential check-ins; and (3) future preference extractor, which equips with a two-layer attention aggregation for inferring multi-step future preferences. As such, our proposed CFPRec is capable of incorporating past, current and future preferences for more accurate next POI recommendation. Before diving into each module, we first give the problem formulation and then introduce our proposed CFPRec.

Problem Formulation. Let $\mathcal{U} = \{u_1, u_2, \ldots, u_{|\mathcal{U}|}\}$ denote a set of users; $\mathcal{L} = \{l_1, l_2, \ldots, l_{|\mathcal{L}|}\}$ be a set of POIs (i.e., locations); $\mathcal{C} = \{c_1, c_2, \ldots, c_{|\mathcal{C}|}\}$ be a set of categories of POIs; and $\mathcal{T} = \{t_1, t_2, \ldots, t_{24}, t_{w_0}, t_{w_1}\}$ be a set of time slots (map one day into 24 slots) as well as weekday (t_{w_0}) and weekend (t_{w_1}). A check-in record $r = (u, l, c, g, t)$ indicates that

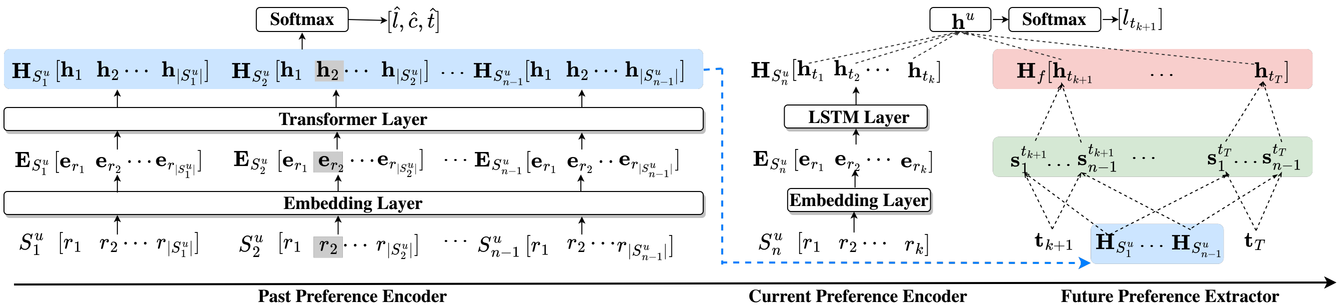

Figure 2: The overall architecture of our proposed CFPRec, which consists of past and current preference encoders as well as the future preference extractor; and the rectangles in grey, pink and green color represent the masked record, inter-sequence attention aggregation and intra-sequence attention aggregation, respectively.

user u visited POI l at time t, where l is characterized by category c and geocoded by g, i.e., (longitude, latitude). For each user $u \in \mathcal{U}$, let $S^u = \{S^u_1, S^u_2, \ldots, S^u_n\}$ denote her trajectories (i.e., check-in sequences), where the i-th trajectory $S^u_i = \{r_1, r_2, \ldots, r_{|S^u_i|}\}$ denotes a set of chronologically ordered check-in records within a day. User u's latest trajectory $S^u_n = \{r_1, r_2, \ldots, r_k\}$ is regarded as the current sequential behaviors S^u_{cur} (i.e., short-term behaviors), and the past behaviors S^u_{past} (i.e., long-term behaviors) are formed by the trajectories $\{S^u_1, \ldots, S^u_{n-1}\}$ that happened before S^u_{cur} in the past several days (e.g., 7 or 14 days).

Research Problem. Given a user u's past behaviors S^u_{past} and current behaviors S^u_{cur}, we aim to recommend a set of POIs for user u to visit at the next timestamp t_{k+1} by inferring her multi-step future preferences, i.e., at $t_{k+1}, \ldots, t_T$.

3.1 Past Preference Encoder

Generally, users often visit POIs for different purposes (i.e., various contextual environments) and such complex nature of users' check-in behaviors brings significant challenges to learn accurate user preference. Fortunately, the periodic property of users' behaviors provides essential signals for user preference learning. Inspired by the recent advance of the context-aware location embedding method [Lin *et al.*, 2021], we employ a bidirectional Transformer [Vaswani *et al.*, 2017] as the past preference encoder for two reasons. First, in the *model training* process, the Transformer encoder is able to model both left and right contexts for a target POI, which enables to learn more expressive representations for POIs. Second, it shows a powerful ability in capturing context-aware correlations between non-consecutive check-ins and aggregating the most relevant behaviors automatically within a trajectory [Lin *et al.*, 2021].

Accordingly, as shown in the left part of Fig. 2, we propose an extensional Transformer encoder to capture a user's past preference which encodes her spatiotemporal-aware dynamic preference over POI and static preference regarding activity. For example, a user may visit a different restaurant (i.e., POI) at lunchtime with the same activity (i.e., food). Given a user u's past trajectory $S^u_i = \{r_1, r_2, \ldots, r_{|S^u_i|}\}$ and $S^u_i \in S^u_{past}$, each check-in behavior $r_i = (u, l, c, g, t)$ is represented as:

$$\mathbf{e}_{r_i} = \mathbf{u} \oplus \mathbf{l} \oplus \mathbf{c} \oplus \mathbf{t}, \;\; \mathbf{e}_{r_i} \in \mathbb{R}^{5D} \tag{1}$$

where $\oplus$ denotes the concatenation operation; $\mathbf{u}, \mathbf{l}, \mathbf{c} \in \mathbb{R}^D$ are the embeddings of user, location, category, respectively; $\mathbf{t} \in \mathbb{R}^{2D}$ encodes the time slot and weekday/weekend, i.e., $\mathbf{t} = \mathbf{t}_k \oplus \mathbf{t}_{w_0} / \mathbf{t}_{w_1}$ [1]. Thus S^u_i is transformed by the embedding layer with $\mathbf{E}_{S^u_i} = [\mathbf{e}_{r_1}, \mathbf{e}_{r_2}, \ldots, \mathbf{e}_{r_{|S^u_i|}}]$, which is then fed into the Transformer layer:

$$\begin{aligned} \mathbf{H}_{S^u_i} &= [\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_{|S^u_i|}] = \mathrm{TransLayer}(\mathbf{E}_{S^u_i}) \\ &= \mathrm{TransLayer}(\mathbf{E}_{S^u_i}\mathbf{W}_Q, \mathbf{E}_{S^u_i}\mathbf{W}_K, \mathbf{E}_{S^u_i}\mathbf{W}_V, \Delta^{dist}), \end{aligned} \tag{2}$$

$$\mathrm{TansLayer}(\mathbf{Q}, \mathbf{K}, \mathbf{V}, \Delta^{dist}) = \left(F\left(\frac{\mathbf{Q}\mathbf{K}^\top + \Delta^{dist}}{\sqrt{5D}}\right) \right) \mathbf{V}, \tag{3}$$

where $F(\cdot)$ is the softmax function; $\mathrm{TransLayer}(\cdot)$ is a Transformer layer; $\mathbf{H}_{S^u_i}$ is the output matrix consisting of the hidden vectors and $\mathbf{h}_i \in \mathbb{R}^{5D}$; $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V \in \mathbb{R}^{5D \times 5D}$ are transformation weight matrices of query, key and value, respectively; $\Delta^{dist} \in \mathbb{R}^{|S^u_i| \times |S^u_i|}$ is the spatial relation matrix; and given the distance $dist(l_i, l_j)$ derived from g_{l_i} and g_{l_j}, $\Delta^{dist}_{i,j} = (1 + \mathrm{dist}(l_i, l_j))^{-1}$ measures the spatial influence between locations l_i and l_j.

It is worth noting that the past preference encoder serves as a significant role in capturing user past preference from long-term sequential behaviors. Inspired by the recent study in constructing self-supervised training objective to achieve the masked token prediction [Lin *et al.*, 2021], we devise three auxiliary tasks to further supervise the past preference learning and help to capture more accurate preference representation. Given a sequence S^u_i, we randomly mask 20% check-in records, e.g., the original $r_2 = (u, l, c, g, t)$ will be replaced by $r_m = (u, l_m, c_m, g_m, t_m)$, and the corresponding embedding vector and hidden state are represented by $\mathbf{e}_m$ and $\mathbf{h}_m$, respectively. As such, we conduct multi-task predictions for the original location l, category c and time t by decoding the hidden vector $\mathbf{h}_m$ via the softmax layer:

$$\hat{l} = F(\mathbf{h}_m \mathbf{W}_l), \;\; \hat{c} = F(\mathbf{h}_m \mathbf{W}_c), \;\; \hat{t} = F(\mathbf{h}_m \mathbf{W}_t), \tag{4}$$

where $\mathbf{W}_l \in \mathbb{R}^{5D \times |\mathcal{L}|}, \mathbf{W}_c \in \mathbb{R}^{5D \times |\mathcal{C}|}, \mathbf{W}_t \in \mathbb{R}^{5D \times |\mathcal{T}|}$ are the transformation matrices corresponding to such three tasks.

[1] All the following $\mathbf{t}$ integrates the embeddings of the time slot and weekday/weekend unless specified otherwise.

The loss (negative log-likelihood) of auxiliary tasks is,

$$\begin{aligned} L^{aux} &= L^{aux}_l + L^{aux}_c + L^{aux}_t \\ &= -\sum\nolimits_{k \in N_m} \log(\hat{l}_k) + \log(\hat{c}_k) + \log(\hat{t}_k), \end{aligned} \tag{5}$$

where N_m is the set of all masked records.

3.2 Current Preference Encoder

The recurrent module is widely adopted to model users' short-term sequential behaviors among the recent studies [Wu *et al.*, 2020; Sun *et al.*, 2020]. We thus employ an LSTM unit to characterize users' temporal-aware sequential dependency from the current check-in behaviors. As shown in the middle part of Fig. 2, given a user u's current trajectory S^u_n, each check-in record $r_i \in S^u_n$ is embedded by $\mathbf{e}_{r_i}$ as in Eq.(1). Accordingly, the embedded current trajectory $\mathbf{E}_{S^u_n} = [\mathbf{e}_{r_1}, \mathbf{e}_{r_2}, \ldots, \mathbf{e}_{r_k}]$ is fed into the LSTM layer:

$$\mathbf{h}_{t_i} = \text{LSTM}(\mathbf{e}_{r_i}, \mathbf{h}_{t_{i-1}}),\ i \in \{1, 2, \ldots, k\}, \tag{6}$$

where $\mathbf{h}_{t_i}$ and $\mathbf{h}_{t_{i-1}}$ are the hidden states corresponding to timestamps t_i and t_{i-1}, respectively. In this way, the LSTM layer finally outputs a sequence of hidden states for the user's current preference, that is, $\mathbf{H}_{S^u_n} = [\mathbf{h}_{t_1}, \mathbf{h}_{t_2}, \ldots, \mathbf{h}_{t_k}]$.

3.3 Future Preference Extractor

As depicted in Fig. 1, users may often have activity planning ahead of their next movements, thus fusing future behaviors would facilitate to capture more accurate user preferences w.r.t. their next POI visits. However, as in Alice's trajectory ②, the activities (going to a cinema at l_8 and shopping at l_9) are regarded as her future behaviors and always unavailable when recommending a restaurant for her at l_7. A few bi-directional based methods have encoded past and future behaviors in the model training, i.e., modeling the left and right contexts for the target POI [Lu *et al.*, 2021]. However, in the model prediction, they are unable to access users' explicit future behaviors, which consequently limits their capability on capturing user preference. Fortunately, the periodic property of users' daily check-in behaviors inspires us to derive an extractor, which is equipped with a two-layer attention aggregation for inferring multi-step implicit future preference in a self-ensembling manner, so as to implicitly mimic a user's activity planning before her next visit.

Intra-sequence Attention Aggregation. As a user usually shows a similar preference in the same temporal context (i.e. periodic property), we thus devise a temporal-aware attention to identify the most relevant behaviors within a past trajectory regarding a future temporal context. Specifically, a user's past trajectory $S^u_i \in S^u_{past}$ has been encoded into the hidden state matrix $\mathbf{H}_{S^u_i} = [\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_{|S^u_i|}]$ via the past preference encoder, wherein each hidden vector $\mathbf{h}_i$ characterizes a user's spatiotemporal-aware dynamic POI preference and static activity preference. Afterwards, a future time embedding $\mathbf{t}_f$ is used as a query vector attending to each hidden state of the past trajectory and thus the future-temporal trajectory embedding ($\mathbf{s}^{t_f}_i$) is represented as:

$$\begin{aligned} \mathbf{s}^{t_f}_i &= \sum\nolimits_{i=1}^{|S^u_i|} \alpha_i \mathbf{h}_i,\ t_f \in \{t_{k+1}, t_{k+2}, \ldots, t_T\} \\ \alpha_i &= \frac{\exp(\mathbf{t}_f^\top \mathbf{h}_i)}{\sum_{i'=1}^{|S^u_i|} \exp(\mathbf{t}_f^\top \mathbf{h}_{i'})}. \end{aligned} \tag{7}$$

Therefore, $\mathbf{s}^{t_f}_i$ is able to indicate potential future behaviors through extracting the most relevant behavior regarding a future timestamp t_f from the past trajectory S^u_i.

Inter-sequence Attention Aggregation. The intra-sequence attention layer emphasizes the effect of future temporal context on the relevant behaviors within a trajectory. Moreover, users' daily sequential preferences are also evolving, such as Alice's two daily trajectories in Fig. 1. We thus propose an inter-sequence attention layer to model the sequential preference evolving process across different trajectories. Consequently, after obtaining multiple future-temporal trajectory embeddings regarding t_f, e.g., $\{\mathbf{s}^{t_{k+1}}_1, \ldots, \mathbf{s}^{t_{k+1}}_{n-1}\}$, we use the dynamic user embedding $\mathbf{u}$ as the query vector to attend to relevant trajectory preference:

$$\begin{aligned} \mathbf{h}_{t_f} &= \sum\nolimits_{i=1}^{n-1} \beta_i \mathbf{s}^{t_f}_i,\ t_f \in \{t_{k+1}, t_{k+2}, \ldots, t_T\} \\ \beta_i &= \frac{\exp(\mathbf{u}^\top \mathbf{s}^{t_f}_i)}{\sum_{i'=1}^{n-1} \exp(\mathbf{u}^\top \mathbf{s}^{t_f}_{i'})}, \end{aligned} \tag{8}$$

where β_i is an attention score; $\mathbf{h}_{t_f}$ refers to the future preference at timestamp t_f. Till now, the future preference extractor achieves to infer multi-step future preferences (i.e., $\mathbf{H}_f = [\mathbf{h}_{t_{k+1}}, \mathbf{h}_{t_{k+2}}, \ldots, \mathbf{h}_{t_T}]$) via self-ensembling, which also inherits users' past preferences.

3.4 Model Training and Complexity Analysis

We now obtain the hidden states characterizing user preference $\mathbf{H}_{S^u_n,f} = [\mathbf{H}_{S^u_n}, \mathbf{H}_f]$ via the past and current preference encoders as well as the future preference extractor, which enables CFPRec to model both left and right contexts for the target POI during training and prediction without data leakage. With the assumption that user's next movement may be affected by her multi-step future preferences in addition to past and current preferences, the final user preference ($\mathbf{h}^u$) is thus formulated by aggregating each hidden state in $\mathbf{H}_{S^u_n,f}$,

$$\begin{aligned} \mathbf{h}^u &= \sum\nolimits_{i=1}^{T} \omega_i \mathbf{h}_{t_i},\ \mathbf{h}_{t_i} \in \mathbf{H}_{S^u_n,f}, \\ \omega_i &= \frac{\exp(\mathbf{u}^\top \mathbf{h}_{t_i})}{\sum_{i'=1}^{T} \exp(\mathbf{u}^\top \mathbf{h}_{t_{i'}})}, \end{aligned} \tag{9}$$

where ω_i is the attention score between the dynamic user embedding $\mathbf{u}$ and each hidden state $\mathbf{h}_{t_i}$. The learned user preference $\mathbf{h}^u$ is then used to decode the probability distribution over the $|\mathcal{L}|$ POIs with the softmax function:

$$\hat{\mathbf{y}} = F(\mathbf{h}^u \mathbf{W}_o), \tag{10}$$

where $\hat{\mathbf{y}} \in \mathbb{R}^{|\mathcal{L}|}$ is the predicted probability distribution regarding t_{k+1}; and $\mathbf{W}_o \in \mathbb{R}^{5D \times |\mathcal{L}|}$ is a transformation matrix. Hence, the objective function for the target POI prediction is:

$$L^{poi} = -\sum\nolimits_{i \in \mathcal{N}} \log(\hat{y}_i), \tag{11}$$

where $\mathcal{N}$ is the set of training samples; and $\hat{y}_i$ is the predicted probability of the ground truth POI corresponding to the i-th training sample. Finally, the overall loss is given by:

$$L = L^{poi} + \eta L^{aux}, \tag{12}$$

where $\eta \in [0, 1]$ is a hyper-parameter to balance the target POI prediction and the auxiliary objective.

Algorithm 1: Training of the CFPRec

Input: $\mathcal{U}, \{S^{u_1}, S^{u_2}, \ldots, S^{u_{|\mathcal{U}|}}\}, max\_iter, \eta, D, T, \gamma$
Output: Trained model $\{\Theta\}$
1 Initialize model parameters $\Theta, \mathcal{N} = \varnothing$;
// Training instance construction
2 **foreach** $u \in \mathcal{U}$ **do**
3 **foreach** $S_i^u \in S^u$ **do**
4 $S_{past}^u = \{S_1^u, S_2^u, \ldots, S_{n-1}^u\}, S_{cur}^u = S_n^u$
5 $\{S_{past}^u, S_{cur}^u\} \mapsto \mathcal{N}$
// Update parameters Θ
6 **foreach** $iter \in \{1, 2, \ldots, max\_iter\}$ **do**
7 Select instance $\{S_{past}^u, S_{cur}^u\}$ from $\mathcal{N}$
8 Update Θ by minimizing the objective in Eq.(12)

Complexity Analysis. Algorithm 1 outlines the training process of the CFPRec, where we first construct training instances (lines 2-5) and then update parameters (lines 6-8). During the model training, the computational complexities of the past preference encoder, current preference encoder and future preference extractor are respectively $\mathcal{O}((n-1) \cdot |\bar{S}|^2 \cdot D)$, $\mathcal{O}(|\bar{S}| \cdot D^2)$ and $\mathcal{O}(D)$, where n is the number of sequences in a training instance; $|\bar{S}|$ is the average sequence length; D is the embedding size. Since $n, |\bar{S}| < D$, the overall complexity of CFPRec for each training instance is around $\mathcal{O}(|\bar{S}| \cdot D^2)$. Normally, $|\bar{S}| < 10$ and $D \leq 100$, implying CFPRec is scalable to large-scale datasets.

4 Experiments

We conduct experiments to investigate the following research questions[2]. (RQ1) Does the proposed CFPRec outperform state-of-the-art baselines? (RQ2) How do different components of CFPRec affect its performance? (RQ3) How do key hyper-parameters of CFPRec affect its performance?

4.1 Experimental Setup

Datasets and Metrics. We adopt three datasets collected from Foursquare [Yang *et al.*, 2016] in three cities, i.e., Singapore (SIN), New York City (NYC) and Phoenix (PHO), as shown in Table 1. Following [Sun *et al.*, 2020], we remove POIs with less than 10 interactions and filter out inactive users with less than 5 trajectories and trajectories with less than 3 check-in records. We then split the trajectories of each user in the ratio of 8:1:1 based on timestamps, where the earliest 80% is regarded as training set; the second last 10% as validation set and the last 10% as test set. Finally, we employ two widely-used metrics, i.e., HitRate@K (HR@K) and NDCG@K, to evaluate the performance of all methods. Generally, higher metric values indicate better ranking performance. For a more robust comparison, we run each method 10 times, and report the averaged results as the final results.

Baselines. We compare CFPRec with eight baselines. (1) MostPop recommends POIs via popularity; (2) BPRMF is a Bayesian Personalized Ranking [Rendle *et al.*, 2009] based matrix factorization method; (3) ST-RNN [Liu *et al.*, 2016] is a pioneer RNN based model that incorporates spatiotermpoal contexts between successive check-ins for next POI recommendation; (4) ATST-LSTM [Huang *et al.*, 2019] is an attention-based spatiotemporal LSTM model which focuses on the relevant historical check-ins in a trajectory; (5) MCARNN [Liao *et al.*, 2018] is a multi-task context aware RNN based model which leverages the spatial-activity topic for the next activity and location prediction; (6) PLSPL [Wu *et al.*, 2020] is a state-of-the-art next POI recommender which equips with two modules to capture user's past and current preference, respectively; (7) iMTL [Zhang *et al.*, 2021] is a state-of-the-art multi-task based next POI recommender which models sequential correlations of activities and locations via a two-channel encoder; (8) CTLE [Lin *et al.*, 2021] is a state-of-the-art contextual location embedding method which is built upon a bi-directional Transformer framework.

[2]Our code is available at https://github.com/wuziqi2/CFPRec

	# User	# POI	# Check-in	Density
SIN	8,648	33,712	355,337	0.12%
NYC	16,387	56,252	511,431	0.06%
PHO	2,946	7,247	47,980	0.22%

Table 1: Statistics of the three real-world datasets.

Parameter Settings. We empirically find out the best parameter settings for all methods. In particular, the embedding size D is searched in $[20, 100]$ stepped by 20; the learning rate γ and regularization coefficient are searched in $\{0.0001, 0.001, 0.01, 0.1\}$. The rest parameters of each method are searched as suggested by the original papers. For CFPRec, we implement it with Pytorch, and Adam is adopted as the optimizer; $D = 60/40/60$ for SIN, NYC and PHO; the number of iterations are correspondingly set as 25/25/15 for SIN, NYC and PHO; $\gamma = 0.0001$; $\eta = 1$; the number of Transformer blocks is 1; the number of LSTM layers is $3/2/2$ for SIN, NYC and PHO.

4.2 Results and Analysis

Comparison Results (RQ1). The performance of all methods are presented in Table 2. For the three datasets, MostPop performs the worst, which suggests the necessity of personalization in next POI recommendation. BPRMF is defeated by RNN based models as it fails to capture the sequential regularity from users' check-in behaviors. The comparisons for all the RNN based methods are discussed as follows. (1) ATST-LSTM performs better than ST-RNN which indicates that leveraging the attention mechanism in modeling sequential behaviors can enhance the quality of next POI recommendation. (2) The relative higher performance of MCARNN, iMTL, and PLSPL shows that jointly modeling users' spatiotemporal-aware activity and location preference is beneficial to capture users' preferences for their next POI visits. (3) PLSPL shows promising results which illustrates the significance of modeling past and current sequential behaviors for more accurate next POI recommendations, while ignoring to model the future preference hinders its user preference learning compared with our CFPRec. (4) Our CFPRec achieves better performance than the state-of-the-art Transformer based CTLE, which also helps verify the benefits of

	SIN				NYC				PHO			
	HR@5	HR@10	NDCG@5	NDCG@10	HR@5	HR@10	NDCG@5	NDCG@10	HR@5	HR@10	NDCG@5	NDCG@10
MostPop	0.0125	0.0293	0.0106	0.0151	0.0139	0.0235	0.0101	0.0145	0.0160	0.0223	0.0114	0.0131
BPRMF	0.0352	0.0525	0.0222	0.0380	0.0381	0.0468	0.0239	0.0339	0.0487	0.0585	0.0256	0.0304
ST-RNN	0.0959	0.1370	0.0655	0.0794	0.1347	0.1826	0.0593	0.1303	0.1240	0.2028	0.0802	0.1229
ATST-LSTM	0.1296	0.1933	0.1027	0.1476	0.1667	0.2031	0.0912	0.1638	0.1579	0.2377	0.1033	0.1385
MCARNN	0.1608	0.1862	0.1169	0.1591	0.1835	0.2397	0.1036	0.1870	0.1905	0.2726	0.1264	0.1617
iMTL	0.1505	0.1801	0.1051	0.1423	0.1798	0.2422	0.0989	0.1861	0.1830	0.2747	0.1301	0.1632
PLSPL	0.1447	0.1719	0.1126	0.1384	0.1741	0.2413	0.0961	0.1825	0.1775	0.2569	0.1285	0.1538
CTLE	0.2041	0.2784	0.1315	0.1556	0.2421	0.3205	0.1513	0.1841	0.2632	0.3605	0.1995	0.2068
CFPRec	**0.2310**	**0.3085**	**0.1588**	**0.1836**	**0.2771**	0.3606	**0.1971**	**0.2190**	**0.3421**	**0.4253**	**0.2432**	**0.2730**
Improve	13.2%	10.8%	11.2%	15.4%	14.5%	12.5%	30.3%	19.0%	30.0%	17.5%	21.9%	32.0%

Table 2: Performance of all methods on three datasets, where the best performance is boldfaced; the runner up is underlined; the row 'Improve' indicates the improvements achieved by CFPRec relative to the runner up, whose significance is determined by a paired t-test with $p < 0.01$.

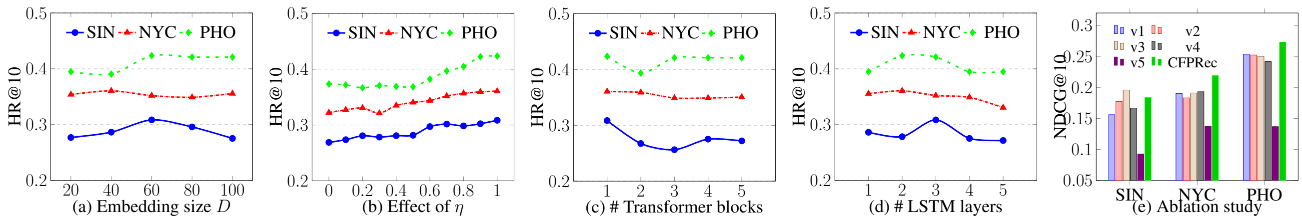

Figure 3: Parameter sensitivity analysis and ablation study of CFPRec on the three datasets.

inferring users' future preferences in the next POI recommendation. Overall, CFPRec significantly outperforms all baselines with average lifts of 16.4% and 21.6% on HR and NDCG, respectively, where the significance is determined by a paired t-test with $p < 0.01$.

Ablation Study (RQ2). We explore the efficacy of different components on CFPRec by comparing with its five variants. (**v1**) it removes the auxiliary objective in the past preference encoder; (**v2**) it omits the future preference extractor; we follow [Wu *et al.*, 2020] to characterize user preference by modeling both past and current sequential behaviors in a concatenation way; (**v3**) it only adopts one-step future preference at t_{k+1}; (**v4**) it replaces the two-layer attention aggregation in the future preference extractor with the average pooling; and (**v5**) it removes both past preference encoder and future preference extractor, i.e., we only adopt LSTM to model users' sequential behaviors for recommendation. Note that we do not consider the variant omitting the past preference encoder, as it is mainly used to infer the implicit future preference with the periodic property. Our CFPRec can model multi-step future preferences. However, in this paper, we consider two-step future preferences due to the limited trajectory length.

We only report the results on NDCG@10 as shown in Fig. 3(e) due to space limitation, and similar trends can be observed on HR@10. From the results, we note that CFPRec generally achieves better performance than its variants, validating the design efficacy of different components. First, CFPRec consistently outperforms v1, implying the benefits of auxiliary objective in past user preference learning. Second, it surpasses both v2 and v5 (the worst variant), which suggests that modeling both users' past and future preference indeed boosts the performance of next POI recommendation. Meanwhile, CFPRec defeats v3 in most cases exhibits the value of multi-step future preferences. Lastly, v4 generally underperforms CFPRec showing the advantage of the two-layer attention aggregation in inferring users' future preferences.

Parameter Sensitivity Analysis (RQ3). We now examine the impacts of key parameters on CFPRec, including embedding size D, the importance of auxiliary tasks η, the number of Transformer blocks, and the number of LSTM layers. We only report the results on HR@10 as depicted in Fig. 3(a-d) due to space limitation, and similar trends can be found on NDCG@10. We note that (1) as the embedding size D increases, the performance of CFPRec first climbs up, then reaches its peak, and finally keeps stable or slightly drops on all datasets; (2) the performance of CFPRec gradually improves with the increasing of η on all datasets, and the best setting is 1; (3) the best performance of CFPRec is achieved with only one block on all datasets, indicating more blocks may not guarantee a better result; and (4) similar performance variation regarding the number of LSTM layers on all datasets is observed as for the embedding size D.

5 Conclusion

In this paper, we propose a novel method, CFPRec, to jointly model users' past, current and future preferences for next POI recommendation. Specifically, we design the past preference encoder to model users' past preferences via a Transformer layer, which further employs three auxiliary objectives to help supervise the user preference learning from past trajectories. The current preference encoder then aims to model users' latest sequential behaviors by an LSTM layer. Lastly, a future preference extractor is devised by equipping with a two-layer attention aggregation to infer multi-step future preferences. Experimental results on three real-world datasets demonstrate the superiority of CFPRec against state-of-the-art baselines.

Acknowledgments

This research is supported by the MOE AcRF Tier 1 funding (RG90/20) awarded to Dr. Jie Zhang, and is also partially supported by the Centre for Frontier AI Research and the Institute of High Performance Computing, A*STAR.

References

[Chen *et al.*, 2021] Yudong Chen, Xin Wang, Miao Fan, Jizhou Huang, Shengwen Yang, and Wenwu Zhu. Curriculum meta-learning for next poi recommendation. In *KDD*, 2021.

[Cheng *et al.*, 2013] Chen Cheng, Haiqin Yang, Michael R Lyu, and Irwin King. Where you like to go next: Successive point-of-interest recommendation. In *IJCAI*, 2013.

[Feng *et al.*, 2015] Shanshan Feng, Xutao Li, Yifeng Zeng, Gao Cong, Yeow Meng Chee, and Quan Yuan. Personalized ranking metric embedding for next new poi recommendation. In *IJCAI*, 2015.

[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*, 2018.

[Guo *et al.*, 2020] Qing Guo, Zhu Sun, Jie Zhang, and Yin-Leng Theng. An attentional recurrent neural network for personalized next location recommendation. In *AAAI*, 2020.

[He *et al.*, 2016] Jing He, Xin Li, Lejian Liao, Dandan Song, and William Cheung. Inferring a personalized next point-of-interest recommendation model with latent behavior patterns. In *AAAI*, 2016.

[Hochreiter and Schmidhuber, 1997] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 1997.

[Huang *et al.*, 2019] Liwei Huang, Yutao Ma, Shibo Wang, and Yanbo Liu. An attention-based spatiotemporal lstm network for next poi recommendation. *TSC*, 2019.

[Liao *et al.*, 2018] Dongliang Liao, Weiqing Liu, Yuan Zhong, Jing Li, and Guowei Wang. Predicting activity and location with multi-task context aware recurrent neural network. In *IJCAI*, 2018.

[Lin *et al.*, 2021] Yan Lin, Huaiyu Wan, Shengnan Guo, and Youfang Lin. Pre-training context and time aware location embeddings from spatial-temporal trajectories for user next location prediction. In *AAAI*, 2021.

[Liu *et al.*, 2016] Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. Predicting the next location: A recurrent model with spatial and temporal contexts. In *AAAI*, 2016.

[Lu *et al.*, 2021] Yujie Lu, Shengyu Zhang, Yingxuan Huang, Luyao Wang, Xinyao Yu, Zhou Zhao, and Fei Wu. Future-aware diverse trends framework for recommendation. In *WWW*, 2021.

[Luo *et al.*, 2021] Yingtao Luo, Qiang Liu, and Zhaocheng Liu. Stan: Spatio-temporal attention network for next location recommendation. In *WWW*, 2021.

[Rendle *et al.*, 2009] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *UAI*, 2009.

[Shao *et al.*, 2021] Erzhuo Shao, Shiyuan Guo, and Zachary A Pardos. Degree planning with plan-bert: Multi-semester recommendation using future courses of interest. In *AAAI*, 2021.

[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*, 2020.

[Sun *et al.*, 2021] Zhu Sun, Chen Li, Yu Lei, Lu Zhang, Jie Zhang, and Shunpan Liang. Point-of-interest recommendation for users-businesses with uncertain check-ins. *TKDE*, 2021.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

[Wu *et al.*, 2020] Yuxia Wu, Ke Li, Guoshuai Zhao, and QIAN Xueming. Personalized long-and short-term preference learning for next poi recommendation. *TKDE*, 2020.

[Yang *et al.*, 2016] Dingqi Yang, Daqing Zhang, and Bingqing Qu. Participatory cultural mapping based on collective behavior data in location-based social networks. *TIST*, 2016.

[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*, 2017.

[Yuan *et al.*, 2020] Fajie Yuan, Xiangnan He, Haochuan Jiang, Guibing Guo, Jian Xiong, Zhezhao Xu, and Yilin Xiong. Future data helps training: Modeling future contexts for session-based recommendation. In *WWW*, 2020.

[Zhang *et al.*, 2019] Lu Zhang, Zhu Sun, Jie Zhang, Horst Kloeden, and Felix Klanner. Modeling hierarchical category transition for next poi recommendation with uncertain check-ins. *Information Sciences*, 2019.

[Zhang *et al.*, 2021] Lu Zhang, Zhu Sun, Jie Zhang, Yu Lei, Chen Li, Ziqing Wu, Horst Kloeden, and Felix Klanner. An interactive multi-task learning framework for next poi recommendation with uncertain check-ins. In *IJCAI*, 2021.

[Zhao *et al.*, 2016] Shenglin Zhao, Tong Zhao, Haiqin Yang, Michael R Lyu, and Irwin King. Stellar: spatial-temporal latent ranking for successive point-of-interest recommendation. In *AAAI*, 2016.

[Zhao *et al.*, 2019] Pengpeng Zhao, Haifeng Zhu, Yanchi Liu, Jiajie Xu, Zhixu Li, Fuzhen Zhuang, Victor S Sheng, and Xiaofang Zhou. Where to go next: A spatio-temporal gated network for next poi recommendation. In *AAAI*, 2019.

[Zhao *et al.*, 2020] Kangzhi Zhao, Yong Zhang, Hongzhi Yin, Jin Wang, Kai Zheng, Xiaofang Zhou, and Chunxiao Xing. Discovering subsequence patterns for next poi recommendation. In *IJCAI*, 2020.