Variational Reasoning about User Preferences for Conversational Recommendation

Zhaochun Ren[1†*] Zhi Tian[1*] Dongdong Li[1] Pengjie Ren[1] Liu Yang[1]
Xin Xin[1] Huasheng Liang[2] Maarten de Rijke[3] Zhumin Chen[1†]
[1]Shandong University, Qingdao, China
[2]WeChat, Tencent, Shenzhen, China
[3]University of Amsterdam, Amsterdam, The Netherlands
{zhaochun.ren,xinxin,chenzhumin}@sdu.edu.cn,zhi_tian@mail.sdu.edu.cn,{lddsdu,yangliushirry}@gmail.com
jay.ren@outlook.com,watsonliang@tencent.com,m.derijke@uva.nl

ABSTRACT

Conversational recommender systems (CRSs) provide recommendations through interactive conversations. CRSs typically provide recommendations through relatively straightforward interactions, where the system continuously inquires about a user's explicit attribute-aware preferences and then decides which items to recommend. In addition, topic tracking is often used to provide naturally sounding responses. However, merely tracking topics is not enough to recognize a user's real preferences in a dialogue.

In this paper, we address the problem of accurately recognizing and maintaining user preferences in CRSs. Three challenges come with this problem: (1) An ongoing dialogue only provides the user's short-term feedback; (2) Annotations of user preferences are not available; and (3) There may be complex semantic correlations among items that feature in a dialogue. We tackle these challenges by proposing an end-to-end variational reasoning approach to the task of conversational recommendation. We model both long-term preferences and short-term preferences as latent variables with topical priors for explicit *long-term* and *short-term preference exploration*, respectively. We use an efficient stochastic gradient variational Bayesian (SGVB) estimator for optimizing the derived evidence lower bound. A policy network is then used to predict topics for a clarification utterance or items for a recommendation response. The use of explicit sequences of preferences with multi-hop reasoning in a heterogeneous knowledge graph helps to provide more accurate conversational recommendation results.

Extensive experiments conducted on two benchmark datasets show that our proposed method outperforms state-of-the-art baselines in terms of both objective and subjective evaluation metrics.

CCS CONCEPTS

• **Information systems** → *Recommender systems*; *Users and interactive retrieval*.

[*]Equal contribution.
[†]Corresponding author.

SIGIR '22, July 11–15, 2022, Madrid, Spain

ACM ISBN 978-1-4503-8732-3/22/07…$15.00
https://doi.org/10.1145/3477495.3532077

KEYWORDS

Conversational recommendation, Variational inference, User preference tracking, Task-oriented dialogue systems

ACM Reference Format:
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), July 11–15, 2022, Madrid, Spain.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3477495.3532077

1 INTRODUCTION

The task of a conversational recommender system (CRS) is to provide recommendations to users through conversational interactions. Interest in conversational recommendations (CRs) is rising. Algorithmic approaches to CRs can be divided into *attribute-aware* and *topic-guided*. The former kind accomplishes a specific recommendation goal in a multi-turn conversation scenario by inquiring about the user's preferred attributes [6, 19, 22, 34, 49]. In Figure 1(a) we see an example of an attribute-aware CRS that continually asks attribute-aware questions, while the user only needs to answer "yes/no" to let the system understand the user's explicit preferences. Attribute-aware CRSs focus on when and what to ask before deciding about the item(s) to be recommended.

Topic-guided approaches to CRs are usually integrated with task-oriented dialogue systems (TDSs). Several recent studies on CRSs focus on providing responses through naturally sounding conversations, where the user's preference is implicitly reflected by their utterances. To help the system comprehend complicated dialogue interactions, so-called dialogue topics, i.e., sets of keywords, are introduced to guide the conversational recommender system (CRS) model to output responses [28, 31, 35, 55, 56]. Early studies on topic-guided approaches to CRSs extract topics from each utterance independently. Later work applies *topic threads* to guide the conversation [see, e.g., 57]. In Figure 1(b) we show an example of a topic-guided CRSs as well as the dialogue topics.

Preferences. Topics are not enough to fully capture user preferences in CRSs as they only reflect part of the information shared in a conversation. In Figure 1(b) we put the topics and user preference side-by-side and find inconsistencies between the two. How can we capture user preferences more accurately so as to provide accurate recommendation results? It is customary to distinguish between a user's long-term preferences and their short-term preferences [12,

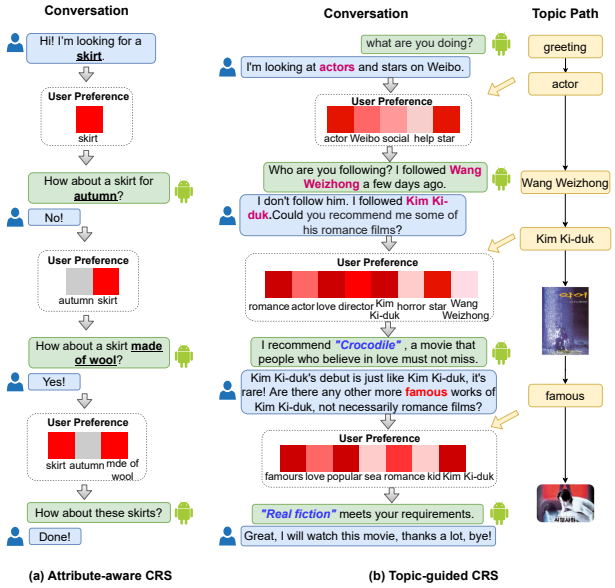

Figure 1: Examples from two kinds of CRSs: attribute-aware CRSs (left) and topic-guided CRSs (right).

17, 26, 41]. Long-term preferences refer to the user's long-term tastes and interests, e.g., items often clicked or purchased, whereas short-term preferences are reflected by the user's short-term behavior, e.g., clicks or feedback in the current search or recommendation session. In a topic-guided CRS, as topics are extracted to guide the whole conversation, we assume that both long-term preferences and short-term preferences can be captured by lists of topics.

Challenges. The development of preference-aware CRS solutions faces a number of challenges: (1) It is hard to acquire accurate user preferences just through a user's utterances in an ongoing dialogue. To the best of our knowledge, there is no CR approach that simultaneously models the user long-term preference and short-term preference. (2) In a CRS, it is difficult to annotate long-term and short-term preferences due to privacy concerns and substantial costs. (3) Existing approaches to CRS have a limited semantic understanding of items and topics, which makes it hard to generate knowledgeable responses in a recommendation context.

Our proposal. We propose a method for jointly modeling long-term and short-term preferences in CR, *user preference conversational recommender* (UPCR). UPCR consists of a *user preference explorer* component and a *policy network* component to detect user preferences and perform recommendations, respectively. By jointly inferring long-term preferences from historical conversations and short-term preferences from the current conversation, UPCR performs variational reasoning to acquire accurate user preferences.

To tackle the challenge of limited annotation, UPCR considers the user's long-term and short-term preference as dual latent variables inferred in a variational Bayesian manner. We employ a stochastic gradient variational Bayesian (SGVB) estimator to efficiently approximate the exact posterior distribution of preferences via two approximate inference processes simultaneously. The policy network component decodes topics for clarification questions and recommendations as preferred items.

We tackle the challenge of limited semantic understanding by means of external knowledge. We infer explicit topics and recommendations jointly through user preference exploration and knowledge path reasoning.

We conduct experiments on two benchmark datasets for conversational recommendation and find that UPCR significantly outperforms state-of-the-art CRS baselines.

Our contributions. To sum up, our contributions are as follows:

- To the best of our knowledge, we are the first to explore user preferences in topic-guided CRSs.
- We propose a method, UPCR, to address challenges about long-term and short-term preferences modeling.
- UPCR performs variational reasoning for user preference modeling, where long-term preferences and short-term preferences are inferred in a variational Bayesian manner.
- UPCR generates recommendations by integrating user preferences with external knowledge.
- Experiments show that UPCR outperforms state-of-the-art baselines on CRS.

2 RELATED WORK

In this section, we discuss related work on conversational recommendation, attribute-aware CRSs, and topic-guided CRSs.

Conversational recommendation. Building on advances in interactive recommendation [4, 25, 45, 61], conversational recommendation has been proposed to address the task of recommendation through a conversation between a system and a user [2, 40]. Early studies on CRSs formulate the task as a specific application of task-oriented multi-turn dialogue systems (TDS) [8, 16, 47, 53]. Two main types of conversational recommender systems (CRSs) have been studied: attribute-aware and topic-guided.

Attribute-aware CRSs. Attribute-aware CRSs focus on the recommendation strategy in CRSs, including "whether to ask or recommend", "which attributes to ask" or "which items to recommend." Early work tends to obtain user preferences based on asking about items directly [5, 44, 46, 54, 62], or asking attributes through a heuristic method [4, 29, 39, 52]. Several strategies for attribute-aware CRSs ask a fixed number of questions and make a recommendation at the last turn [18, 19, 59], whereas others automatically decide on an appropriate time to recommend instead of continuing to ask questions. Reinforcement learning strategies are widely applied to attribute-aware CRSs. Christakopoulou et al. [5], Li et al. [22] and Zhang et al. [51] focus on cold-start users in conversational recommendation and extend bandit-based algorithms to balance the trade-off between exploration and exploitation.

Recent studies focus on enabling CRS agents to automatically decide an appropriate time to recommend instead of continuing to ask questions [6, 18, 19, 39, 55]. Deng et al. [6] and Lei et al. [19] use a knowledge graph derived from external data sources to improve the recommendation performance. Attribute-based CRS methods tend to consider simple and clear replies, and neglect complicated user-system interactions in conversations [30].

Topic-guided CRSs. Topic-guided CRSs focus on interacting with users through natural language conversations, emphasizing fluent response generation and precise recommendations [3, 21, 27, 31,

Table 1: Glossary.

Symbol	Description
$u, \mathcal{U}$	a user and collection of all users
$i, \mathcal{I}$	a item and collection of all items
$k, \mathcal{K}$	a topic and collections of topics
tp	topic path
w	a word
$\mathcal{V}$	vocabulary
T	max conversation turn
X	the input of encoder
s	an utterance
C	a conversation
m	short-term user preference
l	long-term user preference
A	the action
θ	parameters in the model
θ_a	parameters in *Policy Network*
θ_g	parameters in *Response Generator*
$p_{\theta_l}(l)$	the prior distribution of l
$q_{\phi_l}(l)$	the posterior distribution of l
$p_{\theta_m}(m)$	the prior distribution of m
$q_{\phi_m}(m)$	the posterior distribution of m
$\mathcal{G}$	knowledge graph
d	dimension of embedding and hidden vector
H	a hidden state generated in a transformer

42, 55–58, 60]. Unlike attribute-aware CRSs, topic-guided CRSs focus on making recommendations using free text, which creates considerable flexibility to influence how a dialogue continues. In the context of topic-guided CRSs, external knowledge has been used to enhance the dialogue semantics [31] or update the user representation [3, 55, 56].

Chen et al. [3] integrate a recommendation system and a dialogue system via an end-to-end framework to bridge the gap between the two systems. Liao et al. [24] utilize a pointer network to incorporate a graph convolution network-based recommendation method and global task control in response generation. Li et al. [21] utilize an *autoencoder* [37] for recommendation and a hierarchical RNN for response generation. Liu et al. [27] propose a multi-goal driven conversation generation framework. It utilizes a goal planning module and a goal-guided response module to proactively lead a conversation from chit-chat to recommendation. Zhou et al. [57] propose a topic-guided CRS method that incorporates topic threads to enforce transitions actively towards a final recommendation using a combination of a sequential recommender and a response generator. Chen et al. [3] utilize knowledge graphs (KGs) to enhance the semantics of contextual items for recommendation. Zhou et al. [56] incorporate both word-oriented and entity-oriented KGs and bridge the semantic gap between the two KGs to enhance the user representations. Finally, Ma et al. [31] perform tree-structured reasoning on a knowledge graph, for recommendation and generation.

In contrast with existing topic-guided CRS approaches, our proposed method targets exploring the user preferences in a CRS, which has not yet been addressed by previous studies.

3 METHOD

3.1 Overview

In this section, we detail the user preference conversational recommender (UPCR). Before we detail each component of the method, we first give an overview. We first introduce the main concepts and formulate our research problem and UPCR in Section 3.2. Since almost each component in UPCR has an encoder-decoder architecture, we describe our encoder and decoder structure in Section 3.3.

As illustrated in Fig. 3, training UPCR comprises four processes: (1) In the *input representation* component, a user encoder, a topic path encoder, and a context encoder encode the input information into hidden representations (see Section 3.4). (2) The *preference explorer* component contains long-term and short-term preference explorers to track user preferences (see Section 3.5). (3) Based on the user preferences, a *policy network* component generates topics for clarification questions or items for recommendation (see Section 3.6). (4) Finally, a *response generator* generates a response in natural language (see Section 3.7).

3.2 Problem formulation

Table 1 lists the notation used in this paper.

We assume that users u are taken from a set $\mathcal{U}$ and that items i are taken from a set $\mathcal{I}$. Words w are taken from a vocabulary $\mathcal{V}$. Following [57], we define a topic k to be a tag that can be linked to external knowledge (e.g., DBpedia [1] or ConceptNet [38]). We refer to a conversation C with T turns as a list of utterances, i.e., $C = \{s_j\}_{j=1}^{T}$, where utterance s_j at the j-th turn is composed of a sequence of words, i.e., $s = \{w_r\}_{r=1}^{n}$, $w_r \in \mathcal{V}$. Given C, we define a *topic path* tp to be a sequence of topics, i.e., $tp = \{k_j\}_{j=1}^{T}$, where k_j refers to the topics discussed at the j-th turn. The *conversational context* at the t-th turn is written as $C_t = \{s_j\}_{j=1}^{t-1}$, with a topic path $tp_t = \{k_j\}_{j=1}^{t-1}$.

Conversational recommendation. We assume that a conversational recommender system (CRS) consists of three main stages: preference tracking, policy management, and response generation. At the j-th turn, given conversatonal context C_j and topic path tp_j, a CRS explores user preferences via the preference tracking stage. Then, the CRS generates an action A_j according to the explored user preference, where A_j consists of a set of topics or recommended items. If the action refers to topics, the recommender raises a clarification question or a chit-chat response, otherwise the action results in a response with recommended results. Given A_j, the CRS generates a response s_j to reply to the user.

We consider UPCR as a model with parameters θ. Given a user u, conversational context C_j, and corresponding topic path tp_j, we aim to maximize the probability distribution $P_\theta(A_j, s_j | u, C_j, tp_j)$ in UPCR to infer the target function y^* as follows:

$$y^* = \prod_{j=1}^{T} P_\theta(A_j, s_j | u, C_j, tp_j). \tag{1}$$

We divide the user preferences into two parts: *long-term preferences* and *short-term preferences*. Given n dialogues, we define a text span l (i.e., a sequence of words) to be the user's long-term preference reflected in historical $n-1$ dialogues $\{C^{u_1}, C^{u_2}, \ldots, C^{u_{n-1}}\}$. For the

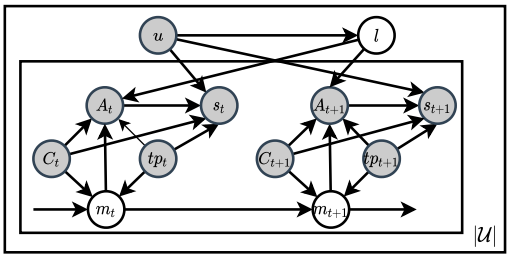

Figure 2: The graphical representation of UPCR. Shaded nodes represent observed variables.

current n-th dialogue, we define a text span m_j as the user's short-term preference at the j-th turn. The short-term user preference m_j is predicted given conversational context C_j and its corresponding topic path tp_j. To infer user preferences l and m_j, we formulate UPCR as a variational Bayesian generative model.

Variational Bayesian generative model. The graphical representation of UPCR is shown in Figure 2. As annotations for long-term and short-term user preferences are impractical, we regard l and m_j as two latent variables within a Bayesian generative model, so we formulate y^* as follows:

$$y^* = \prod_{j=1}^{T} P_{\theta_g}(s_j|u, C_j, tp_j, A_j) \cdot \prod_{j=1}^{T} \sum_{l,m_j} P_{\theta_a}(A_j|u, C_j, tp_j, l, m_j) \cdot P_{\theta_l}(l) \cdot P_{\theta_m}(m_j), \tag{2}$$

where $P_{\theta_g}(s_j|u, C_j, tp_j, A_j)$ is derived using a response generator, $P_{\theta_a}(A_j|u, C_j, tp_j, l, m_j)$ is derived using a policy network, $P_{\theta_l}(l)$ and $P_{\theta_m}(m_j)$ are estimated through a long-term preference explorer and a short-term preference explorer, respectively. At the t-th turn, m_t is derived depending on the previous short-term preference m_{t-1}, context C_t and topic path tp_t. So we define $P_{\theta_l}(l)$ and $P_{\theta_m}(m_t)$ as:

$$\begin{aligned} p_{\theta_l}(l) &\triangleq p_{\theta_l}(l|u), \\ p_{\theta_m}(m_t) &\triangleq p_{\theta_m}(m_t|m_{t-1}, C_t, tp_t), \end{aligned} \tag{3}$$

where θ_l and θ_m are parameters in the long-term preference explorer and the short-term preference explorer respectively. Then we derive an action A_t through $P_{\theta_a}(A_t|u, C_t, tp_t, l, m_t)$ with parameters θ_a, and draw a response s_t from $P_{\theta_g}(s_t|u, C_t, tp_t, A_t)$ with parameters θ_g.

At the t-th turn, to maximize Eq. 2, we need to estimate the posterior distribution $p_{\theta_{l,m}}(l, m_t|u, C_t, tp_t, A_t)$. However, the exact posterior distribution is intractable due to its complicated posterior expectation estimation. Thus we apply variational inference [15] to approximate the posterior distribution with two inference networks, i.e., $q_{\phi_l}(l)$ and $q_{\phi_m}(m_t)$, respectively:

$$\begin{aligned} q_{\phi_l}(l) &\triangleq q_{\phi_l}(l|u, C^{u_1}, C^{u_2}, \ldots, C^{u_n}) = q_{\phi_l}(l|u, C^{u_n}), \\ q_{\phi_m}(m_t) &\triangleq q_{\phi_m}(m_t|m_{t-1}, C_t, tp_t, A_t), \end{aligned} \tag{4}$$

where $\{C^{u_1}, C^{u_2}, \ldots, C^{u_{n-1}}\}$ are the historical conversations that user u was involved in, C^{u_n} refers to the current conversation. After substituting Eq. 4 into $\prod_{j=1}^{t} p_\theta(A_j|C_j, tp_j, u)$ at the t-th turn, we have the following approximation:

$$\prod_{j=1}^{t-1} \sum_{l,m_j} p_{\theta_a}(A_j|C_j, tp_j, u) q_{\varphi_l}(l) q_{\varphi_m}(m_j) \cdot \sum_{l,m_t} p_{\theta_a}(A_t|C_t, tp_t, u, l, m_t) p_{\theta_l}(l) p_{\theta_m}(m_t). \tag{5}$$

As the previous $t-1$ turns of utterances already exist, we assume the marginal distribution $p_{\theta_a}(A_j|C_j, tp_j, u) = 1$ when $1 \le j < t$. Following the homogeneous Markov hypothesis [33], short-term preference m_t at the t-th turn purely relies on m_{t-1}. Then we infer the evidence lower bound (ELBO) to optimize both prior and posterior networks simultaneously as follows:

$$\begin{aligned} &\log p_\theta(A_t|C_t, tp_j, u) \\ &\ge \mathrm{E}_{q_{\varphi m}(m_{t-1})} \Big[\mathrm{E}_{q_{\varphi m}(m_t)} \mathrm{E}_{q_{\varphi_l}(l)} \log p_{\theta_a}(A_t|C_t, tp_t, u, l, m_t) \\ &\quad - KL(p_{\theta_m}(m_t)||q_{\varphi_m}(m_t)) - KL(p_{\theta_l}(l)||q_{\varphi_l}(l)) \Big] \\ &= -\mathcal{L}_a, \end{aligned} \tag{6}$$

where E is the expectation, and KL is the Kullback-Leibler divergence. To estimate Eq. 6, we first sample m^q_{t-1} from $q_{\phi_m}(m_{t-1})$, which is for inferring $p_{\theta_m}(m_t)$ and $q_{\theta_m}(m_t)$; then we sample the prior short-term preference m^p_t from $p_{\theta_m}(m_t)$ and posterior short-term preference m^q_t from $q_{\theta_m}(m_t)$. We sample the prior long-term preference l^p from $p_{\theta_l}(\mathrm{l})$ and posterior long-term preference l^q from $q_{\phi_l}(\mathrm{l})$. Finally, $p_{\theta_a}(A_t|\cdot)$ generates A_t depending on m^q_t and l^q. After obtaining action A_t, a response generator with parameters θ_g generates response s_t. The above procedure is illustrated in Fig. 3.

3.3 Encoder and decoders

We begin by describing the encoder used in UPCR. We then describe three types of decoder: a decoder with ground truth, one without ground truth, and a decoder with a copy mechanism.

Encoder. We use a transformer encoder [43] as the backbone to encode text sequence, which contains N identical layers. X=[x_1, x_2, …, $x_{|X|}$] denotes a sequence of tokens, where x_r indicates the r-th token and $|X|$ is the length of X. The encoder encodes X into a sequence of hidden vectors H^X as follows:

$$H^X = [h^x_1, h^x_2, \ldots, h^x_{|X|}] = \mathrm{encoder}(X), \tag{7}$$

where h^x_r is the hidden vector of the r-th word x_r, $1 \le r \le |X|$.

Decoder with ground-truth. We employ a transformer decoder [43] with N identical layers as the backbone to generate a sequence. We learn the decoder under a teacher-forcing strategy [10, 48] by feeding it with a shifted ground truth sequence. Given a ground truth word sequence $y_1, y_2, \ldots, y_{|Y|-1}$, we denote the input as $Y = [bos, y_1, y_2, \ldots, y_{|Y|-1}]$, where bos is a special token to begin the sequence. At the ζ-th decoding step, $1 \le \zeta \le |Y|$, given H^X and $Y^s_{1:\zeta-1} = [bos, y_1, y_2, \ldots, y_{\zeta-1}]$, the decoder outputs a hidden vector h^y_ζ of the ζ-th word. After projecting h^y_ζ to the vocabulary space by a multilayer perceptron [9] (mlp), we have:

$$p(y_\zeta) = \mathrm{decoder}^g(H^X, Y^s_{1:\zeta-1}) = \mathrm{softmax}\left(\mathrm{mlp}\left(h^y_\zeta\right)\right), \tag{8}$$

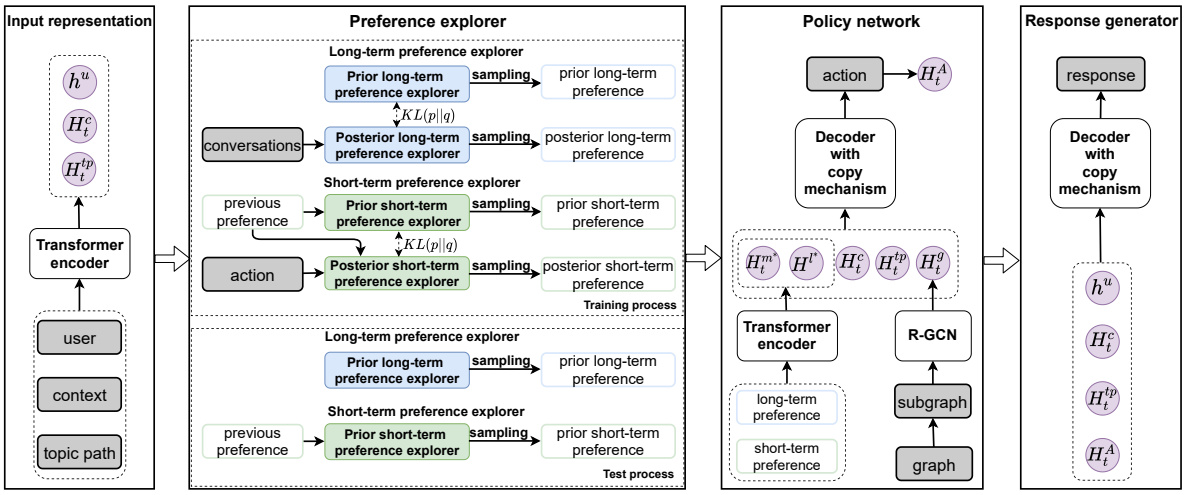

Figure 3: An overview of UPCR. UPCR has four components: an input representation, a preference explorer, a policy network and a response generator.

where $p(y_\zeta)$ is the ζ-th word distribution over vocabulary, $decoder^g$ is denoted to reflect the whole process.

Decoder without ground-truth. A ground-truth sequence Y is not always provided in CRSs stages, e.g., user preference tracking. Following [50], we address this problem by sampling input words via a probabilistic distribution over the preceding output. At the ζ-th decoding step, we denote the input of the decoder as H^X and sampled word sequence $Y^v_{1:\zeta-1} = [bos, y^v_1, y^v_2, \ldots, y^v_{\zeta-1}]$, in which the r-th word y^v_r is sampled from $p^v(y_r)$. After performing the same operation as in Eq. 8, we obtain decoder^v as follows:

$$p^v(y_\zeta) = \text{decoder}^v\left(H^x, Y^v_{1:\zeta-1}\right) = \text{softmax}\left(\text{mlp}\left(h^{y^v}_\zeta\right)\right). \quad (9)$$

Decoder with copy mechanism. In UPCR, we apply a copy mechanism [11] to copy words from context. Formally, given the input X, we get output $H^X = [h^x_1, h^x_2, \ldots, h^x_{|X|}]$ using Eq. 7. At the ζ-th decoding step, the decoder outputs a hidden vector h^y_ζ. The probability of generating the target word y_ζ is the sum of the two probabilities as follows:

$$\begin{aligned} p^g(y_\zeta) &= \frac{1}{Z}\exp\left(\text{mlp}\left(h^y_\zeta\right)\right), p^c(y_\zeta) \\ &= \frac{1}{Z}\sum_{x_r = y_\zeta}\exp\left(\left(h^y_\zeta\right)^\top \cdot h^x_r\right), \\ p^{gc}(y_\zeta) &= p^g(y_\zeta) + p^c(y_\zeta), \end{aligned} \quad (10)$$

where Z is a normalization term shared between the two probabilities. For convenience, we write the above process as:

$$p^{gc}(y_\zeta) = \text{decoder}^{gc}(X, H^X, Y^s_{1:\zeta-1}). \quad (11)$$

3.4 Input representation

We detail various input representations of UPCR: user representation, context representation, and topic path representation. We encode the inputs to representations using Eq. 7. We encode user u into a hidden vector h^u. At the t-th turn, we concatenate utterances in context $C_t = \{s_j\}_{j=1}^{t-1}$ in a chronological order and encode them into H^c_t. The topic path tp_t corresponding to the context is encoded into topic-level hidden representation H^{tp}_t. Note that the parameters of encoders that encode different types of input are not shared. The input representation processes are formulated as:

$$h^u = \text{encoder}(u),\ H^c_t = \text{encoder}(C_t),\ H^{tp}_t = \text{encoder}(tp_t). \quad (12)$$

3.5 Preference explorer

The preference explorer contains two parts, a long-term preference explorer and a short-term preference explorer.

Long-term preference explorer. For the prior long-term preference distribution, we use the user representation h^u to infer the long-term preference. As annotations of user preference are unavailable, we employ a decoder without ground-truth (i.e., Eq. 9) to decode l^p sequentially. Then the prior distribution $p_{\theta_l}(l)$ is estimated as follows:

$$p_{\theta_l}(l) = \prod_{\zeta=1}^{|l|} \text{decoder}^v(h^u, l^p_{1:\zeta-1}), \quad (13)$$

Similarly, we estimate the posterior long-term preference distribution given the conversation C. We encode C into a hidden representation H^c, then we incorporate H^c with h^u to infer the posterior distribution $q_{\phi_l}(l)$, formulated as:

$$q_{\phi_l}(l) = \prod_{\zeta=1}^{|l|} \text{decoder}^v([h^u; H^c], l^q_{1:\zeta-1}), \quad (14)$$

where $[\cdot;\cdot]$ denotes a concatenation operation.

Short-term preference explorer. Unlike the user long-term preference, the short-term preference is constantly updated as the conversation progresses. At the t-th turn, we first sample the previous user short-term preference m^q_{t-1} from $q_{\phi_m}(m_{t-1})$ to infer the prior distribution. Here, m^q_{t-1} reflects the user specific preference in the previous turns. Then we encode m_{t-1} to a hidden representation H^{mq}_{t-1}. We denote the context as $[H^{mq}_{t-1}; H^C_t; H^{tp}_t]$ and employ a decoder in Eq. 9 to decode m^p_t. At each decoding step, we project the

hidden representation to the vocabulary space, thus we have:

$$\begin{aligned} H^X &= [H^{m^q}_{t-1}; H^c_t; H^{tp}_t] \\ p_{\theta_m}(m_t) &= \prod_{\zeta=1}^{|m|} \text{decoder}^v(H^X, m^q_{t,1:\zeta-1}). \end{aligned} \tag{15}$$

To approximate the posterior preference distribution, we need to encode H^A_t. For the whole decoding process, we calculate $q_{\phi_m}(m_t)$ as follows:

$$\begin{aligned} H^X &= [H^{m^q}_{t-1}; H^c_t; H^{tp}_t, H^A_t] \\ q_{\phi m}(m_t) &= \prod_{\zeta=1}^{|m|} \text{decoder}^v(H^X, m^q_{t,1:\zeta-1}). \end{aligned} \tag{16}$$

3.6 Policy network

At the t-th turn, action A_t denotes a set of topics from $\mathcal{K}$ or items in $\mathcal{I}$. The CRS will ask a clarification question if A_t refers to a set of topics, otherwise it recommends the selected items to the user.

External knowledge has been shown to be effective for improving the performance of dialogue actions in TDSs [32]. Following [55, 56], we present an external knowledge graph $\mathcal{G} = \{\mathcal{E}, \mathcal{R}\}$, where $\mathcal{E}$ indicates a set of entities and $\mathcal{R}$ indicates a set of relations. $\mathcal{E}$ contains topics and items, i.e., $\mathcal{E} = \mathcal{K} \cup \mathcal{I}$. A triple in $\mathcal{G}$ is denoted as $\langle e_1, r, e_2 \rangle$, where $e_1, e_2 \in \mathcal{E}$ are entities, and $r \in \mathcal{R}$ is the relation. Since the semantics of a relationship are crucial to examine, we use R-GCN [36] to learn entity representations. Formally, the representation of entity at the $(l+1)$-st layer is calculated as:

$$\boldsymbol{n}_e^{(l+1)} = \sigma\left(\sum_{r \in \mathcal{R}} \sum_{e' \in \mathcal{E}_e^r} \frac{1}{Z_{e,r}} \mathbf{W}_r^{(l)} \boldsymbol{n}_{e'}^{(l)} + \mathbf{W}^{(l)} \boldsymbol{n}_e^{(l)} \right), \tag{17}$$

where $\boldsymbol{n}_e^{(l)} \in R^d$ is the representation of entity e at the l-th layer. Given the relation r, $\mathcal{E}_e^r$ denotes the collection of nearby entities for e. $\mathbf{W}_r^{(l)}$ is a learnable relation-specific transformation matrix for the embeddings from neighborhood nodes with relation r; $\mathbf{W}^{(l)}$ is a learnable matrix for transforming the representations of entities at the l-th layer; and $Z_{e,r}$ is a normalization factor.

At the final layer L, the representation h_e^L is taken as the entity representation. Starting from the topics in topic path tp_t, we extract their 2-hop triples on the $\mathcal{G}$ as subgraph $\mathcal{G}^{sub}$. Then we concatenate the representations of entities on the $\mathcal{G}^{sub}$ as knowledge representation as:

$$H^g_t = [\mathbf{h}^L_{e_1}; \mathbf{h}^L_{e_2}; \ldots; \mathbf{h}^L_{e_{|g|}}], \tag{18}$$

where $|g|$ is the number of entities on the subgraph, d is the dimension of hidden vector, $[\cdot;\cdot]$ denotes vector concatenation.

During the training stage, at the t-th turn, we first sample long-term user preference l^q from $q_{\phi_l}(l)$ and short-term user preference m^q_t from $q_{\phi_m}(m_t)$. Then we utilize a transformer encoder to encode l^q to H^{l^q}, m^q_t to $H^{m^q}_t$ respectively. At the ζ-th decoding step, the decoder regards $[H^{l^q}; H^{m^q}_t; H^c_t, H^{tp}_t, H^g_t]$ as context, and sequentially outputs $b^A_{t,\zeta}$ given previous token embedding $e^A_{t,\zeta-1}$. The decoder then projects $b^A_{t,\zeta}$ into action space. Suppose A_t's length is $|A_t|$, the generative probability of A_t is calculated using Eq. 11:

$$\begin{aligned} X &= [l^q; m^q_t; C_t; tp_t; \mathcal{G}^{sub}] \\ H^X &= [H^{l^q}; H^{m^q}_t; H^c_t, H^{tp}_t, H^g_t] \\ p_{\theta_a}\left(A_{t,\zeta} | X, A_{t,1:\zeta-1}\right) &= \text{decoder}^{gc}(X, H^X, A_{t,1:\zeta-1}). \end{aligned} \tag{19}$$

We aim to maximize $p_{\theta_a}(A_t|\cdot)$ and minimize the divergence between prior and approximate posterior distributions. By using Eq. 6 we formulate the objective function as:

$$\begin{aligned} \mathcal{L}_a = &- \frac{1}{|A_t|} \sum_{\zeta=1}^{|A_t|} \log(p_{\theta_a}(A_{t,\zeta}|X, A_{t,1:\zeta-1})) \\ &+ KL(p_{\theta_m}(m_t) \| q_{\varphi_m}(m_t)) + KL(p_{\theta_l}(l) \| q_{\varphi_l}(l)). \end{aligned} \tag{20}$$

During the test stage, we only execute prior preference explorers to get preference distributions as substitutes for posterior preference distributions.

3.7 Response generator

For user u, given context C, topic path tp_t and action A_t, UPCR aims to generate a reply utterance s_t in a CRS. Using Eq. 7, we encode A_t into H^A_t, then we concatenate H^A_t, h^u, H^C and H^{tp}_t. At the ζ-th decoding step, the decoder outputs $b^s_{t,\zeta}$ given previous embedding $e^s_{t,j-1}$. In order to copy words from the input, the probability of generating $s_{t,\zeta}$ is calculated using Eq. 11 as follows:

$$\begin{aligned} X &= [A_t, C_t, tp_t, u] \\ H^X &= [H^A_t, H^c_t, H^{tp}_t, h^u] \\ p_{\theta_g}\left(s_{t,\zeta}\right) &= \text{decoder}^{gc}(X, H^X, s_{t,1:\zeta-1}). \end{aligned} \tag{21}$$

Suppose the length of s_t is $|s_t|$, we set a cross-entropy loss to learn the parameter in our response generator:

$$\mathcal{L}_{gen} = -\frac{1}{|s_t|} \sum_{\zeta=1}^{|s_t|} \log(p_{\theta_g}(s_{t,\zeta})). \tag{22}$$

To train UPCR in an end-to-end way, we integrate our optimization objectives with a weighted parameter λ to get the final objective $\mathcal{L}$:

$$\mathcal{L} = \mathcal{L}_a + \lambda \cdot \mathcal{L}_{gen}. \tag{23}$$

4 EXPERIMENTAL SETUP

4.1 Research questions

In our experiments, we address the following research questions: (RQ1) Does UPCR outperform state-of-the-art CR methods in terms of action prediction and response generation? (RQ2) How much does each component of UPCR contribute to its overall performance? (RQ3) Does the length of the latent variables (i.e., $|l|$ and $|m|$) have an effect on the performance?

4.2 Datasets

We conduct our experiments on two widely-applied benchmark datasets on conversational recommendation to evaluate the effectiveness of UPCR. Statistics of the two datasets are shown in Table 2.

TG-ReDial dataset. The TG-ReDial dataset [57] is composed of 10,000 two-party dialogues between a user and a recommender

Table 2: Statistics of the datasets we use in our experiments.

Dataset	Conversation	Utterance	Movie	Topic
TG-ReDial	10,000	129,392	33,834	2,571
REDIAL	10,006	182,150	51,699	12,669

in the movie domain. In total, it contains 129, 392 utterances from 1, 482 users. The dataset is constructed in a topic-guided way, viz., each conversation in the TG-ReDial dataset includes a topic path to enforce natural semantic transitions towards recommendation. On average, a dialogue in the TG-ReDial dataset has 7.9 topics and an utterance contains 19 words.

REDIAL dataset. The REDIAL dataset [21] is widely used in CRSs [28, 31, 55, 56]. The REDIAL dataset is collected by crowdsourcing workers from the Amazon Mechanical Turk platform. The workers create conversations for the task of movie recommendation in a user-recommender pair setting after following a set of detailed instructions. The REDIAL dataset has 10, 006 discussions with 182, 150 utterances relating to 51, 699 films. Following Zhou et al. [56], we obtain topics mentioned in each utterance. For each interaction, we generate dialogue actions and reply utterances.

4.3 Baselines and comparisons

Our baselines for assess the performance of action prediction and response generation come in three groups: (1) **Recommendations**: To evaluate the performance of UPCR in action prediction, we use Popularity, TextCNN [13], and BERT [7] as baselines. Popularity ranks items according to the number of interactions. TextCNN adopts a CNN-based model to extract textual features from contextual utterances. BERT is a pre-trained language model that directly encodes the concatenated historical utterances for recommendation. (2) **Knowledge grounded conversations**: To evaluate the performance of UPCR in action prediction and response generation, we use PostKS [23] as a baseline. PostKS uses dialogue context and responses to infer the posterior knowledge distribution. (3) **Conversational recommendations**: To evaluate the performance of UPCR in action prediction and response generation, we use KBRD [3], DCR [24], REDIAL [21], MGCG [27], TG-ReDial [57], KGSF [56] and CR-Walker [31] as baselines. KBRD utilizes knowledge graphs to enhance the semantics of contextual items for recommendation, then applies a transformer to generate responses. DCR uses a pointer network to incorporate global topic control and GCN-based recommendations in response generation. REDIAL is a benchmark model of the REDIAL dataset. It utilizes an auto-encoder [37] for recommendation and a hierarchical RNN for response generation. MGCG uses CNN-based multi-task classification to predict the current topic. TG-ReDial is a benchmark of the TG-ReDial dataset, which predicts topics or items in the first state, and then generates responses based on the predictions. KGSF incorporates word-oriented and entity-oriented knowledge graphs to enhance the user representations. CR-Walker uses reasoning on a knowledge graph to obtain a so-called reasoning tree and generates responses conditioned on the reasoning tree and user utterances.

For topic prediction, we compare UPCR with MGCG and TG-Redial as other CR baselines do not provide explicit topic prediction results. We compare UPCR with all the CR baselines for recommendation and generation evaluations.

To assess the performance of response generation with an encoder-decoder framework, we consider the transformer [43] as an additional baseline.

To answer RQ2, we consider three variations of UPCR: (1) *UPCR*$\backslash g$ removes the subgraph from the policy network in UPCR; (2) *UPCR*$\backslash l$ removes the long-term preference explorer from UPCR; (3) *UPCR*$\backslash m$ removes the short-term preference explorer in the preference explorer from UPCR.

4.4 Evaluation metrics

Automatic evaluation. To evaluate the performance on the topic prediction task, following TG-ReDial [57], we adopt Hit@k(k = 1, 3, 5) as evaluation metrics for ranking all the possible topics. To measure the effectiveness on the recommendation task, we evaluate UPCR in different settings using metrics proposed in the original datasets: For the TG-ReDial dataset, following Zhou et al. [57], we adopt NDCG@k and MRR@k (k = 10, 50) as evaluation metrics. For the REDIAL dataset, following [56], we adopt Recall@k (k = 1, 10, 50) for evaluation. To assess the quality of the generated responses, following [57], we adopt BLEU and Distinct-n [20] (n = 1, 2), for word-level matches and diversity in the TG-ReDial dataset. For the REDIAL dataset, following [56], we adopt Distinct-n (n = 2, 3, 4) as evaluation metrics.

Human evaluation. The system's ability to provide informative replies relating to items or topics is critical for CRS. Hence, we adopt human evaluation. We take 100 dialogues from our model and their respective generations, as well as the baselines, at random. Following [31, 56], we enlist the help of three experienced annotators from a third-party organization to assess the results of several models in two aspects, namely *fluency* and *informativeness*. *Fluency* measures if the generated response is smooth; *informativeness* measures whether the system introduces rich movie knowledge or related topics. The score ranges from 0 to 2. The average ratings of the three annotators are used to calculate the final performance.

4.5 Implementation details

We implement UPCR in PyTorch. The default parameter settings across all experiments are as follows: we set the batch size to 16, gradient accumulation step to 8. The embedding size is set as 512. In the path reasoning process, we set the maximum number of nodes to 200, the graph hidden size to 512. The topic vocabulary size is 2,571 for the TG-ReDial dataset and 12,669 for the REDIAL dataset. The word vocabulary size is 19,119 for the TG-ReDial dataset and 23,928 for the REDIAL dataset. For the REDIAL dataset, we used negative sampling for the candidate items during the recommendation. The lengths of the long-term preference text span and short-term preference text span are set to 10 and 5, respectively. We set the temperature of Gumbel-Softmax to $\tau = 3.0$, and anneal to 0.1 in 30,000 training steps. We use the Adam optimizer [14], and the learning rate is initialized to $1e^{-4}$ and decreases to $1e^{-5}$ gradually.

5 EXPERIMENTAL RESULTS

5.1 Performance on action prediction (RQ1)

We start to address RQ1 by evaluating the performance on the action prediction task. Recall that on the TG-Redial dataset, actions consist

Table 3: Automatic evaluation of topic prediction on TG-Redial dataset. Bold face indicates best results. Significant improvements over best baseline marked with * (t-test, $p < 0.05$).

Model	Hit@1	Hit@3	Hit@5
Popularity	0.0412	0.0815	0.0962
TextCNN	0.3815	0.4621	0.5163
BERT	0.6114	0.8189	0.8341
PostKS	0.3308	0.4527	0.5083
MGCG	0.6098	0.8128	0.8294
TG-Redial	0.6231	0.8370	0.8497
UPCR	**0.8078***	**0.8827***	**0.9066***

of topics or items. If the action refers to topics, the system will continue by asking a clarification question. If the action incorporates items, the system will recommend items to the user. On the REDIAL dataset, the action only consists of items to be recommended.

Table 3 and 4 show the experimental results of topic prediction and recommendation, respectively. When the action refers to a set of topics, Table 3 presents the performance of various methods on the TG-Redial dataset. Popularity does not perform well, since it cannot consider the context of conversations. We find TG-ReDial outperforms other baselines, since it jointly models context, topics, and user profiling. UPCR gives an increase of 29.6%, 5.5%, and 6.7% over TG-ReDial in terms of Hit@1, Hit@3, and Hit@5 respectively as UPCR captures user preferences rather than using topics directly.

In summary, we conclude that better understanding of user preferences is helpful for improving the performance of CR.

Table 4 shows the recommendation performance on the TG-ReDial and REDIAL datasets. Content-based recommendation models (i.e., TextCNN and BERT) outperform Popularity, which indicates that historical utterances are useful for making recommendations. Knowledge-based CRS methods (i.e., KGSF and CR-Walker) outperform other baselines, indicating that a knowledge graph is crucial for making recommendations in CR. UPCR outperforms all the baselines on both datasets. For the TG-ReDial dataset, UPCR achieves a significant improvement over CR-Walker: 47.7%, 26.8%, 35.9%, 27.6% in terms of NDCG@10, NDCG@50, MRR@10, MRR@50, respectively. For the REDIAL dataset, UPCR achieves an increase of 11.1%, 21.4%, and 22.3% over (second best) CR-Walker in terms of R@1, R@10, R@50, respectively.

In summary, by tracking user preferences, UPCR is able to provide more accurate recommendations to meet users' needs than the current state-of-the-art.

5.2 Performance of response generation (RQ1)

We continue to address RQ1 and examine response generation. We evaluate the performance in terms of automatic and human evaluation metrics.

Automatic evaluation. In Table 5, we examine the generation performance in terms of automatic metrics on the TG-Redial and REDIAL datasets. For the TG-ReDial dataset, we find that TG-ReDial performs best among the baselines, which shows that pre-trained

Table 4: Automatic evaluation of recommendation on TG-Redial and REDIAL datasets. Bold face indicates best result. Significant improvements over best baseline results marked with * (t-test, $p < 0.05$).

Model	TG-Redial				REDIAL		
	NDCG		MRR		Recall		
	@10	@50	@10	@50	@1	@10	@50
Popularity	0.0015	0.0036	0.0011	0.0015	0.012	0.061	0.179
TextCNN	0.0144	0.0215	0.0119	0.0133	0.017	0.096	0.159
BERT	0.0246	0.0439	0.0182	0.0221	0.018	0.117	0.191
PostKS	0.0031	0.0048	0.0029	0.0038	0.019	0.122	0.236
KBRD	0.0064	0.0111	0.0040	0.0049	0.030	0.163	0.338
DCR	0.0261	0.0498	0.0129	0.0179	0.027	0.148	0.306
REDIAL	0.0006	0.0025	0.0003	0.0007	0.023	0.129	0.287
MGCG	0.0184	0.0412	0.0130	0.0210	0.027	0.121	0.264
TG-Redial	0.0348	0.0527	0.0240	0.0277	0.041	0.164	0.310
KGSF	0.0154	0.0259	0.0114	0.0135	0.039	0.183	0.378
CR-Walker	0.0565	0.0771	0.0489	0.0565	0.040	0.187	0.376
UPCR	**0.0835***	**0.1105***	**0.0665***	**0.0721***	**0.046***	**0.238***	**0.484***

generative models improve the generation performance. KGSF outperforms REDIAL and KBRD in terms of diversity on the REDIAL dataset as it incorporates KG-enhanced representations of items and words to generate an utterance. CR-Walker utilizes a pre-trained language model to generate an utterance and achieves better performance than KGSF.

UPCR performs best on both datasets. For the TG-ReDial dataset, UPCR achieves an increase of 11.4%, 21.7%, 27.9%, 4.5%, 28.7% over TG-Redial in terms of BLEU-1, BLEU-2, BLEU-3, Distinct-1, Distinct-2. For the REDIAL dataset, UPCR achieves an increase of 5.2%, 20.3%, 31.0% over CR-Walker in terms of Distinct-2, Distinct-3, Distinct-4.

Human evaluation. Table 5 also lists the human evaluation results of response generation on the TG-ReDial dataset. By using a pre-trained model, TG-ReDial achieves a comparable fluency performance as UPCR. We find that UPCR achieves an increase of 7.3% over TG-ReDial in terms informativeness, which indicates that UPCR is able to generate more informative utterances by incorporating multi-type data representations. Besides, UPCR uses a copy mechanism, which is also helpful for generating informative utterances.

5.3 Ablation study (RQ2)

Next, we turn to RQ2. To analyze where the improvements of UPCR come from, we conduct an ablation study on the TG-ReDial and REDIAL datasets; see Table 6 for the results. Briefly, all components are helpful for recommendation because the performance drops without any of them. Specifically, on the TG-ReDial dataset, the performance of UPCR\l and UPCR\m drops by 19.1% and 22.1% on NDCG@50, respectively.

On the REDIAL dataset, the performance of UPCR\l and UPCR\m drops by 14.7% and 16.4% on Recall@10,respectively. Results on both datasets show that long-term and short-term preferences are helpful to improve the recommendation performance

Table 5: Automatic and human evaluation of response generation on TG-Redial and REDIAL datasets. For comparability with prior work, we adopt the BLEU and Distinct metrics for the TG-Redial dataset [56], and the Distinct metrics for the REDIAL dataset [21]. Bold face indicates best results. Significant improvements over best baseline marked with * (t-test, $p < 0.05$).

Model	TG-Redial							REDIAL		
	Automatic					Human		Automatic		
	BLEU-1	BLEU-2	BLEU-3	Distinct-1	Distinct-2	Fluency	Informativeness	Distinct-2	Distinct-3	Distinct-4
PostKS	0.142	0.018	0.006	0.005	0.021	1.23	0.83	0.074	0.126	0.224
KBRD	0.221	0.028	0.009	0.004	0.008	1.17	1.12	0.086	0.153	0.265
DCR	0.128	0.021	0.007	0.008	0.021	0.92	0.91	0.081	0.138	0.233
REDIAL	0.069	0.008	0.002	0.015	0.062	1.22	1.02	0.082	0.143	0.245
MGCG	0.242	0.057	0.023	0.011	0.041	1.34	1.34	0.101	0.189	0.261
TG-Redial	0.280	0.065	0.031	0.021	0.094	**1.45**	1.36	0.086	0.153	0.216
KGSF	0.239	0.042	0.013	0.015	0.062	1.38	1.33	0.114	0.204	0.282
CR-Walker	0.271	0.059	0.028	0.019	0.081	1.43	1.34	0.163	0.289	0.365
Transformer	0.261	0.061	0.027	0.014	0.083	1.28	0.72	0.067	0.139	0.227
UPCR	**0.316***	**0.083***	**0.043***	**0.022**	**0.132***	1.44	**1.51***	**0.172***	**0.363***	**0.529***

Table 6: Results of the ablation study for the recommendation task. Bold face indicates best results. Significant improvements over best baseline marked with * (t-test, $p < 0.05$).

Model	TG-Redial				REDIAL		
	NDCG		MRR		Recall		
	@10	@50	@10	50@10	@1	@10	@50
UPCR\g	0.0411	0.0611	0.0323	0.0363	0.041	0.210	0.445
UPCR\l	0.0586	0.0894	0.0456	0.0518	0.042	0.203	0.453
UPCR\m	0.0524	0.0861	0.0408	0.0476	0.039	0.199	0.433
UPCR	**0.0835***	**0.1105***	**0.0665***	**0.0721***	**0.046***	**0.238***	**0.484***

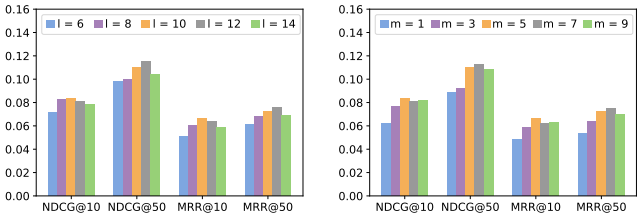

Figure 4: The effect of preference length on recommendation performance on the TG-ReDial dataset. Left: l. Right: m.

in CR. Without path reasoning in the policy network, the performance of UPCR drops on both datasets, which verifies its importance in UPCR; the performance drops sharply on the TG-Redial dataset, which shows that external knowledge plays an essential role on that dataset.

5.4 Impact of $|l|$ and $|m|$ (RQ3)

To address RQ3, we conduct experiments on the TG-ReDial dataset by setting $|l|$ to values in {6,8,10,12,14} while fixing $|m|$ to 5, and select $|m|$ from {1,3,5,7,9} while fixing $|l|$ to 10. The results are shown in Fig. 4. As $|l|$ increases from 6 to 10 and $|m|$ increases from 1 to 5, UPCR achieves 12.4% and 23.5% improvements in terms of NDCG@10, respectively. A short text span is not able to capture user preferences, thus the model cannot generate accurate recommendations. As $|l|$ increases from 10 to 14 and $|m|$ increases from 5 to 9, the NDCG@10 drops by 5.6% and 1.7% in terms of NDCG@10, respectively. The main reason is that a long text span may contain meaningless or repetitive topics and bring noise to the model.

5.5 Case study

To gain a more qualitative understanding of the relative quality of responses generated by UPCR vs. TG-ReDial, we randomly sample a conversation generated by the two models on the TG-ReDial dataset. See Table 7. UPCR is able to generate accurate recommendations and interpretative responses. For example, in the third utterance, UPCR captures the user preference through the long-term and short-term user preference explorer and recommends the movie "Coming Soon" to the user.

In addition to recommendations, UPCR generates explanations that not only conform to the characteristics of the movie, but also appear to meet the needs of the user. In contrast, TG-ReDial makes a wrong recommendation. A possible reason is that TG-ReDial does not explicitly explore the user preference, and only utilizes context and history interactions to represent the user, which cannot accurately represent the user preference.

6 CONCLUSIONS

In this paper, we have focused on the task of conversational recommendation. We have noticed that long-term and short-term user preferences play an important role in topic prediction and recommendation. To tackle the challenge about unannotated user preference, we have proposed a new model, namely UPCR, which regards long-term user preference and short-term user preference as latent variables. We utilize variational Bayesian generative approach to estimate posterior distributions over long-term and short-term

Table 7: One case extracted from TG-ReDial. Due to space limitations, we only show the first turns of the conversation.

Context	**s1(recommender)**: What are you doing, let me recommend a movie for you. **s2(user)**: OK, I want to see a logical horror movie.
Response	**Ground truth**: Then I strongly recommend ***Coming Soon***, it's real and scary.
	TG-ReDial: You can watch ***The Reaping***, it's one of the scariest horror movies I've ever seen.
	UPCR: I recommend you go to see ***Coming Soon***, some of the horror atmosphere is well created, and the ending is unexpected. *long-term preference*: *plot, thriller, horror, surprise, natural, real, exciting, economic, wealth, emotional.* *short-term preference*: *logical, horror, thriller, scary, reasonable.*

user preferences. Extensive experiments on two conversational recommendation datasets show that UPCR achieves state-of-the-art performance.

Our method proves the value of exploring long-term and short-term user preferences for conversational recommendation. However, a limitation of UPCR is that it does not consider optimizing pre-trained language models by leveraging multi-type external data for recommendation and response generation. As to our future work, we plan to take fine-grained external knowledge into account within a pre-training procedure to improve the generation performance. Also, using coarse-to-fine grained self-supervision should give insights in conversational recommendation.

REPRODUCIBILITY

This work uses publicly available data. To facilitate reproducibility of the results reported in this paper, the code used is available at https://github.com/tianz2020/UPCR.

ACKNOWLEDGMENTS

This work was supported by the Natural Science Foundation of China (61902219, 61972234, 62072279, 62102234), the Natural Science Foundation of Shandong Province (ZR2021QF129), the Key Scientific and Technological Innovation Program of Shandong Province (2019JZZY010129), Shandong University multidisciplinary research and innovation team of young scholars (No. 2020QNQT017), Meituan, the Tencent WeChat Rhino-Bird Focused Research Program (JR-WXG-2021411), the Hybrid Intelligence Center, a 10-year program funded by the Dutch Ministry of Education, Culture and Science through the Netherlands Organisation for Scientific Research, https://hybrid-intelligence-centre.nl. All content represents the opinion of the authors, which is not necessarily shared or endorsed by their respective employers and/or sponsors.

REFERENCES

[1] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. DBpedia: A Nucleus for a Web of Open Data. In *The Semantic Web*. Springer, 722–735.
[2] Bartoz Balcer, Martin Halvey, Stephen A Brewster, and Joemon M Jose. 2012. COPE: Interactive Image Retrieval Using Conversational Recommendation. In *Proceedings of BCS-HCI*. 1–10.
[3] 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 EMNLP*. 1803–1813.
[4] Konstantina Christakopoulou, Alex Beutel, Rui Li, Sagar Jain, and Ed H Chi. 2018. Q&R: A Two-stage Approach toward Interactive Recommendation. In *Proceedings of KDD*. 139–148.
[5] Konstantina Christakopoulou, Filip Radlinski, and Katja Hofmann. 2016. Towards Conversational Recommender Systems. In *Proceedings of KDD*. 815–824.
[6] 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 SIGIR*. 1431–1441.
[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of NAACL-HLT*. 4171–4186.
[8] Bhuwan Dhingra, Lihong Li, Xiujun Li, Jianfeng Gao, Yun-Nung Chen, Faisal Ahmed, and Li Deng. 2017. Towards End-to-End Reinforcement Learning of Dialogue Agents for Information Access. In *Proceedings of ACL*. 484–495.
[9] Matt W Gardner and SR Dorling. 1998. Artificial Neural Networks (The Multilayer Perceptron) Review of Applications in the Atmospheric Sciences. *Atmospheric environment* 32, 14-15 (1998), 2627–2636.
[10] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT press.
[11] Jiatao Gu, Zhengdong Lu, Hang Li, and Victor OK Li. 2016. Incorporating Copying Mechanism in Sequence-to-Sequence Learning. In *Proceedings of ACL*.
[12] Yangyang Guo, Zhiyong Cheng, Liqiang Nie, Yinglong Wang, Jun Ma, and Mohan Kankanhalli. 2019. Attentive Long Short-term Preference Modeling for Personalized Product Search. *ACM Transactions on Information Systems (TOIS)* 37, 2 (2019), 1–27.
[13] Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. In *Proceedings of EMNLP*. 1746–1751.
[14] Diederik P Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *Proceedings of ICLR*.
[15] Diederik P Kingma and Max Welling. 2014. Auto-Encoding Variational Bayes. In *Proceedings of ICLR*.
[16] Yuquan Le, Xian Li, Suixue Wang, Peng Wang, Haiqian Lin, and Guanyu Jiang. 2018. CVTE SLU: A Hybrid System for Command Understanding Task Oriented to the Music Field. In *Proceedings of CCKS*. 13–18.
[17] Chih-Hen Lee, Jun-En Ding, Chih-Ming Chen, Jing-Kai Lou, Ming-Feng Tsai, and Chuan-Ju Wang. 2021. LSTPR: Graph-based Matrix Factorization with Long Short-term Preference Ranking. In *Proceedings of SIGIR*. 2222–2226.
[18] 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 *Proceedings of WSDM*. 304–312.
[19] 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 KDD*. 2073–2083.
[20] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A Diversity-Promoting Objective Function for Neural Conversation Models. In *Proceedings of NAACL*. 110–119.
[21] Raymond Li, Samira Ebrahimi Kahou, Hannes Schulz, Vincent Michalski, Laurent Charlin, and Chris Pal. 2018. Towards Deep Conversational Recommendations. In *Proceedings of NeurIPS*. 9748–9758.
[22] 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 Transactions on Information Systems* 39, 4 (2021), 1–29.
[23] Rongzhong Lian, Min Xie, Fan Wang, Jinhua Peng, and Hua Wu. 2019. Learning to Select Knowledge for Response Generation in Dialog Systems. In *Proceedings of IJCAI*. 5081–5087.
[24] Lizi Liao, Ryuichi Takanobu, Yunshan Ma, Xun Yang, Minlie Huang, and Tat-Seng Chua. 2019. Deep Conversational Recommender in Travel. *CoRR* abs/1907.00710 (2019).
[25] Yong Liu, Yingtai Xiao, Qiong Wu, Chunyan Miao, Juyong Zhang, Binqiang Zhao, and Haihong Tang. 2020. Diversified Interactive Recommendation with Implicit Feedback. In *Proceedings of AAAI*. 4932–4939.
[26] Yu Liu, Haiping Zhu, Yan Chen, Feng Tian, Dailusi Ma, Jiangwei Zeng, and Qinghua Zheng. 2020. Long- and Short-Term Preference Model Based on Graph Embedding for Sequential Recommendation. In *Proceedings of DASFAA*. 241–257.
[27] Zeming Liu, Haifeng Wang, Zheng-Yu Niu, Hua Wu, Wanxiang Che, and Ting Liu. 2020. Towards Conversational Recommendation over Multi-Type Dialogs. In *Proceedings of ACL*. 1036–1049.
[28] Yu Lu, Junwei Bao, Yan Song, Zichen Ma, Shuguang Cui, Youzheng Wu, and Xiaodong He. 2021. RevCore: Review-Augmented Conversational Recommendation. In *Proceedings of ACL-IJCNLP*. 1161–1173.
[29] Kai Luo, Hojin Yang, Ga Wu, and Scott Sanner. 2020. Deep Critiquing for VAE-based Recommender Systems. In *Proceedings of SIGIR*. 1269–1278.

[30] Shengnan Lyu, Arpit Rana, Scott Sanner, and Mohamed Reda Bouadjenek. 2021. A Workflow Analysis of Context-driven Conversational Recommendation. In *Proceedings of WWW*. 866–877.

[31] Wenchang Ma, Ryuichi Takanobu, and Minlie Huang. 2021. CR-Walker: Tree-Structured Graph Reasoning and Dialog Acts for Conversational Recommendation. In *Proceedings of EMNLP*. 1839–1851.

[32] Chuan Meng, Pengjie Ren, Zhumin Chen, Weiwei Sun, Zhaochun Ren, Zhaopeng Tu, and Maarten de Rijke. 2020. DukeNet: A Dual Knowledge Interaction Network for Knowledge-Grounded Conversation. In *Proceedings of SIGIR*. 1151–1160.

[33] Kevin P Murphy. 2012. *Machine Learning: A Probabilistic Perspective*. MIT press.

[34] Xuhui Ren, Hongzhi Yin, Tong Chen, Hao Wang, Zi Huang, and Kai Zheng. 2021. Learning to Ask Appropriate Questions in Conversational Recommendation. In *Proceedings of SIGIR*. 808–817.

[35] Rajdeep Sarkar, Koustava Goswami, Mihael Arcan, and John P. McCrae. 2020. Suggest Me a Movie for Tonight: Leveraging Knowledge Graphs for Conversational Recommendation. In *Proceedings of COLING*. 4179–4189.

[36] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. Modeling Relational Data with Graph Convolutional Networks. In *Proceedings of ESWC*. 593–607.

[37] Suvash Sedhain, Aditya Krishna Menon, Scott Sanner, and Lexing Xie. 2015. Autorec: Autoencoders Meet Collaborative Filtering. In *Proceedings of WWW*. 111–112.

[38] Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. ConceptNet 5.5: An Open Multilingual Graph of General Knowledge. In *Proceedings of AAAI*. 4444–4451.

[39] Yueming Sun and Yi Zhang. 2018. Conversational Recommender System. In *Proceedings of SIGIR*. 235–244.

[40] Cynthia A Thompson, Mehmet H Goker, and Pat Langley. 2004. A Personalized System for Conversational Recommendations. *Journal of Artificial Intelligence Research* 21 (2004), 393–428.

[41] Thanh Tran, Di You, and Kyumin Lee. 2020. Quaternion-Based Self-Attentive Long Short-term User Preference Encoding for Recommendation. In *Proceedings of CIKM*. 1455–1464.

[42] Quan Tu, Shen Gao, Yanran Li, Jianwei Cui, Bin Wang, and Rui Yan. 2022. Conversational Recommendation via Hierarchical Information Modeling. In *Proceedings of SIGIR*.

[43] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. In *Proceedings of NeurIPS*. 5998–6008.

[44] Ivan Vendrov, Tyler Lu, Qingqing Huang, and Craig Boutilier. 2020. Gradient-Based Optimization for Bayesian Preference Elicitation. In *Proceedings of AAAI*. 10292–10301.

[45] Huazheng Wang, Qingyun Wu, and Hongning Wang. 2017. Factorization Bandits for Interactive Recommendation. In *Proceedings of AAAI*. 2695–2702.

[46] Qing Wang, Chunqiu Zeng, Wubai Zhou, Tao Li, S Sitharama Iyengar, Larisa Shwartz, and Genady Ya Grabarnik. 2018. Online Interactive Collaborative Filtering Using Multi-armed Bandit with Dependent Arms. *IEEE Transactions on Knowledge and Data Engineering* 31, 8 (2018), 1569–1580.

[47] Tsung-Hsien Wen, David Vandyke, Nikola Mrksic, Milica Gasic, Lina M Rojas-Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. 2017. A Network-based End-to-End Trainable Task-oriented Dialogue System. In *Proceedings of EACL*. 438–449.

[48] Ronald J Williams and David Zipser. 1989. A Learning Algorithm for Continually Running Fully Recurrent Neural Networks. *Neural computation* 1, 2 (1989), 270–280.

[49] 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 WSDM*. 364–372.

[50] Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. 2017. SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient. In *Proceedings of AAAI*. 2852–2858.

[51] Xiaoying Zhang, Hong Xie, Hang Li, and John Lui. 2019. Toward Building Conversational Recommender Systems: A Contextual Bandit Approach. *arXiv preprint arXiv:1906.01219* (2019).

[52] Yongfeng Zhang, Xu Chen, Qingyao Ai, Liu Yang, and W Bruce Croft. 2018. Towards Conversational Search and Recommendation: System Ask, User Respond. In *Proceedings of CIKM*. 177–186.

[53] Zheng Zhang, Minlie Huang, Zhongzhou Zhao, Feng Ji, Haiqing Chen, and Xiaoyan Zhu. 2019. Memory-augmented Dialogue Management for Task-oriented Dialogue Systems. *ACM Transactions on Information Systems* 37, 3 (2019), 1–30.

[54] Xiaoxue Zhao, Weinan Zhang, and Jun Wang. 2013. Interactive Collaborative Filtering. In *Proceedings of CIKM*. 1411–1420.

[55] Jinfeng Zhou, Bo Wang, Ruifang He, and Yuexian Hou. 2021. CRFR: Improving Conversational Recommender Systems via Flexible Fragments Reasoning on Knowledge Graphs. In *Proceedings of EMNLP*. 4324–4334.

[56] 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 KDD*. 1006–1014.

[57] Kun Zhou, Yuanhang Zhou, Wayne Xin Zhao, Xiaoke Wang, and Ji-Rong Wen. 2020. Towards Topic-Guided Conversational Recommender System. In *Proceedings of COLING*. 4128–4139.

[58] 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 WSDM*. 1488–1496.

[59] Jie Zou, Yifan Chen, and Evangelos Kanoulas. 2020. Towards Question-based Recommender Systems. In *Proceedings of SIGIR*. 881–890.

[60] Jie Zou, Evangelos Kanoulas, Pengjie Ren, Zhaochun Ren, Aixin Sun, and Cheng Long. 2022. Improving Conversational Recommender Systems via Transformer-based Sequential Modelling. In *Proceedings of SIGIR*.

[61] Lixin Zou, Long Xia, Pan Du, Zhuo Zhang, Ting Bai, Weidong Liu, Jian-Yun Nie, and Dawei Yin. 2020. Pseudo Dyna-Q: A Reinforcement Learning Framework for Interactive Recommendation. In *Proceedings of WSDM*. 816–824.

[62] Lixin Zou, Long Xia, Yulong Gu, Xiangyu Zhao, Weidong Liu, Jimmy Xiangji Huang, and Dawei Yin. 2020. Neural Interactive Collaborative Filtering. In *Proceedings of SIGIR*. 749–758.