Conversation-Based Adaptive Relational Translation Method for Next POI Recommendation With Uncertain Check-Ins

Lu Zhang, Zhu Sun, Jie Zhang, Yiwen Wu, and Yunwen Xia

***Abstract*— The uncertain check-ins bring challenges for current static next point-of-interest (POI) recommendation methods. Fortunately, the conversation-based recommendation has been shown the merit of integrating immediate user preference for more accurate recommendations. We, therefore, propose a conversation-based adaptive relational translation (CART) approach for the next POI recommendation over uncertain check-ins. It is equipped with recommender and conversation modules to interactively acquire users' immediate preferences and make dynamic recommendations. Specifically, the recommender built upon the adaptive relational translation method performs location prediction via modeling both users' historical sequential behaviors and the immediate preference received from conversations; the conversation module aims to achieve successful recommendations in fewer conversation turns by learning a conversational strategy, whereby the recommender can be updated via the user response. Extensive experiments on four real-world datasets show the superiority of our proposed CART over the state of the arts.**

***Index Terms*— Conversational recommender systems, next POI recommendation, policy network, translation-based model, uncertain check-ins.**

I. Introduction

THE rapid development of the next point-of-interest (POI) recommendation benefits from a large number of check-ins delivered by users, which, in turn, helps a user explore his surroundings. Accordingly, most existing studies assume that such check-ins reflect users' actual visits [1], [2], i.e., *certain check-ins*. In reality, users may provide *uncertain check-ins* at a *collective POI*, e.g., a shopping mall that contains multiple individual POIs, since users do not always disclose some specific check-ins due to their privacy concerns, or the bias of the indoor navigation of GPS, thus resulting in the transition vanishing issue [3].

Manuscript received December 2, 2020; revised May 22, 2021, September 10, 2021, and November 16, 2021; accepted January 14, 2022. This work was supported in part by the MOE AcRF Tier 1 Funding (RG90/20) awarded to Dr. J. Zhang, the Center for Frontier AI Research and the Institute of High Performance Computing, A*STAR. *(Corresponding author: Zhu Sun.)*

Lu Zhang, Jie Zhang, Yiwen Wu, and Yunwen Xia are with the School of Computer Science and Engineering, Nanyang Technological University, Singapore (e-mail: zhang_lu@ntu.edu.sg; zhangj@ntu.edu.sg; ywu029@ntu.edu.sg; yunwen.xia@ntu.edu.sg).

Zhu Sun is with the Institute of High Performance Computing (IHPC) and the Centre for Frontier AI Research, A*STAR, Artificial Intelligence, Analytics and Informatics (AI3) Horizontal Technology Program Office (HTPO), Singapore (e-mail: sunzhuntu@gmail.com).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TNNLS.2022.3146443.

Digital Object Identifier 10.1109/TNNLS.2022.3146443

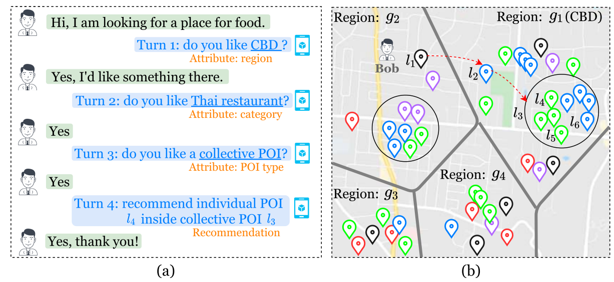


Fig. 1. Illustration of conversation-based next POI recommendation with uncertain check-ins. (a) Example of conversational recommendation process. (b) Examples of certain and uncertain check-ins, where the location markers are individual POIs (e.g., l_1, l_2); the POIs marked with the same color correspond to the same category, i.e., Travel (black), Shop & Service (blue), Food (green), Entertainment (purple), Nightlife spot (red); and the circle contains multiple individual POIs is collective POI (e.g., l_3).

As an example in Fig. 1(b), Bob left successive certain check-ins at individual POIs l_1 (e.g., a bus stop) and l_2 (e.g., gym) and then visited an individual POI l_4 (e.g., Thai restaurant) inside the *collective POI* l_3 (e.g., a shopping mall). We are more likely to obtain his movements $l_1 \rightarrow l_2 \rightarrow l_3$ with uncertain check-ins at l_3 involved, instead of the certain check-ins $l_1 \rightarrow l_2 \rightarrow l_4$ used in most existing studies. In this case, the state of the arts hardly locates a user's next POI visit l_4 based on such observed records $l_1 \rightarrow l_2 \rightarrow l_3$. Although the pioneer efforts [3], [4] attempt to model the interplay between activity (i.e., category) and location for alleviating the issue of incomplete sequential dependencies caused by uncertain check-ins, we argue that such solutions solely model users' historical check-ins via static recommendation methods. It is, thus, hard to recommend accurate POIs when an undesired activity is predicted.

Due to the development of conversational recommender system (CRS) techniques [5]–[9], which brings unprecedented potential in resolving the limitation of static recommenders, they allow the recommender to acquire a user's immediate preference and achieve high-quality recommendations via real-time conversations. This motivates us to incorporate CRS in the next POI recommendation. Recent studies [10], [11] have demonstrated the efficacy of multiround[1] setting of conversations for more accurate recommendations. However, their

[1]A multiround CRS refers to that the system interacts with a user and makes recommendation multiple times until the user accepts the recommendation or chooses to quit.

factorization-based recommender modules [12] adopt the inner product to measure the similarity of user–item interactions, which does not satisfy the condition of the triangle inequality and, thus, limits their generalization ability [13]. Contrarily, the translation-based recommenders [14], [15] perform metric learning (i.e., satisfy the condition of the triangle inequality), showing scalability and promising performance over traditional factorization-based methods. As next POI recommendation is highly context-dependent (e.g., spatiotemporal and high-order sequential regularities) [16]–[18] and faces severe data sparsity issue of user–location interactions, we believe that the location-aware CRS should integrate the merit of multiround setting of conversation and the powerful generalization ability of the recommender module.

In this study, we investigate the location-aware CRS under an uncertain check-in scenario. It is a more realistic research problem since we cannot always obtain users' certain check-ins, which, in turn, benefits the protection of user privacy to some extend [3]. This, consequently, leads to more challenges in modeling user preference due to the transition vanishing issue and extremely sparsity issue compared with conventional next POI recommendation. Fortunately, CRS enables interactions with users to acquire their immediate preference, e.g., the desired activities and POI types, which is one of the key challenges for a successful recommendation under an uncertain check-in scenario, while the static recommendation method under this scenario is hard to suggest accurate POIs when an undesired activity and a POI type are predicted [3], [4].

Toward realizing more accurate next POI recommendations with uncertain check-ins, we propose a simple yet effective conversation-based adaptive relational translation framework (CART) in a multiround paradigm,[2] which consists of recommender and conversation modules. Inspired by the recent success of translation-based methods in location recommendation [15], we propose an adaptive relational translation-based recommender to model the rich context of users' historical check-ins and immediate preference in conversations. Meanwhile, the recommender adapts to user feedback and achieves online updates by treating the rejected POIs obtained from conversations as negative samples. In this way, it can naturally alleviate the issue of the absence of the true negative samples in the static recommendation models [4], [17]. The conversation module seeks the best conversational strategy (i.e., what attributes to ask to quickly clarify the user preference and when to recommend POIs), aiming to achieve successful recommendations with fewer conversation turns. Moreover, it specially designs the auxiliary reward for successful collective POI recommendation and adopts the rating-based sampling strategy for individual POI selection inside such a collective POI. As such, the proposed CART naturally resolves the issue of uncertain check-ins.

Running Example: Fig. 1(a) depicts a running example of a conversation-based next POI recommendation over uncertain check-ins. Bob is seeking a place for food to start the conversation after visiting l_2, and the system then asks the attribute of region g_1. If Bob confirms the asked region, the system continues to identify the attributes of a fine-grained category (e.g. Thai restaurant) and POI type (e.g. collective POI). Given sufficient confirmed attributes after several turns and his historical sequential check-ins, the recommender is confident to recommend a collective POI l_3 at region g_1. Meanwhile, the system further selects individual POIs (e.g., l_4) inside the confirmed l_3 via our devised rating-based sampling strategy. To this end, our CART is able to locate his preferred individual POIs via accumulating his immediate preference from a coarse- to fine-grained fashion, i.e., category-aware granularity: *food* $\rightarrow$ *Thai restaurant* and spatial-aware granularity: *region* $\rightarrow$ *collective POI* $\rightarrow$ *individual POI*.

In summary, our main contributions lie in threefold.

1) We investigate the task of conversation-based next POI recommendation with uncertain check-ins, highlighting the efficacy of multiround conversations between a user and the system in predicting a more accurate next POI. To our best knowledge, this is the first work to exploit CRS to address the issue of uncertain check-ins in the location recommendation community, which facilitates the research of user mobility prediction.
2) We devise the CART framework by taking advantage of the adaptive relational translation-based method and CRS. That is, CART consists of the tailored recommender module and the conversation module, seeking to model a user's rich context of historical check-ins and immediate preference in conversations to achieve a successful next POI recommendation with fewer conversation turns.
3) We conduct extensive experiments to evaluate the performance of our proposed CART framework on four real-world datasets. The results demonstrate the superiority of CART over both state-of-the-art CRSs and static recommendation methods for the next POI recommendation with uncertain check-ins.

II. Related Work

This section reviews the related works from two aspects, including: 1) the most recent studies on next POI recommendation and investigations on uncertain check-ins and 2) conversational recommender systems.

A. Next POI Recommendation

Next POI recommendation has been widely studied via modeling rich context information of a user's historical check-in behaviors [19]–[22], such as spatiotemporal context [23], [24], sequential regularity [1], [18], or their joint effects [2], [25]. This mainly attempts to learn better representations of users and POIs [26]. However, these efforts posit that users' accurate check-ins at individual POIs (i.e. certain check-ins) are always available and accessible, ignoring the presence of uncertain check-ins, which brings the issue of incomplete context information of check-ins and, thus, limits the capability of the recommender system in decision-making [27]. To ease such an issue, recent studies

[2]The objective of multiround paradigm CRS is to make a successful recommendation with fewer turns of conversations, where the template (e.g., Do you like __?) is used for wrapping attributes (e.g., Thai restaurant) to simulate the conversation (e.g., Do you like Thai Restaurant?).

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

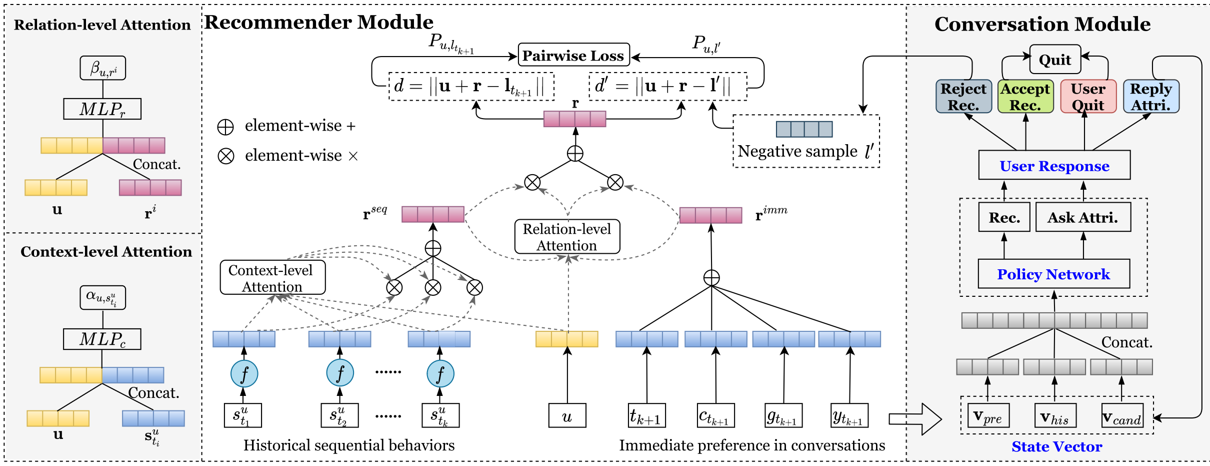


Fig. 2. Framework of CART that is composed of recommender and conversation modules.

turn to model the context-aware sequential regularity and characterize the underlying activity over uncertain check-ins [3], [4]. However, these static recommendation approaches face such an intrinsic limitation: they cannot well capture a user's immediate preference solely based on his historical behaviors since they are not able to interactively clarify a user's current preference in the static model setting. It is, therefore, hard to locate the desired POI when the system fails to predict accurate user preference (e.g., category) for the approaches in [3], [4]. As collecting feedback on whether the recommended POI satisfies the user demand could help enhance the recommender systems [28], this directly motivates us to exploit a CRS in the next POI recommendation over the scenario of uncertain check-ins due to its promising ability in capturing immediate user preference through an interactive conversion.

B. Conversational Recommender Systems

The CRS has been widely explored under different assumptions and application scenarios [29], which provides a new possibility for capturing immediate user preference via dynamic interaction with users [5], [6], [8], [9], [30]–[40]. For example, a line of research [9], [30], [41] adopts the bandit methods (e.g., the Thompson sampling [42]) to solve the exploit-explore issue in a CRS for cold start users. Other works that incorporate natural language processing module [33], [34], [43]–[45] aim to understand the user preference from their utterances and generate fluent responses for natural and effective dialogs. As the CRS needs to interact with users multiple times for asking the user preference on attributes or making recommendations, a good CRS should learn a conversational strategy for when to ask or make recommendations.

There is another line of research that is closely related to our study. In particular, the conversational recommender model (CRM) [5] trains a deep policy network to decide whether to ask an attribute or make recommendations at each turn under the single-round setting (i.e., the CRS terminates once making a recommendation regardless of the result is satisfactory or not). By contrast, estimation action reflection (EAR) [10] and simple conversational path reasoning (SCPR) [11] follow the more realistic multiround setting and aim to strategically ask an attribute or make recommendations with fewer turns. In this work, we formalize our research problem under the line of multiround setting, wherein the CRS focuses on simulating such a scenario instead of real dialog: the system interacts with a user via asking attributes (e.g., Thai restaurant) wrapped by the template (e.g., Do you like __?) multiple times to clarify his preference and make recommendations until the user accepts the recommendation results or chooses to quit [10].

We argue that an efficient location-aware CRS should delicately accommodate both the rich context of users' check-in behaviors and the merit of multiround CRS, so as to ease the issue of uncertain check-ins in the next POI recommendation. As such, our proposed CART advances the state-of-the-art CRSs in two aspects: 1) the recommender module models a user's historical check-ins and immediate preference obtained in conversations, which helps safely locate the accurate preference for a user's next movement and 2) the conversation module aims to learn the conversational strategy and achieve successful recommendations with fewer turns. Most importantly, it specially designs the auxiliary reward for successful collective POI recommendations and adopts the rating-based sampling strategy for individual POI selection inside such a collective POI. In sum, this results in a new angle of CRS for the next POI recommendation over uncertain check-ins.

III. Proposed Methods

Fig. 2 depicts the architecture of our proposed CART under the multiround setting, aiming to simulate the conversation process, that is, inquiring a user's preference toward attributes or making recommendations until the user accepts the recommendation or chooses to quit. This enables users to interact with an agent via iterative conversations and then performs recommendation by two modules: 1) Recommender Module, which is an adaptive relational translation method to jointly model a user's historical sequential behaviors and the immediate preference obtained in

TABLE I
NOTATIONS

Notations	Descriptions
$\mathcal{U}, \mathcal{L}, \mathcal{C}$	user set, location set, category set
u, l, c, t	user $u \in \mathcal{U}$, location $l \in \mathcal{L}$, category $c \in \mathcal{C}$, time t
g, y	geographical region, POI type
$\mathcal{S}^u$	all check sequences of user u
$\mathcal{S}^{u,j}$	the j-th check sequence of user u: $\mathcal{S}^{u,j} \in \mathcal{S}^u$
$s^u_{t_i}$	check-in activity of u at t_i: $s^u_{t_i} = (l_{t_i}, t_i, c_{t_i}, g_{t_i}, y_{t_i})$
$\mathbf{u}, \mathbf{l}, \mathbf{t}, \mathbf{c}, \mathbf{g}, \mathbf{y}$	embeddings associated with u, l, t, c, g, y
$\mathcal{P}_u$	u's immediate preferences obtained in conversations
α, β	attention weights of relations
$\mathbf{v}$	state vector
$\mathbf{r}^{seq}, \mathbf{r}^{imm}$	relation embeddings
$d(u,l)$	distant between u and l
P_{u,l_t}	probability of u visiting l at time t
$\mathcal{A}$	action space
Re^+_*, Re^-_*	positive/negative reward
$\pi_\theta(a_n\|\mathbf{v}_n)$	policy network and $a_n \in \mathcal{A}$
γ	discount factor

conversations and 2) Conversation Module,[3] which seeks the best strategy for action selection and achieves a successful recommendation with fewer conversation turns. In particular, it consists of a *State Vector*, *Policy Network*, and *User Response*. At each conversation turn, the *State Vector* serves as a bridge between the recommender and conversation modules, encoding the user preference and conversation history. The *Policy Network* plays the role of action generator, which decides on whether to ask an attribute (e.g., a category) or make a recommendation (e.g., a specific POI) based on the input state vector and returns an up-to-date reward according to the user response.

Notations: Let $u \in \mathcal{U}$ denote a user u from the user set $\mathcal{U}$ and $l \in \mathcal{L}$ denote a POI (i.e., location)[4] from the POI set $\mathcal{L}$. For each user u, we order all his check-ins by timestamp and then split them into sequences by day, denoted as $\mathcal{S}^u$. Thus, the jth check-in sequence of u refers to a set of time-ordered check-ins within a day: $\mathcal{S}^{u,j} = \{s^u_{t_1}, s^u_{t_2}, \ldots, s^u_{t_n}\}$, $\mathcal{S}^{u,j} \in \mathcal{S}^u$; each check-in $s^u_{t_i} = (l_{t_i}, t_i, c_{t_i}, g_{t_i}, y_{t_i})$ means that a user u visits a POI l_{t_i} at geographical region g_{t_i} and time t_i (timestamps are discretized into 24 slots in a day); $c_{t_i} \in \mathcal{C}$ is the category of l_{t_i}; and y_{t_i} is the type of l_{t_i}, where $y_{t_i} = 0$ refers to an individual POI, otherwise a collective POI. We list some important notations in Table I.

Research Problem: Given user u's historical check-in sequence $\mathcal{S}^{u,j} = \{s^u_{t_1}, s^u_{t_2}, \ldots, s^u_{t_k}\}$ and the next check-in time t_{k+1}, our goal is to recommend the next POI $l_{t_{k+1}}$ in multiround conversations by maximizing the accumulated reward given the limited conversation turns. Note that, if the recommended $l_{t_{k+1}}$ is a collective POI, we further predict individual POIs inside $l_{t_{k+1}}$.

A. Recommender Module

With the aim of making more accurate next POI recommendations over uncertain check-ins, we design a translation-based recommender under the multiround setting of CRS, which is capable of modeling the rich context information (e.g., spatiotemporal context and immediate preference) of location visits. Hence, the proposed recommender not only delicately captures users' context-aware preference but also overcomes the inherent limitation of inner product in FM [15], which is adopted as the recommender in state-of-the-art CRS methods [5], [10].

1) Basic Translation Method: TransE is a representative method among the various knowledge graph based techniques [14], [46], which aims to embed the triples (h, r, t) into a transition space that satisfies $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$, where h, r, and t represent head entity, relation, and tail entity, respectively.[5] It has been widely studied in recommender systems due to its promising ability over factorization-based methods [14], [47]. Hence, the affinity of user u and POI l is defined as

$$d(u,l) = \|\mathbf{u} + \mathbf{r} - \mathbf{l}\|_2^2. \tag{1}$$

Note that there are various translation-based methods [48]–[50] that can model different relational patterns, e.g., symmetric/asymmetric, inversion, and composition. In our study, we mainly focus on taking advantage of translation-based methods to alleviate the inherent issue of factorization-based methods, instead of exploring the efficiency of different translation-based methods. We, thus, propose a simple yet effective TransE-based recommender. The key point is, therefore, to capture the translation vector $\mathbf{r}$ (i.e., the relation embedding encodes a user's spatiotemporal sequential check-in behaviors and immediate preference in conversations) in the following model description.

2) Adaptive Relational Translation Method: In the next POI recommendation scenario, the sequential regularity of a user's location visit is of significance in capturing his personalized preference [2], [17], [25]. Enlightened by the recent success of context-aware relation representation in [15], we construct an adaptive relation vector to translate a user u to the next POI l by considering both historical sequential behaviors $\mathcal{S}^{u,j} = \{s^u_{t_1}, s^u_{t_2}, \ldots, s^u_{t_k}\}$ and immediate preference $\mathcal{P}_u$ obtained in conversations, i.e., $(u, < \mathcal{S}^{u,j}, \mathcal{P}_u >, l)$, via attention-based mechanisms.

Since both $\mathcal{S}^{u,j}$ and $\mathcal{P}_u$ affect user u's next movement, they can be naturally treated as translation relations inspired by [14], [15]. We, thus, encode the two types of relations derived from the above two factors. Regarding $\mathcal{S}^{u,j}$, we adopt the context-level attention to capture u's personalized varying attentions on different historical check-ins

$$\mathbf{s}^u_{t_i} = f(\mathbf{l}_{t_i}, \mathbf{t}_i, \mathbf{c}_{t_i}, \mathbf{g}_{t_i}, \mathbf{y}_{t_i}), \quad s^u_{t_i} \in \mathcal{S}^{u,j} \tag{2}$$

$$o^s_{t_i} = \mathrm{MLP}_c([\mathbf{u}; \mathbf{s}^u_{t_i}]) \tag{3}$$

$$\alpha_{u,s^u_{t_i}} = \mathrm{softmax}(o^s_{t_i}) = \frac{\exp(o^s_{t_i})}{\sum_{i=1}^k \exp(o^s_{t_i})} \tag{4}$$

where $\mathbf{s}^u_{t_i}$ encodes the context-aware check-in activity[6]; $f(\cdot)$ is the elementwise summation; $\mathrm{MLP}_c(\cdot)$ is a two-layer attention

[3]Following the user simulation settings in [10], [11], we use templates as wrappers to interact with a user for conversation process rather than considering language understanding and generation.

[4]The two terms, i.e., POI and location, are exchangeable in this article.

[5]In this article, we use lowercase in bold (e.g., $\mathbf{h}, \mathbf{r}, \mathbf{t}, \mathbf{u}, \mathbf{l}$) to denote the embedding of the corresponding notation (e.g., h, r, t, u, l).

[6]$\mathbf{c}_{t_i}$ is represented by the weighted combination of embeddings of all categories inside l_{t_i} if $y_{t_i} = 1$ as in [4].

network (see bottom left of Fig. 2), and its input is the concatenation of $\mathbf{u}$ and $\mathbf{s}^u_{t_i}$; the softmax function is used to calculate the normalized impact weight; and k is the length of $\mathcal{S}^{u,j}$. Therefore, the sequential regularity can be encoded by the relation embedding $\mathbf{r}^{\text{seq}}$, which is the weighted sum of the check-in activity embeddings

$$\mathbf{r}^{\text{seq}} = \sum_{i=1}^{k} \alpha_{u,s^u_{t_i}} \cdot \mathbf{s}^u_{t_i}. \tag{5}$$

In our conversation scenario, we can acquire u's immediate preferred attributes $\mathcal{P}_u$ for next location visit at time step t_{k+1}, for example, Bob specifies a *Thai restaurant* (i.e., category) as his preferred attribute. A good relation needs to encode such acquired attributes and the time information to better localize his desired POIs since the time factor is also vital context information in users' next movement prediction [4], [15], e.g., the check-ins of shops usually happen between 10 A.M. and 6 P.M., while the check-ins of nightlife spots are more likely to occur after 7 P.M. We, thus, generate the relation embedding $\mathbf{r}^{\text{imm}}$ based on $\mathcal{P}_u$

$$\mathbf{r}^{\text{imm}} = \sum_{p_i \in \mathcal{P}_u} \mathbf{p}_i + \mathbf{t}_{k+1} \tag{6}$$

where p_i would be one of the preferred attributes at t_{k+1} in the conversation session, e.g., $c_{t_{k+1}}$, $g_{t_{k+1}}$, and $y_{t_{k+1}}$.

Although the above two relations (i.e., r^{seq} and r^{imm}) contribute to the next preferred POI prediction, different users are supposed to have varying attention to different relations due to users' complex behavioral preferences. To capture the importance of such two relations, we further propose a relation-level attention to calculate the impact weight

$$o_{r^i} = \text{MLP}_r([\mathbf{u}; \mathbf{r}^i]) \tag{7}$$

$$\beta_{u,r^i} = \text{softmax}(o_{r^i}) = \frac{\exp(o_{r^i})}{\sum_{i=1}^{|\mathcal{R}|} \exp(o_{r^i})} \tag{8}$$

where $\text{MLP}_r(\cdot)$ is a two-layer attention network (see top left of Fig. 2), and its input is the concatenation of $\mathbf{u}$ and $\mathbf{r}^i$; $|\mathcal{R}|$ is the number of relations, and two types of relations r^{seq} and r^{imm} are considered in our study. As a result, the final relation embedding $\mathbf{r}$ in (1) is formulated as

$$\mathbf{r} = \sum_{i=1}^{|\mathcal{R}|} \beta_{u,r^i} \mathbf{r}^i. \tag{9}$$

Finally, given u's historical sequential behaviors $\mathcal{S}^{u,j}$ and immediate preference $\mathcal{P}_u$ in the conversation session, the probability of u visiting $l_{t_{k+1}}$ at t_{k+1} is predicted by

$$P_{u,l_{t_{k+1}}} \propto \frac{1}{d(u, l_{t_{k+1}})}. \tag{10}$$

B. Conversation Module

The conversation module consists of a *State Vector*, a *Policy Network*, and a *User Response*. The *State Vector* serves as the bridge between the recommender and conversation modules; the *Policy Network* aims to learn the strategy of action selection to determine whether to ask an attribute or make a recommendation; and the *User Response* returns user feedback on such an action selection, whereby the immediate user preference can be obtained.

1) State Vector: The state vector encodes a user's preference over different attributes and the conversation history. As such, the state vector $\mathbf{v}$ is represented by the concatenation of three components

$$\mathbf{v} = [\mathbf{v}_{\text{pre}}; \mathbf{v}_{\text{his}}; \mathbf{v}_{\text{cand}}] \tag{11}$$

$$\tilde{d}(u, p) = \left\| (\mathbf{u} + \sum_{p_i \in \mathcal{P}_u} \mathbf{p}_i + \mathbf{t}_{k+1}) - \mathbf{p} \right\|_2^2 \tag{12}$$

where each dimension of $\mathbf{v}_{\text{pre}} \in \mathbb{R}^{|\mathcal{P}|}$ encodes the user preference on each attribute p, evaluated by (12); $|\mathcal{P}|$ is the size of all attributes.[7] The term $(\mathbf{u} + \sum_{p_i \in \mathcal{P}_u} \mathbf{p}_i + \mathbf{t}_{k+1})$ represents the context-aware user representation by considering user u's immediate preference $\mathcal{P}_u$ at time step t_{k+1}. $\mathbf{v}_{\text{his}} \in \mathbb{R}^N$ (N is the maximum conversation turn) encodes the conversation history, where each dimension encodes u's feedback (i.e., reward) at each turn n, and we will detail the settings on reward in the *Policy Network*. $\mathbf{v}_{\text{cand}}$ encodes the size of current candidate POI set ($|\mathcal{L}_{\text{cand}}|$) via the binary features [10], where $|\mathcal{L}_{\text{cand}}|$ will be reduced with the increasing of preferred attributes in $\mathcal{P}_u$, and the recommendation action will be triggered if $|\mathcal{L}_{\text{cand}}|$ is smaller than a threshold ($L_{\min}$) to avoid asking more attributes.

2) Policy Network and User Response: The action selection is derived from the policy network in the conversation module. Inspired by [5], [10], we adopt a two-layer neural network that can be optimized with the standard policy gradient method. Specifically, the input is generated by the *State Vector*, and the output is normalized to be a probability distribution over all actions by a softmax operation. Our action space consists of all attributes of POIs and a recommendation action, i.e., $\mathcal{A}^{|\mathcal{P}|+1} = \{\{a_{\text{ask}}(p), p \in \mathcal{P}\} \cup a_{\text{rec}}\}$. At each turn, the conversation module will take an action determined by the *Policy Network* and receive an up-to-date reward from the *User Response* (i.e., a user provides positive/negative response on either a_{ask} or a_{rec}). We define the rewards as follows.

1) $\text{Re}^+_{\text{ask}}/\text{Re}^-_{\text{ask}}$*:* A positive/negative reward when the user gives positive/negative feedback on the asked attribute.
2) $\text{Re}^+_{\text{rec}}/\text{Re}^-_{\text{rec}}$*:* A positive/negative reward when the user accepts/rejects the recommended POIs.
3) $\text{Re}^-_{\text{quit}}$*:* A strongly negative reward if the user quits the conversation, i.e., reaching the maximum turn N.

In the scenario of uncertain check-ins, the proposed CRS interacts with a user to obtain his immediate preference, e.g., the desired activity and POI type, which helps to locate the desired individual POI inside a collective POI. We, thus, specially design an auxiliary reward w.r.t. Re^+_{rec} inspired by the observations in [4]. That is, user's check-in behaviors at a collective POI l are affected by the diversity of l since users usually prefer collective POIs with more choices. Such an auxiliary reward is considered when the user accepts the

[7]The attributes considered in this study include categories, geographical regions, and types of POIs.

recommended collective POI l

$$\mathrm{Re}^{+}_{\mathrm{rec}} = \begin{cases} \mathrm{Re}^{+}_{\mathrm{rec}}, & y = 0 \\ \mathrm{Re}^{+}_{\mathrm{rec}} + \omega \dfrac{m_p}{M_l}, & y = 1 \end{cases} \tag{13}$$

where (m_p/M_l) is the auxiliary reward; m_p is the number of individual POIs w.r.t. the preferred attribute p, e.g., two individual POIs (l_4 and l_5) belong to Thai restaurant in Fig. 1(b); M_l is the total number of individual POIs in l, e.g., $M_{l_3} = 8$, as l_3 contains eight individual POIs; and ω is the normalization coefficient.

We denote the policy network as π_θ, and $\pi_\theta(a_n|\mathbf{v}_n)$ represents the probability of taking an action $a_n \in \mathcal{A}$ given the state $\mathbf{v}_n$ at the nth turn. Following [51], the policy network is optimized through reinforcement learning (RL)

$$\theta \leftarrow \theta + \eta \bigtriangledown \log \pi_\theta(a_n|\mathbf{v}_n)\bar{R}e_n, \quad \theta \in \Theta \tag{14}$$

where $\theta \in \Theta$ is the parameter of the policy network; η is the learning rate; $\bar{R}e_n = \sum_{n'=n}^{N} \gamma^{n'} \mathrm{Re}_{n'}$ is the sum of rewards from turn n to the final turn N; and γ is a discount factor.

C. Connection of Recommender and Conversation Modules

Once a recommendation action is triggered in the conversation module, the recommender needs to suggest top-K POIs for user u. To this end, it first constructs the relation vector $\mathbf{r}$ derived from user u's historical check-ins $\mathcal{S}^{u,j}$ and immediate preference $\mathcal{P}_u$ in conversations using (5–9) and then calculates the probability of user u visiting each candidate POI based on (10). Note that the rejected POIs in conversations will be treated as negative samples to update the recommender. In the scenario of uncertain check-ins, considering that user u's desired POI could be a collective POI l, we propose a rating-based sampling strategy, i.e., $l_i \sim \mathrm{rate}(l_i)/\sum_{i=1}^{M_l} \mathrm{rate}(l_i)$, to select the individual POI $l_{t_{k+1}}$ corresponding to user u's preferred attributes $\mathcal{P}_u$ inside POI l, where $\mathrm{rate}(l_i)$ is the rating of POI l_i. This helps achieve a relatively fair recommendation rather than simply selecting such individual POIs with higher ratings as in [4].

D. Discussion on the Issue of Uncertain Check-Ins

We highlight the overview of CART in dealing with the issue of uncertain check-ins in the next POI recommendation from three aspects: 1) the uncertain check-ins over collective POIs increase the difficulty of predicting accurate user activity (i.e., category) of next movement; the proposed CRS enables the system to interact with users to acquire, for example, their desired activities and POI types, which is one of the key challenges for a successful next POI recommendation under the uncertain check-in scenario; 2) in the scenario of uncertain check-ins, user's check-in behaviors at a collective POI l are affected by the diversity of l since users usually prefer collective POIs with more choices [4]; we, thus, specially design an auxiliary reward when the user accepts the recommended collective POI l; and 3) considering that a user's desired POI l could be a collective POI, we need to further select an individual POI inside l. Hence, we design a rating-based sampling strategy for fairly choosing individual POI corresponding to the user's activity inside a collective POI, rather than simply selecting individual POIs with higher ratings.

E. Model Optimization

Most existing static next POI recommendation studies directly treat the noninteracted POIs as negative samples to optimize the Bayesian personalized ranking (BPR) or binary cross-entropy (BCE) objectives [4], [17], [22], [52], [53], assuming that the interacted POIs should gain higher prediction scores than those not being interacted with. However, users do not explicitly indicate that they dislike those noninteracted POIs, which results in the lack of true negative samples for such static models. In CRS, we are not only able to acquire a user's immediate preference but also explicitly obtain rejected POIs from his response. We, thus, take such rejected POIs as the negative samples to train the recommender. The loss function is defined via a BPR loss

$$\mathcal{J} = -\sum_{u \in \mathcal{U}} \sum_{\mathcal{S}^{u,j} \in \mathcal{S}^u} \sum_{l' \in \mathcal{L}^u_{\mathrm{rej}}} \ln \sigma(P_{u,l_{t_{k+1}}} - P_{u,l'}) + \lambda ||\bar{\Theta}||^2 \tag{15}$$

where $\mathcal{L}^u_{\mathrm{rej}}$ is a set of rejected POIs for u received from conversations; λ is the regularization coefficient; and $\bar{\Theta}$ is the parameter set of the recommender.

Note that, we DO NOT train our proposed CART from scratch. Following [10], [11], our training process comprises two stages: 1) the off-line training aims to pretrain the recommender and conversation modules on the training set, where the maximum entropy strategy is used for determining which attribute to ask and 2) in the online training stage, the pretrained CRS is optimized by interacting with the user simulator through the conversation module on the training set. Algorithm 1 shows the online training of the CART. In particular, at each conversation turn n, the conversation module performs state generation and action selection (lines 6 and 7). If an action a_n asking an attribute p (i.e., $a_{\mathrm{ask}}(p)$) is accepted by u, p will be added into u's preferred attribute set $\mathcal{P}_u$; otherwise, $a_{\mathrm{ask}}(p)$ will be removed from u's action space $\mathcal{A}_u$ (lines 8–12). If a_n is making a recommendation (i.e., a_{rec}), then the top-K POI list $\mathcal{L}^u_{\mathrm{rec}}$ is generated by the recommender. The conversation succeeds and quits if $\mathcal{L}^u_{\mathrm{rec}}$ contains u's desired POI; otherwise, such rejected POIs are removed from the POI candidate set $\mathcal{L}^u_{\mathrm{cand}}$ but added into $\mathcal{L}^u_{\mathrm{rej}}$ (explicit negative samples) for updating the recommender (lines 13–20). Finally, the policy network gets updated if the conversation session succeeds or quits (line 21). Moreover, once the model has been well-trained, CART performs incremental training with the ever-growing of users' check-in records, instead of updating it from scratch. In particular, if there are new check-in records for some users coming, we will only update the relevant model parameters (e.g., user and POI embeddings) for these users.

IV. Experiments

To evaluate the performance of our proposed CART, we conduct experiments on four real-world datasets with the goal of answering the following research questions.[8]

[8]Our code is available at: http://github.com/CART2020/CART

Algorithm 1 Online Training for CART

Input: $\mathcal{U}, \mathcal{L}, \mathcal{S}, N, \mathcal{P}, L_{min}, \omega, \eta, \gamma, \lambda$
Output: Top-K recommendation list
Initialize model parameters Θ and $\bar{\Theta}$;
foreach $u \in \mathcal{U}$ **do**
  foreach *turn* $n = 1, 2, \ldots, N$ **do**
    if $n > N$ **then**
      Fail & quit
    // State Vector
    Compute state $\mathbf{v}_n$ according to Eq.(11)
    // Policy Network
    Take an action $a_n \sim \pi_\theta(\cdot|\mathbf{v}_n)$
    // User Response
    if $a_n = a_{ask}(p)$ **then**
      if *u accepts p* **then**
        $\mathcal{P}_u = \mathcal{P}_u \cup p$, get a reward Re^{+}_{ask}
      else
        $\mathcal{A}_u = \mathcal{A}_u \setminus a_{ask}(p)$, get a reward Re^{-}_{ask}
    else
      Recommend top-K POIs $\mathcal{L}^u_{rec} \subset \mathcal{L}^u_{cand}$
      if *u accepts* $\mathcal{L}^u_{rec}$ **then**
        Succeed & quit, get a reward Re^{+}_{rec}
      else
        $\mathcal{L}^u_{cand} = \mathcal{L}^u_{cand} \setminus \mathcal{L}^u_{rej}$
        $\mathcal{L}^u_{rej} = \mathcal{L}^u_{rec}$, get a reward Re^{-}_{rec}
        Update recommender $\bar{\Theta}$ using $\mathcal{L}^u_{rej}$
  Update policy network Θ

RQ1: Does our proposed CART outperform the representative CRS methods?
RQ2: How does our CART compared with state-of-the-art static recommendation methods?
RQ3: How does the quality of user response affect CART?
RQ4: How do different components of CART affect its performance?
RQ5: How do different hyper-parameters affect CART?

A. Experimental Setup

1) Datasets: We perform experiments on four real-world datasets utilized in [3], [4], which contains collective POIs constructed from Foursquare in four cities, i.e., Calgary (CAL), Charlotte (CHA), Phoenix (PHO), and Singapore (SIN),[9] as summarized in Table II. These are the only available public datasets for the study on uncertain check-ins. Following [10], [14], we remove the users with less than ten interactions and split the sequences of each user in the ratio of 7:2:1 for training, validation, and test.

TABLE II
STATISTICS OF THE FOUR DATASETS

Dataset	#User	#POI	#1st-Layer	#2nd-Layer	#Check-in
CAL	301	985	9	184	13,954
CHA	1,580	1,791	10	239	20,940
PHO	1,623	2,441	8	251	22,620
SIN	2,676	3,440	15	264	116,757

2) Action Space: In real applications, asking for an attribute from a large attribute space leads to lengthy conversations. Following [10], we consider the first-layer category provided by *Foursquare Venue Category Hierarchy*[10] as the parent attributes, e.g., "food" is the first-layer category that contains several second-layer child categories {"Korean restaurant," "Malay restaurant," ... , "Thai restaurant"}. As depicted by Table II, the number of first-layer categories is far less than that of the second-layer category. In our location-service scenario, the parent attribute of geographical region contains child attributes, such as $\{g_1, g_2, g_3, g_4\}$ in Fig. 1(b). Similarly, the parent attribute of POI type contains such child attributes $y = \{0, 1\}$. As a result, we obtain the shrinking attribute space, which only contains the parent attributes of category, geographical region, and POI type. Taking CAL as an example, its final output (i.e., action space) size of the policy network is 12, that is, 9 (first-layer category) + 1 (geographical region) + 1 (POI type) + 1 (recommendation) = 12. Following [10], the system selects a parent attribute to ask, and the user can reply with multiple child attributes.

3) User Simulator: The CRS needs to interact with users to make recommendations, which is expensive to acquire real dialog resources [9]. Following [5], [10], we create a user simulator to enable CART to be trained and evaluated in the interactive process. Given u's historical sequential behaviors $\mathcal{S}^{u,j}$ (e.g., $l_1 \rightarrow l_2$), the user simulator aims to simulate a conversation session for an observed interaction (u, l), where l is treated as u's desired POI to seek for next movement and $\mathcal{P}_l$ is the oracle set of attributes preferred by u in this session. As such, the session is initialized by a randomly selected attribute from $\mathcal{P}_l$ and then goes in the loop, as shown in Algorithm 1.

4) Evaluation Metrics: We adopt two standard metrics to evaluate the CRS methods by following [10]: 1) success rate (SR@n), which measures the ratio of successful conversations, i.e., correctly recommending the desired POI by turn n and 2) average turns (AT), which records the average turns (i.e. average conversation length) needed to end the session. As such, higher SR denotes better recommendation, and smaller AT represents more efficient conversations. Specifically, SR@n and AT are defined by

$$\text{SR@}n = \frac{\sum_{i=1}^{n} \#\text{successful\_conversations}(i)}{\#\text{conversations}} \tag{16}$$

$$\text{AT} = \frac{\sum_{i=1}^{n} i \times \#\text{successful\_conversations}(i)}{\sum_{i=1}^{n} \#\text{successful\_conversations}(i)} \tag{17}$$

where #successful_conversations(i) is the number of successful conversations (i.e., successful test samples) at turn i among

[9]Note that there is no rating information in SIN dataset, we, thus, use the popularity (i.e., the check-in frequency of a POI) as a substitute since it is a significant factor in recommendation [54]. Specifically, we evenly divided such frequency into five groups corresponding to the range of ratings [1, 5].

[10]https://developer.foursquare.com/docs/build-with-foursquare/categories/

all the test samples; #conversations is equal to the number of all test samples. In addition, to compare the performance of the static state of the arts with CART, we adopt recall (Recall@K) and mean reciprocal rank (MRR@K) to measure the ranking quality for the top-K recommended POIs. Note that we calculate the recall and MRR of top-K POIs by turn n for the CART, that is, the recall and MRR are the accumulated scores from conversation turn 1 to turn n. Note that we calculate the recall and MRR for the CART after the conversation finishes.

5) Baselines: Following [10], we compare the proposed CART with five state-of-the-art CRS methods.

1) *Max Entropy:* It is a rule-based method that aims to select an attribute with the maximum entropy within the current candidate POIs, and such system follows certain probability to make recommendations as in [10].
2) *CRM [5]:* It is a single-round CRS method that consists of a belief tracker to encode the state vector and the policy network to perform action selection. We follow [10] to adapt it to a multiround setting without considering the natural language understanding module; therefore, the adapted CRM serves as an upper bound study.
3) *Qrec [8]:* It is a question-based recommender system based on the extended matrix factorization, which is originally designed for interactively asking questions based on the descriptions and reviews of items. To generate the question pool, we use the item attributes in this work.
4) *EAR [10]:* It is a state-of-the-art approach with a multiround CRS setting, where a three-stage solution is devised and its recommender is built upon factorization machine [12].
5) *SCPR [11]:* It is a CRS method that performs interactive path reasoning on a graph under the multiround setting.

The recent CRS methods, such as [6], [30], [33] introduced in Section II, are beyond the scope of our study since they either let the CRS recommend items without asking users' preferred attributes or focus on natural language understanding and generation. Besides, we compare with nine state-of-the-art static methods for next POI recommendation:

1) *MostPop:* It recommends the next POI via popularity.
2) *LBPR [53]:* It is a listwise Bayesian personalized ranking method, which predicts the next category and POI in a twofold way.
3) *ST-RNN [52]:* It is an RNN-based method that models temporal and spatial context with time- and distance-specific transition matrices.
4) *MCARNN [19]:* It is a multitask learning framework to capture a user's activity and location preferences.
5) *STA [15]:* It is a translation-based method to model the users' spatiotemporal context.
6) *PLSPL [25]:* It is a unified framework to characterize a user's long-term and short-term preferences.
7) *HCT [3]:* It is the pioneering work to handle the next POI recommendation with uncertain check-ins.
8) *iMTL [4]:* It is a multitask learning framework to jointly model a user's context of check-in behaviors and targets at resolving the issue of uncertain check-ins.
9) *CART-CRS:* It is a variant of our CART, which removes the conversion module and only encodes a user's historical sequential behaviors (see the "Recommender module" in Fig. 2) for the next POI recommendation.

6) Training Details and Parameter Settings: We tune all hyperparameters on the validation set and empirically find out the optimal parameter settings for all the methods. Specifically, the embedding size is searched in the range of $[20, 200]$ stepped by 10; the learning rate and regularization coefficient are searched in $\{0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05, 0.1\}$. For our proposed CART, the size of embeddings is 50, which are randomly initialized over uniform distribution; we adopt SGD optimizer with learning rate and regularization coefficient being 0.001; the rewards are empirically set as $\mathrm{Re}^{+}_{\mathrm{ask}} = 1$, $\mathrm{Re}^{-}_{\mathrm{ask}} = -1$, $\mathrm{Re}^{+}_{\mathrm{rec}} = 2$, $\mathrm{Re}^{-}_{\mathrm{rec}} = -2$, and $\mathrm{Re}^{-}_{\mathrm{quit}} = -3$; the discount factor $\gamma = 0.7$; the normalization coefficient $\omega = 2$; the max turn $N = 10$; the threshold to trigger recommendation action $L_{\min} = 10$; the length of the recommendation list $K = 10$; and the number of recommended individual POI inside a collective POI $K_1 = 3$. For the settings of CRS baselines (i.e., CRM, Qrec, EAR, and SCPR), the embedding size is $\{60, 100, 60, 60\}$; the learning rate and regularization coefficient are $\{0.001, 0.1, 0.01, 0.01\}$ and $\{0.01, 0.01, 0.001, 0.001\}$, respectively; and other parameters are set as suggested in the related articles. Regarding the settings of static baselines, the embedding size and the learning rate for LBPR, ST-RNN, MCARNN, STA, PLSPL, HCT, and iMTL are set to $\{100, 100, 200, 100, 100, 200, 120\}$ and $\{0.001, 0.01, 0.01, 0.0001, 0.001, 0.001, 0.0001\}$, respectively; the list size of LBPR is set to 2; the window size of HCT is set to 2; and the weights of categories at layer 1 and layer 2 are 0.2 and 0.8, which are selected in [0, 1] stepped by 0.1. Furthermore, to illustrate the robustness of the results, we have run the experiments under the same setup with different random seeds ten times and reported the overall performances by mean $\pm$ standard_deviation in Tables III–V.

B. Performance Comparison for CRS Methods (RQ1)

Table III presents the results of all CRS methods on SR@10 and AT across the four datasets. Fig. 3 presents the recommendation success rate (SR*@n) at different turns ($n = 1$ to 10), where SR* denotes the difference of each method with the strongest baseline SCPR, and the SR*@n for SCPR is set to be zero for ease of presentation. As a whole, we can observe that our proposed CART achieves the best performance in comparison with state-of-the-art baselines. This firmly demonstrates the superiority of our proposed CART, that is, generating high-quality recommendations with fewer conversation turns.

Specifically, all methods start with SR $= 0$, as they tend to accumulate a user's immediate preference (i.e., attributes) at the beginning of a conversation session. Max Entropy and Qrec generally underperform (RL)-based baselines (CRM, EAR, SCPR, and CART), which indicates that learning a better conversational strategy benefits the action selection, i.e., whether to ask an attribute or make recommendations. EAR and SCPR perform better than CRM, implying the efficacy

TABLE III

PERFORMANCE (REPRESENTED BY MEANS AND STANDARD DEVIATIONS) OF ALL METHODS ON THE FOUR DATASETS MEASURED BY SR@10 AND AT, WHERE THE BEST PERFORMANCE IS BOLDFACED, AND THE RUNNER-UP IS UNDERLINED

	CAL		CHA		PHO		SIN	
	SR@10	AT	SR@10	AT	SR@10	AT	SR@10	AT
Max Entropy	0.686±0.003	6.13±0.636	0.501±0.001	6.31±0.671	0.503±0.007	6.30±0.836	0.427±0.003	6.89±0.772
CRM	0.699±0.024	4.57±0.511	0.466±0.017	4.59±0.782	0.495±0.052	4.31±0.935	0.634±0.014	5.61±0.823
Qrec	0.702±0.011	5.40±0.220	0.675±0.003	4.56±0.669	0.742±0.013	6.04±0.610	0.765±0.005	5.70±0.440
EAR	0.715±0.019	4.53±1.040	0.691±0.008	4.56±0.653	0.721±0.022	4.29±0.682	0.806±0.009	4.46±0.545
SCPR	0.884±0.010	3.42±0.340	0.922±0.012	2.97±0.722	0.963±0.002	3.65±0.656	0.907±0.012	4.29±0.633
CART	**0.937±0.010**	**3.13±0.500**	**0.948±0.012**	**2.54±0.978**	**0.975±0.004**	**3.58±0.891**	**0.922±0.012**	**4.14±0.642**

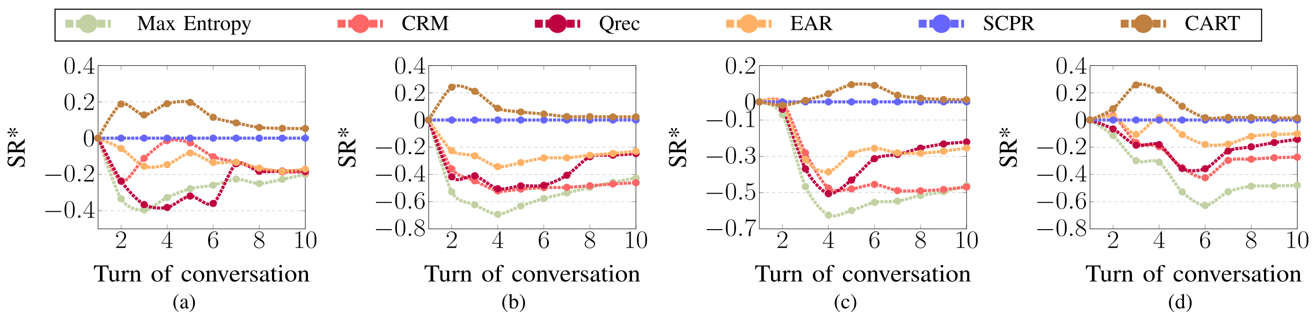


Fig. 3. success rate* (SR*@n) of CRS methods at each conversation turn. SR*@n refers to the difference of each method w.r.t. SCPR. (a) CAL. (b) CHA. (c) PHO. (d) SIN.

of encoding both user preference and conversation history for the state vector of the policy network. Our CART outperforms the strongest baseline SCPR, mainly due to three reasons: 1) it delicately models both historical sequential behaviors and immediate preference via a translation-based method, which benefits the next POI recommendation by absorbing rich context information and overcoming the limitation of the FM-based recommender; 2) it designs an auxiliary reward when recommending a desired collective POI to well accommodate the fact that users usually prefer collective POIs with more choices; and 3) it adopts the rating-based sampling strategy for recommending next individual POIs within the desired collective POI in a fair way (i.e., making such POIs with low ratings have a chance to be recommended).

C. Performance Comparison for Static Methods (RQ2)

We compare the static methods with our proposed CART on the four datasets evaluated by Recall@K and MRR@K ($K = \{5, 10\}$). We follow the setting in [4] to select individual POIs within a predicted collective POI via the ratings for these static methods. Regarding CART, we examine its performance under different conversation turns from 1 to 10. The results are shown in Fig. 4, where we use two y-axes to visualize the results of the static methods (i.e., bar) and the proposed CART (i.e., blue line), respectively. Note that the scale of the left y-axis is generally in the range of $(0, 0.08)$, while the right one is in the range of $(0, 0.9)$.

Our CART significantly improves the performance compared with these static methods, which implies the effectiveness of dynamic interaction (i.e., conversation module) for more accurate recommendations. Specifically, it far exceeds the runner-up iMTL by ten and 17 times w.r.t. MRR@10

TABLE IV

PERFORMANCE COMPARISON OF THE CART AND CART-rand WITH FIXED MAXIMUM TURN AND RANDOM TURN, RESPECTIVELY

	CART		CART-$rand$	
	SR@10	AT	SR@$\hat{N}$	AT
CAL	0.937±0.010	3.13±0.500	0.725±0.005	3.46±0.420
CHA	0.948±0.012	2.54±0.978	0.763±0.003	2.76±0.383
PHO	0.975±0.004	3.58±0.891	0.804±0.003	2.52±0.579
SIN	0.922±0.012	4.14±0.642	0.752±0.016	4.23±0.614

and Recall@10 on average within three turns. That is, CART is able to achieve better next POI recommendations after clarifying the user's immediate preference (i.e., attributes) within three turns. It shows the great merit brought by CRS to the location-based recommendation community. For the comparison of static methods, baselines (MostPop, ST-RNN, and PLSPL) ignoring the user activity (i.e., category of POIs) generally perform worse than those considering the user activity (e.g., LBPR, MCARNN, HCT, iMTL, and CART-CRS). This verifies the efficacy of modeling a user's activity in the next POI recommendation. Both STA and CART-CRS generally outperform LBPR, which suggests the superior generalization ability of the translation-based method for a better recommendation. Moreover, regarding the multitask-based baselines, iMTL performs better than MCARNN, as it is tailored to model a user's uncertain activity and perform POI type prediction, thus helping more accurate activity and POI prediction.

D. Study on Quality of User Response (RQ3)

In this section, we aim to evaluate the impact of the quality of user response on our proposed CART via simulating a user's

TABLE V

PERFORMANCE (REPRESENTED BY MEANS AND STANDARD DEVIATIONS) OF DIFFERENT VARIANTS OF CART W.R.T. SR AND AT ACROSS THE FOUR DATASETS, WHERE THE BEST PERFORMANCE IS BOLDFACED, AND THE RUNNER-UP IS UNDERLINED

	CAL					CHA				
	SR@3	SR@5	SR@7	SR@10	AT	SR@3	SR@5	SR@7	SR@10	AT
CART-$\mathbf{r}^{seq}$	0.029±0.011	0.228±0.045	0.515±0.042	0.918±0.005	7.45±0.442	0.160±0.034	0.367±0.023	0.527±0.016	0.909±0.019	6.09±0.863
CART-r_attn	0.017±0.004	0.228±0.040	0.339±0.034	0.632±0.004	8.55±0.450	0.670±0.042	0.887±0.006	0.891±0.003	0.893±0.024	3.15±0.901
CART(FM)	0.023±0.013	0.415±0.063	0.479±0.020	0.778±0.009	6.24±0.604	0.127±0.021	0.356±0.036	0.472±0.013	0.847±0.021	6.15±0.828
CART-$\mathbf{v}_{pre}$	0.257±0.082	**0.894±0.038**	**0.912±0.012**	0.929±0.005	4.39±0.497	0.614±0.042	0.684±0.014	0.921±0.018	0.946±0.020	4.31±0.934
CART-$\mathbf{v}_{his}$	0.281±0.076	0.707±0.094	0.784±0.007	0.865±0.005	5.55±0.546	0.544±0.031	0.790±0.023	0.915±0.004	0.946±0.009	3.97±0.905
CART-$\mathbf{v}_{cand}$	**0.731±0.052**	0.818±0.063	0.847±0.003	0.865±0.010	3.77±0.281	0.826±0.025	0.925±0.010	0.943±0.011	**0.951±0.018**	2.75±0.832
CART-neg	0.327±0.067	0.679±0.068	0.790±0.010	0.825±0.007	4.62±0.345	0.104±0.019	0.239±0.007	0.488±0.005	0.745±0.013	6.57±0.770
CART-$auxi$	0.677±0.031	0.846±0.026	0.870±0.003	0.914±0.005	3.42±0.458	0.811±0.006	0.893±0.010	0.920±0.012	0.943±0.010	2.64±0.855
CART-$samp$	0.655±0.060	0.789±0.020	0.824±0.004	0.836±0.011	3.27±0.530	0.762±0.034	0.764±0.015	0.767±0.020	0.769±0.022	2.56±0.966
CART	0.695±0.047	0.865±0.036	0.877±0.006	**0.937±0.010**	**3.13±0.500**	**0.905±0.009**	**0.939±0.011**	**0.946±0.014**	0.948±0.012	**2.54±0.978**

	PHO					SIN				
	SR@3	SR@5	SR@7	SR@10	AT	SR@3	SR@5	SR@7	SR@10	AT
CART-$\mathbf{r}^{seq}$	0.306±0.020	0.845±0.014	0.948±0.010	0.970±0.013	4.41±0.790	0.282±0.007	0.821±0.030	0.868±0.026	0.903±0.010	5.26±0.800
CART-r_attn	0.401±0.035	0.651±0.010	0.802±0.017	0.949±0.014	4.67±0.811	0.415±0.028	0.816±0.022	0.872±0.011	0.895±0.008	4.61±0.762
CART(FM)	0.329±0.008	0.705±0.004	0.725±0.004	0.753±0.002	4.34±0.616	0.218±0.010	0.637±0.016	0.753±0.005	0.858±0.011	5.39±0.804
CART-$\mathbf{v}_{pre}$	0.308±0.016	0.547±0.010	0.831±0.015	0.971±0.011	5.57±0.752	0.449±0.040	0.838±0.035	0.880±0.006	0.914±0.012	4.34±0.647
CART-$\mathbf{v}_{his}$	0.507±0.024	0.908±0.008	0.972±0.010	**0.978±0.005**	3.61±0.704	0.473±0.028	0.774±0.012	0.863±0.010	0.887±0.008	4.62±0.755
CART-$\mathbf{v}_{cand}$	0.574±0.020	0.653±0.024	0.820±0.020	0.946±0.012	4.80±0.733	0.555±0.030	0.767±0.027	0.881±0.018	0.913±0.014	4.93±0.813
CART-neg	0.263±0.010	0.489±0.012	0.625±0.004	0.764±0.005	4.58±0.845	0.303±0.014	0.771±0.013	0.825±0.010	0.848±0.006	5.01±0.688
CART-$auxi$	0.580±0.011	0.866±0.012	0.955±0.008	0.962±0.005	3.90±0.725	0.658±0.016	0.822±0.010	0.885±0.012	0.910±0.012	4.57±0.720
CART-$samp$	0.582±0.025	0.721±0.018	0.741±0.012	0.748±0.010	3.82±0.633	0.662±0.020	0.786±0.010	0.890±0.015	0.892±0.014	4.88±0.705
CART	**0.594±0.006**	**0.971±0.010**	**0.973±0.004**	0.975±0.004	**3.58±0.891**	**0.711±0.025**	**0.876±0.020**	**0.920±0.010**	**0.922±0.012**	**4.14±0.642**

random interaction behaviors in the testing process.[11] The user simulator is widely employed to simulate conversation sessions [5], [10], [11]. However, such a user simulator is built upon an assumption: the user would clearly express his preferences by responding to each question (i.e., asked attributes and recommendations) at each conversation turn until he accepts the recommendations or chooses to quit (i.e., reaching the maximum turn). There is no denying that such simulation has many limitations, but it is the most practical way at the current off-line research stage [11].

However, in the real application scenario, one major issue is that the user may randomly interact with the system during the conversation sessions, instead of keeping interactions, as stated in the above assumption. We, therefore, consider the following three cases in the testing process for a more comprehensive study: 1) *no interactions:* a user does not respond to either the asked attributes or recommendations; 2) *random interactions:* a user randomly interacts with the system, for example, he may respond only once to the asked attributes or recommendations; thus, it is hard to identify a user's preference or know whether the recommendations meet his need; and 3) *full interactions:* a user keeps interacting with the system until he accepts the recommendation or chooses to quit.

In this study, our focus is to build a location-aware CRS by employing a user simulator following the state-of-the-art [10] under case (3), for seeking the best strategy of action selection, so as to achieve more accurate next POI recommendations in fewer conversation turns. Hence, our work serves as an upper bound study of practical applications as we do not consider the real users' random behaviors during the interaction process. However, our proposed CART also works in cases (1) and (2). Specifically, in case (1), CART will degenerate to the static recommender to provide recommendations merely based on a user's historical check-in behaviors (see the "Recommender Module" in Fig. 2). In case (2), to simulate a user's random interaction behaviors, we randomly sample a conversation turn $\hat{N}$ in the range of [1, 10]. That is, a user would randomly interact with the system by $\hat{N}$ turns instead of the fixed maximum turn N (i.e., $N = 10$).

Table IV presents the comparison results of our CART in cases (2) and (3). Note that the performance on case (1), that is, the degenerated CRS (i.e., CART-CRS), is shown in the comparison of static recommendation baselines (see Fig. 4). From the table, we notice that overall CART performs better than CART-rand, as it is challenging for accurate recommendations within very limited conversation turns (e.g., CART could recommend a correct POI with four turns, but it may fail when sampling a small conversation turn $\hat{N} = 2$). In addition, we observe that CART-rand generally outperforms both CRM and Qrec that adopt the fixed maximum turn, as indicated in Table III. This suggests the effectiveness of CART on learning conversational strategy in action selection, thus achieving successful recommendations with fewer conversation turns.

E. Ablation Study (RQ4)

To examine the efficacy of different components of CART, we compare its different variants from four aspects: 1) ablating important components from the recommender module, where

[11]We do not simulate a user's random interaction behaviors in our model training because we can control the data quality in the training process.

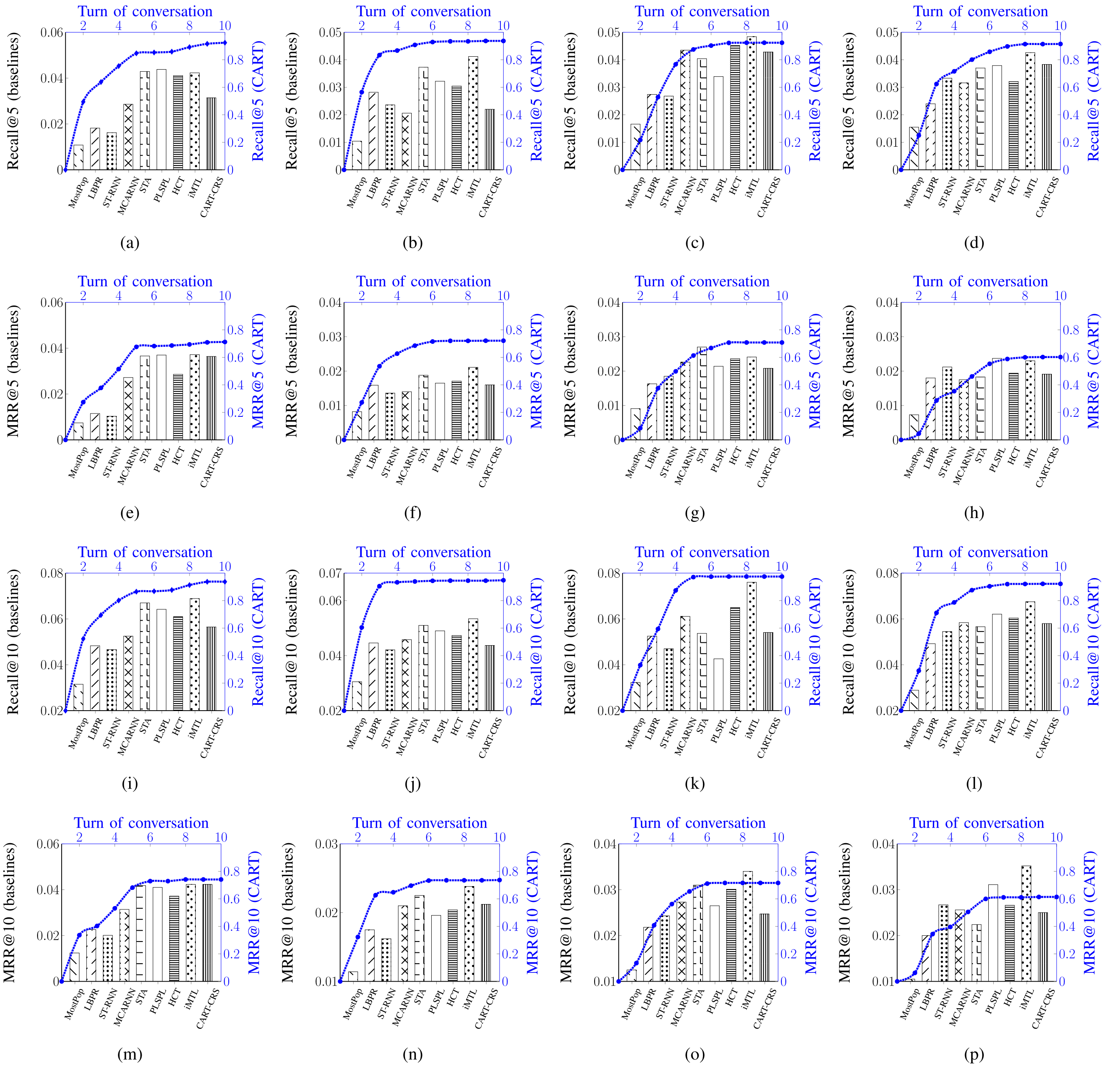


Fig. 4. Performance comparison between static recommendation baselines and the proposed CART w.r.t. Recall and MRR@5/10. (a) CAL-Recall@5. (b) CHA-Recall@5. (c) PHO-Recall@5. (d) SIN-Recall@5. (e) CAL-MRR@5. (f) CHA-MRR@5. (g) PHO-MRR@5. (h) SIN-MRR@5. (i) CAL-Recall@10. (j) CHA-Recall@10. (k) PHO-Recall@10. (l) SIN-Recall@10. (m) CAL-MRR@10. (n) CHA-MRR@10. (o) PHO-MRR@10. (p) SIN-MRR@10.

CART-$\mathbf{r}^{\text{seq}}$ removes the sequential relation; CART-r_atten omits the relation-level attention but directly performs sum aggregation for r^{seq} and r^{imm}; and CART(FM) indicates that the translation-based method is replaced by the FM-based method as adopted in state-of-the-art EAR [10]; 2) ablating each component in the state vector, where CART-$\mathbf{v}_{\text{pre}}$ removes the vector of user preference; CART-$\mathbf{v}_{\text{his}}$ omits the vector of conversation history; and CART-$\mathbf{v}_{\text{cand}}$ discards the vector encoding the size of candidate POIs; 3) testing the impact of online update, where CART-neg implies that we do not update the recommender by the immediate rejected POIs in conversations as negative samples; and 4) validating the efficacy of the designed auxiliary reward and rating-based sampling strategy, where CART-auxi indicates that we do not consider the auxiliary reward in (13), and CART-samp means that we simply select individual POIs within a collective POI via higher rating as in [4].

The results are reported in Table V, where the CART generally outperforms its variants. Following the above four aspects, several interesting findings can be noted.

1) The CART without considering $\mathbf{r}^{\text{seq}}$ or r_attn performs worse than itself, e.g., less SR at the beginning turns (SR@5 in CAL) and larger AT in CAL and CHA. This indicates modeling the user's sequential behaviors and considering the attentive impacts of different relations indeed boost the performance. Besides, CART achieves

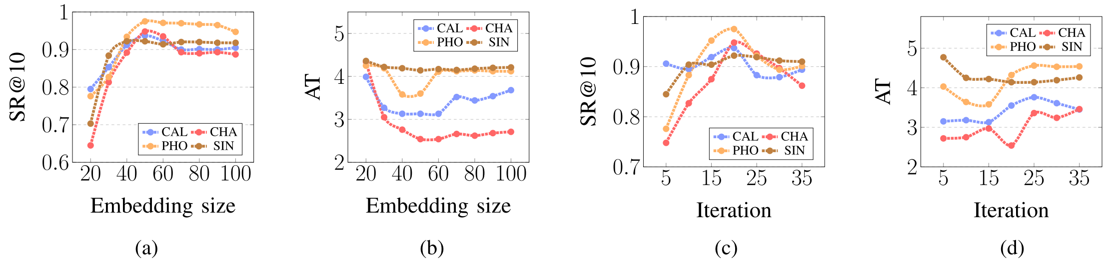


Fig. 5. Results of parameter sensitivity. (a) and (b) Dimensionality. (c) and (d) Convergence.

better performance than CART(FM) under the same CRS settings, which suggests the superior generalization ability of translation-based methods over FM-based methods.

2) By removing $\mathbf{v}_{\text{pre}}$ or $\mathbf{v}_{\text{his}}$, the value of SR drops heavily at the several beginning turns on the four datasets while gradually increasing at the future turns, which naturally results in high AT (i.e., lengthy conversations). The possible reason is that, without $\mathbf{v}_{\text{pre}}$ and $\mathbf{v}_{\text{his}}$ as prior knowledge, the system needs to ask more attributes before making a recommendation; meanwhile, the candidate POI length $\mathbf{v}_{\text{cand}}$ is important since it assists in deciding when to recommend.
3) The performance decrease of CART-neg on SR (with a drop of 30.2% on average) and AT (with a drop of 37.1% on average) indicating the advantage of the online update.
4) CART-auxi and CART-samp underperform CART on SR but achieve comparable AT, which shows that the design of auxiliary reward and the rating-based sampling strategy are beneficial for a better next POI recommendation.

F. Hyper-Parameter Analysis (RQ5)

Fig. 5 reports results regarding the parameter sensitivity on SR@10 and AT across the four datasets. Fig. 5(a) and (b) presents the performance of CART with varying embedding size (other optimal hyper-parameters fixed), where the best performance achieves with the embedding size around 50. Fig. 5(c) and (d) shows the convergence property of the CART, and we observe that it can converge within 20 iterations on the four datasets.

V. Conclusion

In this article, we propose a novel CART, which is the first work to commence conversation-based location service over uncertain check-ins. In particular, the adaptive relational translation-based recommender aims to model a user's historical sequential behaviors and immediate preference received from conversations, and the conversation module seeks the conversational strategy for action selection and achieves successful recommendations with fewer conversation turns. Meanwhile, we design the auxiliary reward and rating-based sampling strategy to delicately consider the presence of collective POI in our scenario. Experimental results on real-world datasets show the superiority of CART in providing more accurate next POI recommendations over uncertain check-ins.

References

[1] P. Zhao *et al.*, "Where to go next: A spatio-temporal gated network for next POI recommendation," in *Proc. 33rd AAAI Conf. Artif. Intell. (AAAI)*, 2019, pp. 5877–5884.

[2] Q. Guo, Z. Sun, J. Zhang, and Y.-L. Theng, "An attentional recurrent neural network for personalized next location recommendation," in *Proc. 34th AAAI Conf. Artif. Intell. (AAAI)*, 2020, vol. 34, no. 1, pp. 83–90.

[3] L. Zhang, Z. Sun, J. Zhang, H. Kloeden, and F. Klanner, "Modeling hierarchical category transition for next POI recommendation with uncertain check-ins," *Inf. Sci.*, vol. 515, pp. 169–190, Apr. 2020.

[4] L. Zhang *et al.*, "An interactive multi-task learning framework for next POI recommendation with uncertain check-ins," in *Proc. 29th Int. Joint Conf. Artif. Intell. (IJCAI)*, Jul. 2020, pp. 3551–3557.

[5] Y. Sun and Y. Zhang, "Conversational recommender system," in *Proc. 41st Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (SIGIR)*, Jun. 2018, pp. 235–244.

[6] Y. Zhang, X. Chen, Q. Ai, L. Yang, and W. B. Croft, "Towards conversational search and recommendation: System ask, user respond," in *Proc. 27th ACM Int. Conf. Inf. Knowl. Manage. (CIKM)*, Oct. 2018, pp. 177–186.

[7] B. Priyogi, "Preference elicitation strategy for conversational recommender system," in *Proc. 12th ACM Int. Conf. Web Search Data Mining (WSDM)*, Jan. 2019, pp. 824–825.

[8] J. Zou, Y. Chen, and E. Kanoulas, "Towards question-based recommender systems," in *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (SIGIR)*, Jul. 2020, pp. 881–890.

[9] S. Li, W. Lei, Q. Wu, X. He, P. Jiang, and T.-S. Chua, "Seamlessly unifying attributes and items: Conversational recommendation for cold-start users," 2020, *arXiv:2005.12979*.

[10] W. Lei *et al.*, "Estimation-action-reflection: Towards deep interaction between conversational and recommender systems," in *Proc. 13th Int. Conf. Web Search Data Mining (WSDM)*, Jan. 2020, pp. 304–312.

[11] W. Lei *et al.*, "Interactive path reasoning on graph for conversational recommendation," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (SIGKDD)*, Aug. 2020, pp. 2073–2083.

[12] S. Rendle, "Factorization machines," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Dec. 2010, pp. 995–1000.

[13] C.-K. Hsieh, L. Yang, Y. Cui, T.-Y. Lin, S. Belongie, and D. Estrin, "Collaborative metric learning," in *Proc. 26th Int. Conf. World Wide Web (WWW)*, Apr. 2017, pp. 193–201.

[14] B. Wu, X. He, Z. Sun, L. Chen, and Y. Ye, "ATM: An attentive translation model for next-item recommendation," *IEEE Trans. Ind. Informat.*, vol. 16, no. 3, pp. 1448–1459, Mar. 2020.

[15] T. Qian, B. Liu, Q. V. H. Nguyen, and H. Yin, "Spatiotemporal representation learning for translation-based POI recommendation," *ACM Trans. Inf. Syst.*, vol. 37, no. 2, pp. 1–24, Mar. 2019.

[16] W. Liu, Z.-J. Wang, B. Yao, and J. Yin, "Geo-ALM: POI recommendation by fusing geographical information and adversarial learning mechanism," in *Proc. 28th Int. Joint Conf. Artif. Intell. (IJCAI)*, Aug. 2019, pp. 1807–1813.

[17] F. Yu, L. Cui, W. Guo, X. Lu, Q. Li, and H. Lu, "A category-aware deep model for successive POI recommendation on sparse check-in data," in *Proc. 29th Int. World Wide Web Conf. (WWW)*, 2020, pp. 1264–1274.

[18] J. Feng *et al.*, "DeepMove: Predicting human mobility with attentional recurrent networks," in *Proc. 27th World Wide Web Conf. (WWW)*, 2018, pp. 1459–1468.

[19] D. Liao, W. Liu, Y. Zhong, J. Li, and G. Wang, "Predicting activity and location with multi-task context aware recurrent neural network," in *Proc. 27th Int. Joint Conf. Artif. Intell. (IJCAI)*, 2018, pp. 3435–3441.

[20] L. Huang, Y. Ma, S. Wang, and Y. Liu, "An attention-based spatiotemporal LSTM network for next POI recommendation," *IEEE Trans. Services Comput.*, vol. 14, no. 6, pp. 1585–1597, Nov. 2021.

[21] W. Liang and W. Zhang, "Learning social relations and spatiotemporal trajectories for next check-in inference," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Oct. 20, 2020, doi: 10.1109/TNNLS.2020.3016737.

[22] Y. Luo, Q. Liu, and Z. Liu, "STAN: Spatio-temporal attention network for next location recommendation," in *Proc. Web Conf. WWW*, 2021, pp. 2177–2185.

[23] H. Wang, H. Shen, W. Ouyang, and X. Cheng, "Exploiting POI-specific geographical influence for point-of-interest recommendation," in *Proc. 27th Int. Joint Conf. Artif. Intell. (IJCAI)*, 2018, pp. 3877–3883.

[24] H. Luo *et al.*, "Spatial object recommendation with hints: When spatial granularity matters," in *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (SIGIR)*, 2020, pp. 781–790.

[25] Y. Wu, K. Li, G. Zhao, and X. Qian, "Personalized long- and short-term preference learning for next POI recommendation," *IEEE Trans. Knowl. Data Eng.*, early access, Jun. 15, 2020, doi: 10.1109/TKDE.2020.3002531.

[26] Y. Li, S. Wang, Q. Pan, H. Peng, T. Yang, and E. Cambria, "Learning binary codes with neural collaborative filtering for efficient recommendation systems," *Knowl.-Based Syst.*, vol. 172, pp. 64–75, May 2019.

[27] C. Angulo, I. Z. Falomir, D. Anguita, N. Agell, and E. Cambria, "Bridging cognitive models and recommender systems," *Cognit. Comput.*, vol. 12, no. 2, pp. 426–427, Mar. 2020.

[28] E. Cambria, D. Das, S. Bandyopadhyay, and A. Feraco, "Affective computing and sentiment analysis," in *A Practical Guide to Sentiment Analysis*. Cham, Switzerland: Springer, 2017, pp. 1–10.

[29] C. Gao, W. Lei, X. He, M. de Rijke, and T.-S. Chua, "Advances and challenges in conversational recommender systems: A survey," 2021, *arXiv:2101.09459*.

[30] K. Christakopoulou, F. Radlinski, and K. Hofmann, "Towards conversational recommender systems," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (SIGKDD)*, Aug. 2016, pp. 815–824.

[31] C. Greco, A. Suglia, P. Basile, and G. Semeraro, "Converse-et-impera: Exploiting deep learning and hierarchical reinforcement learning for conversational recommender systems," in *Proc. Conf. Italian Assoc. Artif. Intell.* Cham, Switzerland: Springer, 2017, pp. 372–386.

[32] K. Christakopoulou, A. Beutel, R. Li, S. Jain, and E. H. Chi, "Q&R: A two-stage approach toward interactive recommendation," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 139–148.

[33] L. Liao, Y. Ma, X. He, R. Hong, and T.-S. Chua, "Knowledge-aware multimodal dialogue systems," in *Proc. 26th ACM Int. Conf. Multimedia*, Oct. 2018, pp. 801–809.

[34] R. Li, S. E. Kahou, H. Schulz, V. Michalski, L. Charlin, and C. Pal, "Towards deep conversational recommendations," in *Proc. 32nd Conf. Neural Inf. Process. Syst. (NIPS)*, 2018, pp. 9725–9735.

[35] T. Yu, Y. Shen, and H. Jin, "A visual dialog augmented interactive recommender system," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2019, pp. 157–165.

[36] N. Sardella, C. Biancalana, A. Micarelli, and G. Sansonetti, "An approach to conversational recommendation of restaurants," in *Proc. Int. Conf. Hum.-Comput. Interact. (HCI)*. Cham, Switzerland: Springer, 2019, pp. 123–130.

[37] X. Zhang, H. Xie, H. Li, and J. C. S. Lui, "Toward building conversational recommender systems: A contextual bandit approach," 2019, *arXiv:1906.01219*.

[38] Q. Chen *et al.*, "Towards knowledge-based recommender dialog system," 2019, *arXiv:1908.05391*.

[39] Z. Liu, H. Wang, Z.-Y. Niu, H. Wu, W. Che, and T. Liu, "Towards conversational recommendation over multi-type dialogs," 2020, *arXiv:2005.03954*.

[40] K. Xu, J. Yang, J. Xu, S. Gao, J. Guo, and J.-R. Wen, "Adapting user preference to online feedback in multi-round conversational recommendation," in *Proc. 14th ACM Int. Conf. Web Search Data Mining (WSDM)*, Mar. 2021, pp. 364–372.

[41] X. Zhang, H. Xie, H. Li, and J. CS Lui, "Conversational contextual bandit: Algorithm and application," in *Proc. 29th Int. World Wide Web Conf. (WWW)*, 2020, pp. 662–672.

[42] O. Chapelle and L. Li, "An empirical evaluation of Thompson sampling," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 24, 2011, pp. 2249–2257.

[43] N. Mrkšić, D. O. Séaghdha, T.-H. Wen, B. Thomson, and S. Young, "Neural belief tracker: Data-driven dialogue state tracking," 2016, *arXiv:1606.03777*.

[44] T. Young, E. Cambria, I. Chaturvedi, H. Zhou, S. Biswas, and M. Huang, "Augmenting end-to-end dialogue systems with commonsense knowledge," in *Proc. 32nd AAAI Conf. Artif. Intell. (AAAI)*, 2018, pp. 4970–4977.

[45] H. Xu, H. Peng, H. Xie, E. Cambria, L. Zhou, and W. Zheng, "End-to-end latent-variable task-oriented dialogue system with exact log-likelihood optimization," *World Wide Web*, vol. 23, no. 3, pp. 1989–2002, May 2020.

[46] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," in *Proc. 27th Conf. Neural Inf. Process. Syst. (NIPS)*, 2013, pp. 2787–2795.

[47] X. Yang, Y. Ma, L. Liao, M. Wang, and T.-S. Chua, "TransNFCM: Translation-based neural fashion compatibility modeling," in *Proc. 33rd AAAI Conf. Artif. Intell. (AAAI)*, 2019, vol. 33, no. 1, pp. 403–410.

[48] Z. Wang, J. Zhang, J. Feng, and Z. Chen, "Knowledge graph embedding by translating on hyperplanes," in *Proc. 28th AAAI Conf. Artif. Intell. (AAAI)*, 2014, vol. 28, no. 1, pp. 1112–1119.

[49] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu, "Learning entity and relation embeddings for knowledge graph completion," in *Proc. 29th AAAI Conf. Artif. Intell. (AAAI)*, 2015, vol. 29, no. 1, pp. 2181–2187.

[50] Z. Sun, Z.-H. Deng, J.-Y. Nie, and J. Tang, "RotatE: Knowledge graph embedding by relational rotation in complex space," 2019, *arXiv:1902.10197*.

[51] R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2000, pp. 1057–1063.

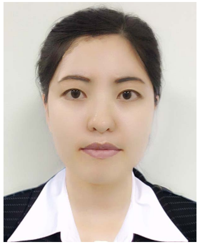
[52] Q. Liu, S. Wu, L. Wang, and T. Tan, "Predicting the next location: A recurrent model with spatial and temporal contexts," in *Proc. 30th AAAI Conf. Artif. Intell. (AAAI)*, 2016, pp. 194–200.

[53] J. He, X. Li, and L. Liao, "Category-aware next point-of-interest recommendation via listwise Bayesian personalized ranking," in *Proc. 26th Int. Joint Conf. Artif. Intell. (IJCAI)*, vol. 17, 2017, pp. 1837–1843.

[54] M. F. Dacrema, P. Cremonesi, and D. Jannach, "Are we really making much progress? A worrying analysis of recent neural recommendation approaches," in *Proc. 13th ACM Conf. Recommender Syst. (RecSys)*, 2019, pp. 101–109.

Lu Zhang is currently pursuing the Ph.D. degree with Nanyang Technological University, Singapore, under the supervision of Associate Professor J. Zhang.

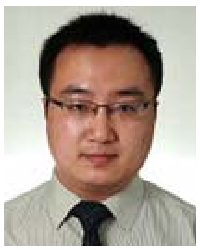
She is currently working on the next point-of-interest (POI) recommendation with uncertain check-ins by leveraging machine learning and data mining techniques. Her research has been published in top-tier international conferences and journals, including IJCAI, Information Sciences, and TKDE.



Zhu Sun received the Ph.D. degree from Nanyang Technological University, Singapore, in 2018.

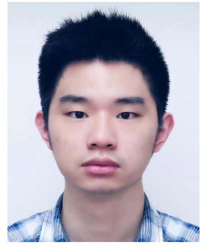
Her research has been published in leading conferences and journals in related domains (e.g., International Joint Conference on Artificial Intelligence (IJCAI), ACM International Conference on Information and Knowledge Management (CIKM), ACM Recommender Systems Conference (RecSys), IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING (TKDE), and IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS (TNNLS). Her research interests include machine learning, data mining, and recommender systems.

Ms. Sun is the AE with Electronic Commerce Research and Applications (ECRA) and a Program Committee Member for IJCAI, AAAI, ACM CIKM, ACM Recsys, and others.



Jie Zhang received the Ph.D. degree from the Cheriton School of Computer Science, University of Waterloo, Waterloo, ON, Canada, in 2009.

He is currently an Associate Professor with Nanyang Technological University, Singapore. He is also an Associate of the Singapore Institute of Manufacturing Technology (SIMTech), A*STAR, Singapore. His articles have been published by top journals and conferences and won several best paper awards.



Yiwen Wu received the bachelor's degree (Hons.) from the University of Auckland, New Zealand, in 2019, and the master's degree in artificial intelligence from Nanyang Technological University, Singapore, in 2021, under the supervision of Associate Prof. J. Zhang.

Yunwen Xia is currently pursuing the M.Eng. degree with Nanyang Technological University, Singapore, under the supervision of Assoc. Prof. Jie Zhang.

She is currently working on a conversational recommender system by leveraging graph embedding techniques.