Adaptive Vague Preference Policy Learning for Multi-round Conversational Recommendation

Gangyi Zhang[1], Chongming Gao[1], Wenqiang Lei[2], Xiaojie Guo[3], Shijun Li[1], Lingfei Wu[3], Hongshen Chen[4], Zhuozhi Ding[4], Sulong Xu[4] and Xiangnan He[1]
[1]University of Science and Technology of China, [2] Sichuan University, [3] JD.COM Silicon Valley Research Center, [4] JD.COM

ABSTRACT

Conversational recommendation systems (CRS) effectively address information asymmetry by dynamically eliciting user preferences through multi-turn interactions. Existing CRS widely assumes that users have clear preferences, i.e., users have a firm belief about the fine-grained preference for one or multiple target items. This assumption leads the agent to overly trust user feedback, treating accepts/rejects as definitive signals to filter items and reduce the candidate space, potentially causing over-filtering. However, in reality, users' preferences are often vague and volatile, with vagueness about their desires and changing decisions during interactions.

To address this issue, we introduce a novel scenario called Vague Preference Multi-round Conversational Recommendation (VPMCR), which considers users' vague and volatile preferences in CRS. VPMCR employs a soft estimation mechanism to assign a non-zero confidence score for all candidate items to be displayed, naturally avoiding the over-filtering problem. In the VPMCR setting, we introduce a solution called Adaptive Vague Preference Policy Learning (AVPPL), which consists of two main components: Ambiguity-aware Soft Estimation (ASE) and Dynamism-aware Policy Learning (DPL). ASE estimates the vagueness of users' vague feedback and captures their dynamic preferences using a choice-based preferences extraction module and a time-aware decaying strategy. DPL leverages the preference distribution estimated by ASE to guide the conversation and adapt to changes in users' preferences to make recommendations or ask for attributes.

Our extensive experiments demonstrate the effectiveness of our method in the VPMCR scenario, highlighting its potential for practical applications and improving the overall performance and applicability of CRS in real-world settings, particularly for users with vague or dynamic preferences.

CCS CONCEPTS

• **Information systems** → **Users and interactive retrieval**; **Recommender systems**; *Personalization*; • **Human-centered computing** → *Interactive systems and tools.*

xxxx '18, June 03–05, 2018, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

KEYWORDS

Conversational Recommendation; Vague Preference; Policy Learning

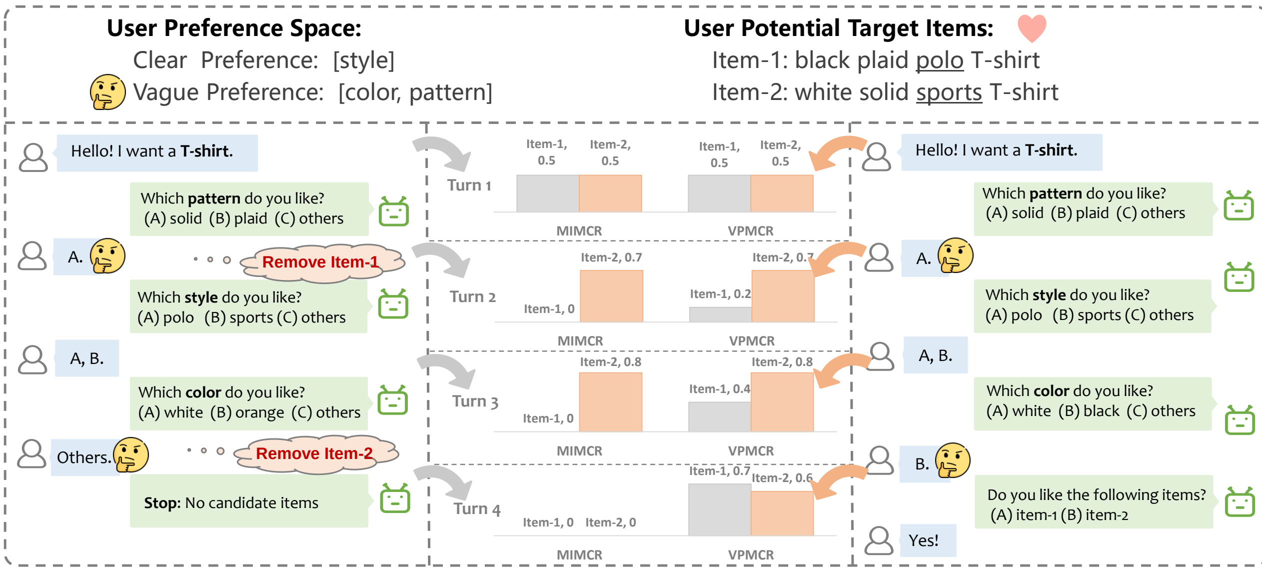
ACM Reference Format:
Gangyi Zhang[1], Chongming Gao[1], Wenqiang Lei[2], Xiaojie Guo[3], Shijun Li[1], Lingfei Wu[3], Hongshen Chen[4], Zhuozhi Ding[4], Sulong Xu[4] and Xiangnan He[1]. 2018. Adaptive Vague Preference Policy Learning for Multi-round Conversational Recommendation. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (xxxx '18).* ACM, New York, NY, USA, 10 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Conversational recommendation systems (CRS) have drawn a lot of research attention recently. These systems interact with users to elicit preferences, understand motivations, and address the long-standing information asymmetry problem [10]. Despite considerable progress, CRS is far from mature, and researchers have focused on specific scenarios [16, 26, 37] to address particular challenges.

One widely adopted scenario [16, 17, 34] is **Multi-round Conversational Recommendation (MCR)**, where the system can ask for attributes or make recommendations multiple times, and the user accepts or rejects accordingly. However, MCR assumes that users have a clear single preferred item in mind, which may not be realistic, as users may have more than one preferred item in mind. To address this, the **Multi-Interest Multi-round Conversational Recommendation (MIMCR) scenario** [37] and **Multi-round Multi-Groundtruth (MGMCR) scenario** [20] were proposed, allowing users to have multiple preferences and accept multiple items. In this setting, a user may accept multiple attribute instances (e.g., red and black) of an attribute type (e.g., color). Despite the improvement, MIMCR (or MGMCR) can still fall short because it assumes that users have clear preferences in mind during the conversation. This can be impractical as users' preferences can be vague or change dynamically over time, leading to randomness in their answers and potential regret for previous choices.

In practical applications, users exhibit vague or dynamic preferences, but MIMCR (or MGMCR) fails to account for the vagueness in users' feedback, treating it as a hard indicator to filter the candidate item set. This results in over-filtering, as numerous potential items are removed when the user selects or does not select corresponding attributes. In Fig. 1 (a), we illustrate a toy example showing a conversation (tailored for vague settings) under the MIMCR scenario. The CRS incorrectly interprets the user's non-clicking attributes (i.e., "plaid" in the first turn) and removes potential target items (i.e., "item-1" in the first turn), causing the user's preference distribution over items to collapse suddenly as shown in the left side of Fig. 1 (b). This wrong inference will naturally affect the reasoning



(a) CRS for MIMCR (b) Preference Distribution (c) CRS for VPMCR

Figure 1: A realistic user simulation example

of the subsequent conversation, leading to the wrong preference estimation (i.e., in Fig. 1 (a), the "black" color of "item-1" was not displayed in the third turn).

To address over-filtering in MIMCR (or MGMCR) and maintain diversity and accuracy in the CRS, we propose a new scenario called **Vague Preference Multi-round Conversational Recommendation (VPMCR)**. This scenario uses a soft estimation mechanism to account for users' vague or dynamic preferences by assigning non-zero confidence scores to all candidate items, avoiding the rigid filtering strategy of MIMCR (or MGMCR) and MCR. Fig. 1 (c) shows an example of the VPMCR, which, in contrast to MIMCR, captures changes in preference distribution of the entire item space as shown in the right side of Fig. 1 (b).

In the VPMCR scenario, several challenges need to be addressed, including estimating the vagueness of the user's vague feedback, capturing the user's dynamic preference throughout the conversation, and making conversational decisions that consider the user's vague or dynamic preferences. To tackle these challenges, we propose an enhanced solution called **Adaptive Vague Preference Policy Learning (AVPPL)**, which consists of:

1. **Ambiguity-aware Soft Estimation (ASE)**: ASE estimates the vagueness of the user's vague feedback in each turn using a choice-based preference extraction method. It captures both explicit and implicit preferences (distinguished based on whether the user explicitly clicks the choices), effectively estimating the vagueness of users' vague feedback. To capture users' dynamic preferences, ASE employs a time-aware preference decay strategy, which gives more weight to recent preferences while gradually reducing the influence of historical preferences.

2. **Dynamism-aware Policy Learning (DPL)**: DPL implements a policy learning framework, leveraging the preference distribution from ASE, to guide the conversation. It constructs a dynamic heterogeneous graph representing the conversation, with ASE's soft estimation scores as edge weights. To expedite graph modeling and policy learning, we introduce a graph sampling strategy and preference-guided action pruning.

In summary, our contributions are as follows:

- We identify the limitations of existing CRS settings and introduce the VPMCR scenario, which accounts for users' vague and volatile preferences in CRS.
- We propose the AVPPL solution for the VPMCR setting, utilizing a unified policy learning framework to make decisions that consider users' current vague preferences and account for their fading historical preferences.
- Our extensive experiments on four real-world datasets demonstrate the effectiveness of AVPPL in the VPMCR scenario, highlighting its potential for practical applications.

2 RELATED WORK

We briefly introduce the related works in conversational recommendation, reinforcement learning, and graph learning.

2.1 Conversational recommendation system

(CRSs) is a novel solution to recommendation that leverage natural language to effectively elicit dynamic user preferences that align with their real needs through multiple rounds of real-time interaction. CRS is considered to be a cutting-edge discipline that incorporates dialogue systems, recommendation systems, and interactive systems [10]. According to the focus on different functions and settings, existing CSR methods can be roughly divided into two types: dialogue-based recommendation [4, 18, 31, 39, 41] and multi-round conversational recommendation (MCR) [7, 11, 14, 17, 19, 34]. In this work, we focus on the MCR setting.

MCR is considered to be the most realistic setting in CRS. Unlike dialogue-based recommenders that need to extract information or generate responses through raw natural language [30], MCR focuses on the core logic of the interaction strategy which involves asking questions [24, 42, 42] and making recommendations. The traditional MCR setting allows users to select only one preferred attribute value at a time, which restricts users' expression in the interaction. To overcome this issue, Zhang et al. [37] propose the MIMCR setting, where a user is allowed to select multiple options for a certain attribute. Though effective, they follow the recommendation philosophy in MCR to directly filter out the items that the user has not mentioned by attributes, which leads to failure as users may not be sure what they want precisely. In our proposed VPMCR setting, we specifically consider users' vague preferences and adjust the recommendation mechanism to consider the items with unmentioned attributes, which better reflect users' needs.

2.2 RL-based Recommendation

Reinforcement Learning (RL) is a type of Machine Learning. It considers how an agent (e.g., a machine) should automatically make decisions within a specific context to pursue a long-term goal. The agent learns and adjusts its policy based on the reward feedback (i.e., reinforcement signals) given by the environment. Recently, RL has shown its effectiveness in recommendation [1, 6, 8]. As fitting user interest is not a bottleneck for now, recommenders care more about users' long-term satisfaction [28, 35, 36]. For instance, Montazeralghaem and Allan [22] use RL to generate the proper questions that can maximally make the system help users search desired products. Gao et al. [9] integrate causal inference into offline RL to maximize users' long-term satisfaction by removing filter bubbles. Sadeghi Eshkevari et al. [25] propose an RL-based dispatching solution for ride-hailing platforms that can conduct robust and efficient on-policy learning and inference while being adaptable for full-scale deployment. In this work, we use RL to learn a policy that can automate question-asking and item recommendation.

2.3 Graph-based Recommendation

Graph-based recommender systems have drawn a lot of research attention [5, 12, 21, 32]. By arranging the various entities (e.g., users, items, and attributes) in a heterogeneous graph, we can leverage lots of properties in modeling the collaborative signals. In CRS, the knowledge graph is utilized in enriching the system with additional knowledge [17, 23, 33, 38, 40]. For example, to better understand concepts that a user mentioned, Zhou et al. [38] propose to incorporate two external knowledge graphs (KGs): a word-oriented KG providing relations (e.g., synonyms, antonyms, or co-occurrence) between words and an item-oriented KG carrying structured facts regarding the attributes of items. With the increasing of nodes, the computational overhead is too large to satisfy the requirement of real-time interaction. Hence, we propose a pruning strategy to overcome this work.

3 PROBLEM DEFINITION

Vague Preference Multi-round Conversational Recommendation (VPMCR). In the VPMCR scenario, we consider a dynamic conversation between a user and a conversational recommendation system (CRS). The user has a clear preference space, denoted as C_{CI} (e.g., "style" in Fig. 1), and a vague preference space, denoted as C_{VI} (e.g., "color" and "pattern" in Fig. 1).

The conversation begins with the user specifying a query attribute p_0 (e.g., "T-shirt"), which initializes the candidate item set containing all relevant items (e.g., all "T-shirts") and the candidate attribute set containing all attributes of those items.

During the conversation, the CRS can either ask questions about attributes or provide recommendations. When the CRS asks questions, the user responds accordingly with their behavior depending on whether the attribute type c belongs to their clear or vague preference space. If $c \in C_{CI}$, the user *honestly* accepts or rejects the displayed attributes. However, if $c \in C_{VI}$, the user may *randomly* accept or reject a potentially preferred attribute. When the CRS provides recommendations, the user can accept or reject one or more items from the recommended set $\mathcal{V}_{rec}$.

The conversation proceeds through multiple iterations of the CRS asking/recommending and the user responding, until a successful recommendation is made or the maximum number of turns is reached. The VPMCR scenario differs from previous MCR or MIMCR settings in that it does not filter $\mathcal{V}_{cand}$ based on the user's clicking or non-clicking attributes. Instead, it only removes $\mathcal{V}_{rec}$ from $\mathcal{V}_{cand}$ when the recommendation fails. Additionally, all candidate attributes linked to candidate items are maintained in $\mathcal{P}_{cand}$.

The main challenges in the VPMCR scenario include estimating the vagueness of the user's vague feedback, capturing the user's dynamic preference throughout the conversation, and making conversational decisions that consider the user's vague or dynamic preferences.

4 METHODOLOGY

To address the challenges in the Vague Preference Multi-round Conversational Recommendation (VPMCR) scenario, we propose the *Adaptive Vague Preference Policy Learning (AVPPL)* solution. AVPPL consists of two main components: Ambiguity-aware Soft Estimation (ASE) and Dynamism-aware Policy Learning (DPL). The ASE component estimates the vagueness of users' vague feedback and captures their dynamic preferences, while the DPL component leverages the preference distribution estimated by ASE to guide the conversation and adapt to changes in users' preferences. By incorporating the VPMCR scenario and the AVPPL solution, we aim to improve the overall performance and applicability of conversational recommendation systems in real-world settings, particularly for users with vague or dynamic preferences.

4.1 Ambiguity-aware Soft Estimation

Ambiguity-aware Soft Estimation (ASE) aims to estimate the vagueness of the user's vague feedback in each turn by considering both explicit and implicit preferences. ASE focuses on understanding users' decision-making processes [2], which reflect the trade-offs they make when providing non-binary feedback. To capture users' dynamic preferences throughout the conversation, ASE employs a

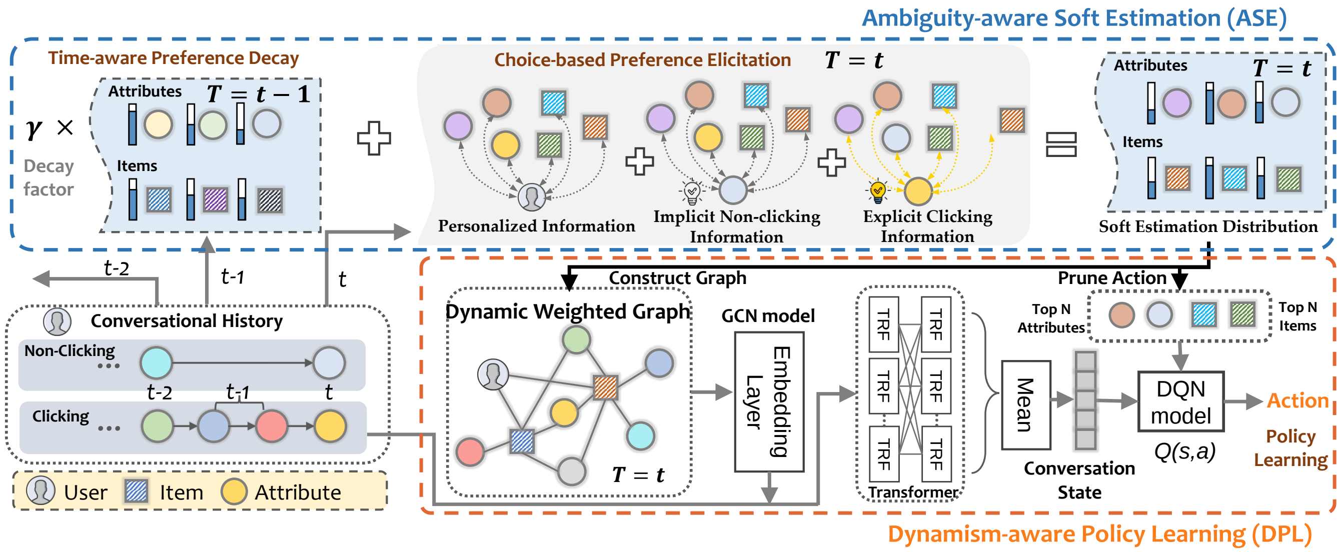


Figure 2: Adaptive Vague Preference Policy Learning (AVPPL) solution for VPMCR scenario.

time-aware preference decay strategy that combines users' recent preferences with fading historical preferences.

In the VPMCR setting, we model the signals of clicking and non-clicking separately based on the decision-making consciousness of users in choice-based questions. For each turn, preference implied by clicking and non-clicking choices is extracted, then the decay mechanism is used to weaken the preference of historical turns. Finally, in the soft estimation, we derive the user's preference distribution toward items and attributes.

4.1.1 Preference Extraction with Choice-based Approach. In each turn of interaction, user preference can be divided into personalized user preference and choice-based preference. We adopt a common personalization modeling strategy [16] to represent the static preference of user u for item v as:

$$w_{v\text{-}u} = e_u^\top e_v, \tag{1}$$

where e_u and e_v denote the embedding vectors of user u and item v, respectively.

To model users' decision-making processes, ASE employs a choice-based preference extraction method that considers the trade-offs users make when providing non-binary feedback. This approach captures both *explicit preferences* (when users actively select an attribute) and *implicit preferences* (when users do not select an attribute but may still have some preference for it) by estimating the importance of clicking choices and non-clicking choices separately.

For item v, we estimate the importance of clicking choices and non-clicking choices, respectively. In turn t, the formula for capturing the user's explicit preference towards clicking choices $\mathcal{P}_{\text{click}}^{(t)}$ and implicit preference towards non-clicking choices $\mathcal{P}_{\text{noclick}}^{(t)}$ are shown as follows:

$$\begin{aligned} w_{v\text{-}click}^{(t)} &= \frac{1}{|\mathcal{P}_{\text{click}}^{(t)}|} \sum_{p \in \mathcal{P}_{\text{click}}^{(t)}} (e_v^\top e_p - w_{v\text{-}avg}^{(t)}), \\ w_{v\text{-}noclick}^{(t)} &= \frac{1}{|\mathcal{P}_{\text{noclick}}^{(t)}|} \sum_{p \in \mathcal{P}_{\text{noclick}}^{(t)}} (e_v^\top e_p - w_{v\text{-}avg}^{(t)}), \end{aligned} \tag{2}$$

where $|\mathcal{P}_{\text{click}}|$ and $|\mathcal{P}_{\text{noclick}}|$ indicates the number of attributes related to clicked items and non-clicked items, respectively. $w_{v\text{-}avg}^{(t)}$ measures the average preference towards all unshown attribute types and is used to mitigate over-estimation of the system-displayed choices, which is defined as:

$$w_{v\text{-}avg}^{(t)} = \sum_{p \in \mathcal{P}_{\text{noshow}}^{(t)}} e_v^\top e_p \Big/ |\mathcal{P}_{\text{noshow}}^{(t)}|, \tag{3}$$

where e_v and e_p represent the embedding vectors of item v and attribute p, respectively, and $\mathcal{P}_{\text{noshow}}^{(t)}$ refers to the set of all unshown attributes associated with the specified attribute type in turn t.

By considering both the personalized preferences and the choice-based preference in turn t, the users' preference for item v in turn t can be calculated as:

$$w_v^{(t)} = \sigma(w_{v\text{-}u} + \lambda_1 w_{v\text{-}click}^{(t)} + \lambda_2 w_{v\text{-}noclick}^{(t)}), \tag{4}$$

where σ is the sigmoid function. λ_1 and λ_2 represent the information intensity coefficients of the information contained in the user's clicked attribute and the user's unclicked attribute, respectively.

4.1.2 Time-aware Preference Decay. In dynamic conversation interactions, the user's global preferences should be viewed as a combination of preferences across all turns. We employ a decay mechanism to adjust the influence of historical preferences, enabling the model to focus more on the user's real-time feedback in the current turn and mitigating the over-emphasized impact related to the user's clicking behavior.

To combine the user's current preference with historical decay preferences, the user's global preference toward the item is estimated as follows:

$$w_v^{(t)} = w_v^{(t)} + \gamma w_v^{(t-1)}, \tag{5}$$

which can be unfolded as:

$$w_v^{(t)} = \sum_{i=0}^{t-1} \gamma^{t-i-1} w_v^{(i)}, \tag{6}$$

where γ is a decay factor satisfying $0 \leq \gamma \leq 1$. The farther the interaction history is from the current turn, the less impact it will have on the current turn. γ should be carefully chosen to balance the influence of historical preferences and the user's real-time feedback.

Finally, for turn t, the user's global preference distribution for items $f_u^{(t)}(v)$ can be calculated by estimating the user's global preference w for each item v in the candidate item set $\mathcal{V}_{\text{cand}}$. When the size of the candidate item set is n, the soft estimation distribution for items is shown as follows:

$$f_u^{(t)}(v) = \{w_{v_1}^{(t)}, w_{v_2}^{(t)}, ..., w_{v_n}^{(t)}\} \tag{7}$$

Similarly, by replacing items with attributes in the aforementioned equations, we derive the user's global preference distribution towards the candidate attribute set $\mathcal{P}_{\text{cand}}$. When the size of the candidate attribute set is m, the soft estimation for attributes is depicted by the following distribution:

$$f_u^{(t)}(p) = \{w_{p_1}^{(t)}, w_{p_2}^{(t)}, ..., w_{p_m}^{(t)}\} \tag{8}$$

4.2 Dynamism-aware Policy Learning (DPL)

The Dynamism-aware Policy Learning (DPL) module utilizes the preference distribution estimated by the Ambiguity-aware Soft Estimation (ASE) module to guide the conversation and adapt to preference changes. The DPL module, as part of the Adaptive Vague Preference Policy Learning (AVPPL) solution, aims to enhance CRS performance for users with vague or dynamic preferences.

4.2.1 Graph-based Conversation Modeling. In the Graph-based Conversation Modeling section, we build on previous work [7, 37] to represent the current conversation state at turn t using a dynamic undirected graph $\mathcal{G}_u^{(t)} = (\mathcal{N}^{(t)}, \mathbf{A}^{(t)})$. we represent the current state of the conversation at turn t using a dynamic undirected graph $\mathcal{G}_u^{(t)} = (\mathcal{N}^{(t)}, \mathbf{A}^{(t)})$. This graph is a subgraph of the heterogeneous graph, which consists of users, items, and attributes.

The nodes in the graph, $\mathcal{N}^{(t)}$, are defined as follows:

$$\mathcal{N}^{(t)} = \{u\} \cup \mathcal{P}_{\text{click}} \cup \mathcal{P}_{n\text{-click}} \cup \mathcal{P}_{\text{cand}}^{(t)} \cup \mathcal{V}_{\text{sample}}^{(t)} \tag{9}$$

The node set $\mathcal{N}^{(t)}$ contains the user, clicked attributes, non-clicked attributes, current candidate attributes, and current sampled candidate items.

To address the issue of a large number of candidate items in the VPMCR setting, we implement a sampling strategy for candidate items $\mathcal{V}_{\text{sample}}^{(t)}$ by randomly selecting from the candidate items in each turn t.

The weighted adjacency matrix, $\mathbf{A}^{(t)}$, is defined as:

$$A_{i,j}^{(t)} = \begin{cases} w_v^{(t)}, & \text{if } n_i = u, n_j \in \mathcal{V} \\ 1, & \text{if } n_i \in \mathcal{V}, n_j \in \mathcal{P} \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

The weight $w_v^{(t)}$ denotes the user's estimated vague preference for the item v, which is calculated via Eq. (6) within the ASE module. The weights of the edge between the item and its associated attributes are set to 1.

A Graph Convolutional Network (GCN) [15] enhances node representations $\mathcal{E}_{\text{node}}$, by capturing the changing interrelationships within the current conversation state $\mathcal{G}_u^{(t)}$. To encode the node representation of the clicking history, $\mathcal{P}_{\text{click}}$, we employ a Transformer [27] to learn sequential patterns of the conversation history. Lastly, we obtain the conversation state $s_{\text{conv}}^{(t)}$ by applying mean pooling to the node embeddings of the Transformer output, as shown in follows:

$$s_{\text{conv}}^{(t)} = \text{MeanPool}(\text{Transformer}(\mathcal{E}_{\mathcal{P}_{\text{click}}})). \tag{11}$$

4.2.2 Vague Preference Policy Learning. We employ a Deep Q-Network (DQN) algorithm to address the challenge of making conversational decisions that consider users' vague or dynamic preferences in CRS. The DQN algorithm has been proven effective in learning action policies in dynamic environments, such as Markov Decision Processes (MDPs), making it well-suited for predicting the next decision based on a series of historical choices.

The Q-value function $Q(s_t, a_t)$ of a policy π is defined to measure the expectation of the accumulated rewards based on the state s and the action a. We adopt the same Dueling DQN and prioritized experience replay as in UNICORN [7] to optimize the Q-function $Q^*(s_t, a_t)$:

$$Q^*(s_t, a_t) = \max_\pi \mathbb{E}[R_{t+1} + \gamma \max_a Q^\pi(s_{t+1}, a) | s_t, a_t] \tag{12}$$

where π is the policy, R_{t+1} is the reward at turn t+1, γ is the discount factor, and $Q^\pi(s_{t+1}, a)$ is the estimated action-value function for the next state and action.

To enhance sampling efficiency, we employ a preference-guided action pruning strategy. Specifically, we select the top-N items $\mathcal{V}_{\text{top}}^{(t)}$ and attributes $\mathcal{P}_{\text{top}}^{(t)}$ with the highest preference scores from ASE to construct a pruned action space. Focusing on likely preferred items and attributes improves learning efficiency. To maintain a balance between efficiency and performance, we adopt the action space size configuration from previous work [7, 37], setting it as $N = 10$. The pruning action space is defined as:

$$\mathcal{A}_{\text{action}}^{(t)} = \mathcal{V}_{\text{top-N}}^{(t)} + \mathcal{P}_{\text{top-N}}^{(t)} \tag{13}$$

For policy learning, the conversation state $s_{\text{conv}}^{(t)}$ captures the user's dynamic conversation state. The pruning action space $\mathcal{A}_{\text{action}}^{(t)}$ is determined by employing a preference-guided action pruning strategy, which partially estimate the user's vague preference distribution. The reward R follows the previous MCR setting [17], and the detailed settings will be described in the Section 5.2.4.

Table 1: Statistics of datasets.

Dataset	Yelp	LastFM	Amazon-Book	MovieLens
#Users	27,675	1,801	30,291	20,892
#Items	70,311	7,432	17,739	16,482
#Interactions	1,368,609	76,693	478,099	454,011
#Attributes	590	8,438	988	1,498
#Attribute-types	29	34	40	24
#Entities	98,576	17,671	49,018	38,872
#Relations	3	4	2	2
#Triplets	2,533,827	228,217	565,068	380,016

5 EXPERIMENTS

In this section, we evaluate the proposed method in VPMCR. We use the following research questions (RQs) to guide our experiment.

- **RQ1.** How does our AVPPL method perform in comparison to state-of-the-art CRS methods in the VPMCR scenario?
- **RQ2.** How do the key components contribute to the overall performance of our AVPPL method?
- **RQ3.** How do the hyperparameters of our method affect its performance?

5.1 Dataset Description

We introduce four datasets, whose statistics are shown in table 1.

- **Yelp and LastFM [16]:** Yelp[1] and LastFM[2] datasets are used for business and music artist recommendations, respectively. We follow the multiple attribute question settings, retaining the original attribute instances and extracting the attribute types they depend on. In Yelp, we utilize the 2-layer taxonomy designed by [16], resulting in 29 categories in the first layer as attribute types and 590 attributes in the second layer as attribute instances. For LastFM, we follow [37], retaining the original 8,438 attributes as attribute instances and employing clustering to obtain 34 attribute types.
- **Amazon-Book [29]:** Amazon Book[3] is a widely used product recommendation dataset. We retain users and items with at least 10 interaction records and consider entities (e.g., science fiction) and relations (e.g., genre) in the knowledge graph as attribute instances and attribute types, respectively.
- **MovieLens:** Movielens is a movie rating dataset. We adopt MovieLens-20M[4] dataset, following [37], retaining interactions > 3 and selecting knowledge graph (KG) entities and relations as attribute instances and attribute types.

5.2 Experimental Setup

5.2.1 User Simulator in VPMCR. Conversational recommendation systems (CRSs) are interactive and require training and evaluation through user interactions. However, obtaining data directly from users in a research lab is impractical, so employing a user simulator is a common practice [3]. The user simulator simulates users' interaction records in the training and test sets.

[1]https://www.yelp.com/dataset/
[2]https://grouplens.org/datasets/hetrec-2011/
[3]http://jmcauley.ucsd.edu/data/amazon.
[4]https://grouplens.org/datasets/movielens/

In the VPMCR scenario, we adopt a user simulation strategy similar to that in MIMCR [37], considering the reasonableness of the multi-interest setting. For a given observed user-items interaction pair $(u, \mathcal{V}_u)$, we simulate a conversation session. Each item v in $\mathcal{V}_u$ is treated as a ground-truth target item, and the union of attribute types and attributes associated with each item are considered as the user's ground-truth intent space C_u and ground-truth attribute space $\mathcal{P}$, respectively. The conversation session is initialized when the user specifies a common attribute p_0 to all $\mathcal{V}_u$, and the user's clear preference space C_{CI} and user's vague preference space C_{VI} are randomly initialized from the ground-truth intent space C_u.

During the interaction, we use the ground-truth attribute space $\mathcal{P}$ as a criterion for the user simulator's acceptance or rejection. The detailed interaction process follows the "system asks or recommends and user responds" rules outlined in Section 3.

5.2.2 Action Inference. The action inference involves either recommending items or asking an attribute-related question.

(1) **Recommendation**: If an item v in the action space has the highest Q-value, the CRS make a recommendation, resulting in a new action space $\mathcal{A}^{(t)} = \mathcal{V}_{top}^{(t)}$.

(2) **Questioning**: If an attribute p in the action space has the highest Q-value, the CRS asks a question. In a multiple-choice setting, a two-level decision process is employed: first selecting an attribute type, then presenting several attributes within that type. A sum-based strategy [37] is used to determine the attribute type for questioning. Specifically, Q-values of all attributes within the attribute action space $\mathcal{P}_{top}^{(t)}$ are summed and allocated to their respective attribute types. The attribute type with the highest total value is selected for questioning, and the top K attributes with the highest Q-values within that type are presented to the user.

5.2.3 Baselines. We use the following baselines. For fairness, all baselines are compared in the VPMCR scenario.

- **Max Entropy**. It selects the attribute with the maximum information entropy and inversely relates the probability of making a recommendation to the length of candidate items.
- **CRM** [26]. It employs a belief tracker to record user preferences as conversation state representation vectors and applies them to a reinforcement learning decision module and factorization machine (FM) recommendation modules.
- **EAR** [16]. This method adopts the three-stage solution framework to enhance the interaction between the conversation component and the recommendation component.
- **SCPR** [17]. SCPR leverages graph-based path reasoning to prune useless candidate attributes. It separates attribute selection from reinforcement learning, which is only used for determining when to ask and recommend.
- **UNICORN** [7]. A state-of-the-art method for the MCR scenario that proposes a unified policy learning framework using dynamic graphs to model conversation states and employs a preference-based scoring to reduce reinforcement learning action space.
- **MCMIPL** [37]. It considers the user's multi-interest space and extends the MCR scenario to a more realistic MIMCR scenario. This method also follows the graph-based unified reinforcement learning framework and employs the multi-interest encoder to learn the conversation state.

5.2.4 Training Details. We split each dataset into training, validation, and testing sets (7:1.5:1.5). In the user simulator, we set the maximum conversation turn T to 15 and the number of target item sets $\mathcal{V}_u$ for the user to 2. We use uniform sampling to initialize the user's vague and clear preference spaces.

In the ASE module, we set the intensity coefficients λ_1 and λ_2 to 0.1 and 0.01, respectively, and the decay discount factor to 0.1. In the DPL module, when constructing the dynamic graph, random sampling is employed to select candidate items when the available number of candidates exceeds 5000. The graph-based conversation modeling architecture consists of two GNN layers and one Transformer layer. We fix the embedding size and hidden size at 64 and 100, respectively. For action pruning in RL, we set the size of the item space and attribute space to 10 (i.e., $N = 10$). For action inference, we set the number of attributes displayed to the user to 2 (i.e., $K = 2$). Following [7], we use TransE [?], implemented throughKE [13], to pre-train the graph node embeddings. For fair comparison, we train the DQN for 10,000 episodes using the same reward settings as the benchmarks: $r_{\text{rec-suc}} = 1$, $r_{\text{rec-fail}} = -0.01$, $r_{ask-suc} = -0.1$, $r_{\text{ask-fail}} = -0.1$, and $r_{\text{quit}} = -0.3$. The experience replay buffer size is 50,000, mini-batch size is 128, learning rate is 1e-4 with L2 regularization of 1e-6, optimized using Adam.

5.2.5 Evaluation Metrics. We evaluate performance using success rate (SR@T) and average turns (AT). SR@T measures the percentage of successful recommendations within T turns; higher is better. AT measures the average conversation length; lower indicates greater efficiency.

We use hierarchical normalized discounted cumulative gain (hDCG@(T, K)) to evaluate the ranking of the top-K recommendations within T turns. hDCG assigns higher scores to recommendations that are more relevant to the user. A higher nDCG@(T, K) indicates a better ranking performance.

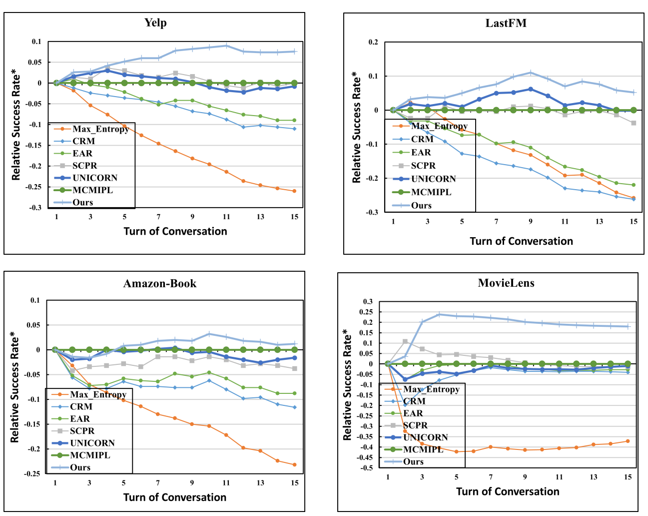


Figure 3: SR* of compared methods at different turns on four datasets (RQ1)

5.3 Performance comparison of AVPPL with existing models (RQ1)

Table 2 reports the SR@15, AT and hDCG@(15, 10) for AVPPL and baseline models. AVPPL achieved significantly higher scores on all metrics and datasets, demonstrating its effectiveness in the VPMCR scenario. The performance gap was largest on MovieLens, likely because movie recommendations are a relatively simple task and AVPPL better models user preferences for items.

Fig. 3 shows the relative success rate (SR*) of each model at every turn compared to the MCMIPL baseline (represented by the dark green line at $y = 0$). Observing the variation trend of curves in Fig. 3, we have the following findings:

- AVPPL almost consistently and substantially surpassed all baselines over the entire conversation session across datasets. Specifically, AVPPL achieved a high success rate in the first a few turns on MovieLens, demonstrating its ability to precisely capture users' preferences.
- As the conversation continues, the performance gap between AVPPL and other baselines widened, especially compared to Max Entropy. The lack of an adaptive policy caused Max Entropy to require excessive turns, while AVPPL dynamically predicts the best action via personalized policies learned through RL.
- Reinforcement learning-based methods like CRM and EAR lag behind more advanced models, as they directly apply RL to a large decision space without effectively representing the conversation state, hindering optimal policy learning. In contrast, graph-based models like SCPR, UNICORN and MCMIPL achieve state-of-the-art performance on some datasets, but underperform AVPPL.

5.4 Evaluating Key Design in AVPPL (RQ2)

5.4.1 Key Components of AVPPL. We examine the effectiveness of Ambiguity-aware Soft Estimation (ASE), our framework's main design, in guiding conversations and adapting to user preference changes in VPMCR scenarios. We separately remove the ASE module for items and attributes (Section 4.1) and replace them with a preference-based scoring strategy [7, 37], which models user preferences using historical click or non-click attributes as mixed signals.

Table 3 rows (a-b) display the ablation study results. Removing the ASE module for both items and attributes significantly degrades performance across all datasets, emphasizing the importance of considering user preference vagueness. The ASE module allows our model to learn a sophisticated conversational state representation and prune a more reasonable action space for the Dynamism-aware Policy Learning (DPL) module, enhancing the upper bound for unified policy learning.

We also find that the ASE component is more effective in measuring user preferences for items than attributes in VPMCR scenarios, suggesting that click behavior provides more direct item-related information.

5.4.2 Key Components of ASE. Table 3 rows (c-e) present the ablation experiments for the ASE component. Row (c) shows that personalized information for user modeling is crucial; without it, the model cannot capture personalized preferences, severely limiting performance. Removing the average preference in Equation 3 (Row (d)) degrades performance across all datasets, with LastFM suffering the most. This may be due to LastFM's numerous attributes and

Table 2: Performance comparison of different models in VPMCR scenario. hDCG stands for hDCG@(15, 10).

Models	Yelp			LastFM			Amazon-Book			MovieLens		
	SR@15	AT	hDCG	SR@15	AT	hDCG	SR@15	AT	hDCG	SR@15	AT	hDCG
Max Entropy	0.062	14.44	0.030	0.376	11.25	0.189	0.180	12.91	0.107	0.448	9.93	0.315
CRM	0.212	13.27	0.070	0.372	12.26	0.126	0.296	12.34	0.109	0.780	5.96	0.341
EAR	0.232	13.05	0.080	0.414	11.61	0.146	0.324	12.14	0.119	0.792	5.50	0.361
SCPR	0.322	12.34	0.115	0.596	10.18	0.206	0.374	11.62	0.139	0.806	4.90	0.387
UNICORN	0.314	12.11	0.140	0.632	9.17	0.280	0.396	11.05	0.193	0.810	4.81	0.548
MCMIPL	0.322	12.16	0.136	0.634	9.52	0.267	0.412	10.90	0.205	0.820	4.39	0.579
AVPPL	**0.398**	**11.26**	**0.175**	**0.686**	**8.58**	**0.306**	**0.424**	**10.75**	**0.206**	**1.000**	**1.60**	**0.689**

Table 3: Ablation study of AVPPL in VPMCR (top) and comparison of AVPPL with other baselines in MIMCR (bottom).

	Yelp			LastFM			Amazon-Book			MovieLens		
	SR@15	AT	hDCG	SR@15	AT	hDCG	SR@15	AT	hDCG	SR@15	AT	hDCG
AVPPL - (VPMCR)	**0.398**	**11.26**	**0.175**	**0.686**	**8.58**	**0.306**	**0.424**	**10.75**	**0.206**	**1.000**	1.60	0.689
(a) - w/o ASE Item.Score	0.328	12.04	0.144	0.618	9.35	0.271	0.386	11.17	0.189	0.852	3.84	0.593
(b) - w/o ASE Attr.Score	0.354	11.88	0.149	0.614	9.44	0.267	0.412	10.91	0.199	1.000	1.75	0.663
(c) - w/o Personalized Preference	0.142	13.84	0.060	0.444	10.79	0.211	0.284	12.10	0.142	0.858	5.22	0.492
(d) - w/o Average Preference	0.368	11.38	0.169	0.630	9.24	0.269	0.416	10.84	0.199	1.000	1.77	0.668
(e) - w/o Decaying Preference	0.382	11.56	0.163	0.628	9.15	0.280	0.410	11.05	0.190	1.000	**1.49**	**0.708**
AVPPL - (MIMCR)	**0.636**	**10.68**	**0.210**	0.840	7.33	**0.350**	**0.610**	**9.81**	**0.251**	**0.988**	**2.42**	**0.640**
MCMIPL - (MIMCR)	0.552	10.95	0.204	**0.856**	**7.21**	0.342	0.544	10.32	0.239	0.838	4.23	0.602
UNICORN - (MIMCR)	0.454	11.01	0.188	0.832	7.42	0.350	0.530	10.23	0.231	0.832	4.35	0.567
SCPR - (MIMCR)	0.452	12.52	0.136	0.688	10.27	0.220	0.450	11.10	0.167	0.834	4.80	0.392

the significant impact of non-displayed attribute information on user preference estimation. Additionally, we remove the historical decay preference in time-aware preference decay (Row (e)), leading to performance degradation on three datasets except for MovieLens. On MovieLens, ASE without decaying information reliably estimates preferences in the current turn, and recommendations succeed within 1-2 rounds. Thus, introducing historical decay preference in short interactive rounds may weaken preference inference on MovieLens.

Overall, the results confirm the ASE module's importance and the proposed AVPPL framework's effectiveness.

5.4.3 VPMCR vs. MIMCR Scenarios. To comprehensively evaluate AVPPL's effectiveness in modeling user preferences based on click behaviors, we relax the scenario assumption and employ the MIMCR scenario involving multi-choice question interactions. In MIMCR, user feedback signals are treated as strong indicators to filter items.

Table 3 compares AVPPL's performance with advanced baselines in the MIMCR scenario. Our method shows significant advantages on Yelp, Amazon-book, and Movielens datasets. On LastFM, although slightly inferior to MCMIPL in SR and AT, AVPPL outperforms all w.r.t. hDCG. These results confirm AVPPL's effectiveness in eliciting user preferences in multi-choice question scenarios, demonstrating its universality and effectiveness in handling both VPMCR and MIMCR scenarios.

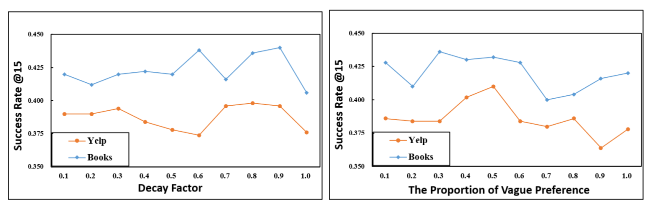


Figure 4: Comparative performance analysis of Success Rate with varying decay factor (left) and proportion of vague preference (right) hyperparameters.(RQ3).

5.5 Model Parameter Analysis RQ3

Table 4: The impact of the coefficient of information intensity w.r.t. SR@15.

Dataset		Yelp			Amazon-Book		
λ_2		0.01	0.1	1	0.01	0.1	1
λ_1	0.01	**0.414**	0.408	0.328	0.424	**0.430**	0.400
	0.1	0.398	0.410	0.344	0.424	0.414	0.384
	1	0.394	0.370	0.302	0.420	0.398	0.406

The previous work on graph-based policy learning [7], has conducted relevant hyperparameter analysis regarding policy learning. Here we focus on the analysis of the hyperparameter impact of the core module (ASE) in AVPPL in the VPMCR scenario. Due to the limited space, we only present results for Yelp and Amazon-Book, but note that LastFM and Movielens exhibit similar trends.

5.5.1 Hyperparameter Analysis in ASE. We identified two key hyperparameters: (1) The information intensity coefficients λ_1 and λ_2 control the importance of explicit versus implicit preferences. The results presented in Table 4 show that larger λ_1 and smaller λ_2 resulted in higher success rates, indicating that explicit preferences (λ_1) are more crucial than implicit preferences (λ_2) in VPMCR. Notably, performance decreases when both λ_1 and λ_2 are large, especially for sparser datasets like Yelp, posing a challenge to the model's robustness. (2) The decay factor γ controls the trade-off between recent and historical preferences. Fig. 4 shows that a moderate decay factor (0.6-0.8) performs best, suggesting that a balance between recent and historical preferences is optimal. Extreme values (0.1 and 1.0) perform poorly, indicating that disregarding historical preferences or solely relying on recent ones is suboptimal.

5.5.2 Proportion of Vague Preferences. We conducted experiments with varying vague preference proportions (0.1 to 1).

In Fig. 4, higher success rates occurred at moderate vague preference proportions. With a moderate level of vague preferences (around 40-50%), the model balances the ability to utilize both vague and explicit preferences, resulting in better recommendations. However, when vague preferences dominated (over 70-80%), the model struggled to accurately determine user needs, hampering performance.

6 CONCLUSION

We propose a realistic Vague Preference Multi-round Conversational Recommendation (VPMCR), which considers the user's vague and volatile preferences. By addressing the limitations of existing CRS scenarios and incorporating the VPMCR scenario and AVPPL solution, we aim to improve the overall performance and applicability of CRS in real-world settings, particularly for users with vague or dynamic preferences. We hope the findings will provide valuable insights into developing user-centric CRSs that can handle users' vague and dynamic preferences. In future work, we plan to explore more sophisticated vague preference modeling and more efficient policy learning techniques to further enhance the performance and generalizability of AVPPL in VPMCR.

REFERENCES

[1] M Mehdi Afsar, Trafford Crump, and Behrouz Far. 2022. Reinforcement Learning based Recommender Systems: A Survey. *Comput. Surveys* 55, 7 (2022), 1–38.

[2] James R Bettman, Mary Frances Luce, and John W Payne. 1998. Constructive consumer choice processes. *Journal of consumer research* 25, 3 (1998), 187–217.

[3] Senthilkumar Chandramohan, Matthieu Geist, Fabrice Lefevre, and Olivier Pietquin. 2011. User simulation in dialogue systems using inverse reinforcement learning. In *Interspeech 2011*. 1025–1028.

[4] Qibin Chen, Junyang Lin, Yichang Zhang, Ming Ding, Yukuo Cen, Hongxia Yang, and Jie Tang. 2019. Towards Knowledge-Based Recommender Dialog System. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, Hong Kong, China, 1803–1813. https://doi.org/10.18653/v1/D19-1189

[5] Yankai Chen, Huifeng Guo, Yingxue Zhang, Chen Ma, Ruiming Tang, Jingjie Li, and Irwin King. 2022. Learning Binarized Graph Representations with Multi-Faceted Quantization Reinforcement for Top-K Recommendation. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '22)*. Association for Computing Machinery, New York, NY, USA, 168–178. https://doi.org/10.1145/3534678.3539452

[6] Romain Deffayet, Thibaut Thonet, Jean-Michel Renders, and Maarten de Rijke. 2023. Offline Evaluation for Reinforcement Learning-based Recommendation: A Critical Issue and Some Alternatives. *arXiv preprint arXiv:2301.00993* (2023).

[7] Yang Deng, Yaliang Li, Fei Sun, Bolin Ding, and Wai Lam. 2021. Unified conversational recommendation policy learning via graph-based reinforcement learning. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1431–1441.

[8] Chongming Gao, Kexin Huang, Jiawei Chen, Yuan Zhang, Biao Li, Peng Jiang, Shiqi Wang, Zhong Zhang, and Xiangnan He. 2023. Alleviating Matthew Effect of Offline Reinforcement Learning in Interactive Recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '23)*. 11. https://doi.org/10.1145/3539618.3591636

[9] Chongming Gao, Wenqiang Lei, Jiawei Chen, Shiqi Wang, Xiangnan He, Shijun Li, Biao Li, Yuan Zhang, and Peng Jiang. 2022. CIRS: Bursting Filter Bubbles by Counterfactual Interactive Recommender System. *arXiv preprint arXiv:2204.01266* (2022).

[10] Chongming Gao, Wenqiang Lei, Xiangnan He, Maarten de Rijke, and Tat-Seng Chua. 2021. Advances and Challenges in Conversational Recommender Systems: A Survey. *AI Open* 2 (2021), 100–126. https://doi.org/10.1016/j.aiopen.2021.06.002

[11] Chongming Gao, Shijun Li, Wenqiang Lei, Jiawei Chen, Biao Li, Peng Jiang, Xiangnan He, Jiaxin Mao, and Tat-Seng Chua. 2022. KuaiRec: A Fully-observed Dataset and Insights for Evaluating Recommender Systems. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management (CIKM '22)*. 11.

[12] Wei Guo, Rong Su, Renhao Tan, Huifeng Guo, Yingxue Zhang, Zhirong Liu, Ruiming Tang, and Xiuqiang He. 2021. Dual Graph Enhanced Embedding Neural Network for CTR Prediction. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining (KDD '21)*. Association for Computing Machinery, New York, NY, USA, 496–504. https://doi.org/10.1145/3447548.3467384

[13] Xu Han, Shulin Cao, Xin Lv, Yankai Lin, Zhiyuan Liu, Maosong Sun, and Juanzi Li. 2018. Openke: An open toolkit for knowledge embedding. In *Proceedings of the 2018 conference on empirical methods in natural language processing: system demonstrations*. 139–144.

[14] Zhankui He, Handong Zhao, Tong Yu, Sungchul Kim, Fan Du, and Julian McAuley. 2022. Bundle MCR: Towards Conversational Bundle Recommendation. In *Proceedings of the 16th ACM Conference on Recommender Systems (RecSys '22)*. Association for Computing Machinery, New York, NY, USA, 288–298. https://doi.org/10.1145/3523227.3546755

[15] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).

[16] Wenqiang Lei, Xiangnan He, Yisong Miao, Qingyun Wu, Richang Hong, Min-Yen Kan, and Tat-Seng Chua. 2020. Estimation–Action–Reflection: Towards Deep Interaction Between Conversational and Recommender Systems. In *WSDM*. 304–312.

[17] Wenqiang Lei, Gangyi Zhang, Xiangnan He, Yisong Miao, Xiang Wang, Liang Chen, and Tat-Seng Chua. 2020. Interactive Path Reasoning on Graph for Conversational Recommendation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '20)*. 2073–2083.

[18] Raymond Li, Samira Ebrahimi Kahou, Hannes Schulz, Vincent Michalski, Laurent Charlin, and Chris Pal. 2018. Towards Deep Conversational Recommendations. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems (NeurIPS '18)*. 9748–9758.

[19] Shijun Li, Wenqiang Lei, Qingyun Wu, Xiangnan He, Peng Jiang, and Tat-Seng Chua. 2021. Seamlessly Unifying Attributes and Items: Conversational Recommendation for Cold-Start Users. *ACM Trans. Inf. Syst.* 39, 4, Article 40 (aug 2021), 29 pages. https://doi.org/10.1145/3446427

[20] Allen Lin, Ziwei Zhu, Jianling Wang, and James Caverlee. 2023. Enhancing User Personalization in Conversational Recommenders. *arXiv preprint arXiv:2302.06656* (2023).

[21] Dugang Liu, Mingkai He, Jinwei Luo, Jiangxu Lin, Meng Wang, Xiaolian Zhang, Weike Pan, and Zhong Ming. 2022. User-Event Graph Embedding Learning for Context-Aware Recommendation. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '22)*. Association for Computing Machinery, New York, NY, USA, 1051–1059. https://doi.org/10.1145/3534678.3539458

[22] Ali Montazeralghaem and James Allan. 2022. Learning Relevant Questions for Conversational Product Search Using Deep Reinforcement Learning. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining (WSDM '22)*. Association for Computing Machinery, New York, NY, USA, 746–754. https://doi.org/10.1145/3488560.3498526

[23] Seungwhan Moon, Pararth Shah, Anuj Kumar, and Rajen Subba. 2019. OpenDialKG: Explainable Conversational Reasoning with Attention-based Walks over Knowledge Graphs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL '19)*. 845–854.

[24] Zhaochun Ren, Zhi Tian, Dongdong Li, Pengjie Ren, Liu Yang, Xin Xin, Huasheng Liang, Maarten de Rijke, and Zhumin Chen. 2022. Variational Reasoning about User Preferences for Conversational Recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information*

Retrieval (SIGIR '22). Association for Computing Machinery, New York, NY, USA, 165–175. https://doi.org/10.1145/3477495.3532077
- [25] Soheil Sadeghi Eshkevari, Xiaocheng Tang, Zhiwei Qin, Jinhan Mei, Cheng Zhang, Qianying Meng, and Jia Xu. 2022. Reinforcement Learning in the Wild: Scalable RL Dispatching Algorithm Deployed in Ridehailing Marketplace. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '22)*. Association for Computing Machinery, New York, NY, USA, 3838–3848. https://doi.org/10.1145/3534678.3539095
- [26] Yueming Sun and Yi Zhang. 2018. Conversational Recommender System. In *SIGIR*. 235–244.
- [27] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).
- [28] Shiqi Wang, Chongming Gao, Min Gao, Junliang Yu, Zongwei Wang, and Hongzhi Yin. 2022. Who Are the Best Adopters? User Selection Model for Free Trial Item Promotion. *IEEE Transactions on Big Data* (2022).
- [29] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019. Kgat: Knowledge graph attention network for recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 950–958.
- [30] Xiaolei Wang, Kun Zhou, Ji-Rong Wen, and Wayne Xin Zhao. 2022. Towards Unified Conversational Recommender Systems via Knowledge-Enhanced Prompt Learning. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '22)*. Association for Computing Machinery, New York, NY, USA, 1929–1937. https://doi.org/10.1145/3534678.3539382
- [31] Wenquan Wu, Zhen Guo, Xiangyang Zhou, Hua Wu, Xiyuan Zhang, Rongzhong Lian, and Haifeng Wang. 2019. Proactive human-machine conversation with explicit conversation goals. *arXiv preprint arXiv:1906.05572* (2019).
- [32] Yuxia Wu, Lizi Liao, Gangyi Zhang, Wenqiang Lei, Guoshuai Zhao, Xueming Qian, and Tat-Seng Chua. 2022. State graph reasoning for multimodal conversational recommendation. *IEEE Transactions on Multimedia* (2022).
- [33] Hu Xu, Seungwhan Moon, Honglei Liu, Bing Liu, Pararth Shah, Bing Liu, and Philip Yu. 2020. User Memory Reasoning for Conversational Recommendation. In *Proceedings of the 28th International Conference on Computational Linguistics (COLING '20)*. 5288–5308.
- [34] Kerui Xu, Jingxuan Yang, Jun Xu, Sheng Gao, Jun Guo, and Ji-Rong Wen. 2021. Adapting User Preference to Online Feedback in Multi-Round Conversational Recommendation. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining (WSDM '21)*. 364–372.
- [35] Wanqi Xue, Qingpeng Cai, Ruohan Zhan, Dong Zheng, Peng Jiang, and Bo An. 2023. ResAct: Reinforcing Long-term Engagement in Sequential Recommendation with Residual Actor. In *International Conference on Learning Representations (ICLR '23)*.
- [36] Qihua Zhang, Junning Liu, Yuzhuo Dai, Yiyan Qi, Yifan Yuan, Kunlun Zheng, Fan Huang, and Xianfeng Tan. 2022. Multi-Task Fusion via Reinforcement Learning for Long-Term User Satisfaction in Recommender Systems. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '22)*. Association for Computing Machinery, New York, NY, USA, 4510–4520. https://doi.org/10.1145/3534678.3539040
- [37] Yiming Zhang, Lingfei Wu, Qi Shen, Yitong Pang, Zhihua Wei, Fangli Xu, Bo Long, and Jian Pei. 2022. Multiple Choice Questions based Multi-Interest Policy Learning for Conversational Recommendation. In *Proceedings of the ACM Web Conference 2022*. 2153–2162.
- [38] Kun Zhou, Wayne Xin Zhao, Shuqing Bian, Yuanhang Zhou, Ji-Rong Wen, and Jingsong Yu. 2020. Improving Conversational Recommender Systems via Knowledge Graph based Semantic Fusion. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (SIGKDD' 20)*. 1006–1014.
- [39] Kun Zhou, Yuanhang Zhou, Wayne Xin Zhao, Xiaoke Wang, and Ji-Rong Wen. 2020. Towards Topic-Guided Conversational Recommender System. In *Proceedings of the 28th International Conference on Computational Linguistics (COLING '2020)*.
- [40] Sijin Zhou, Xinyi Dai, Haokun Chen, Weinan Zhang, Kan Ren, Ruiming Tang, Xiuqiang He, and Yong Yu. 2020. Interactive Recommender System via Knowledge Graph-Enhanced Reinforcement Learning. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR' 20)*. 179–188.
- [41] Yuanhang Zhou, Kun Zhou, Wayne Xin Zhao, Cheng Wang, Peng Jiang, and He Hu. 2022. C^2-CRS: Coarse-to-Fine Contrastive Learning for Conversational Recommender System. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining (WSDM '22)*. Association for Computing Machinery, New York, NY, USA, 1488–1496. https://doi.org/10.1145/3488560.3498514
- [42] Jie Zou, Yifan Chen, and Evangelos Kanoulas. 2020. Towards Question-Based Recommender Systems. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20)*. 881–890.