# Multi-view Hypergraph Contrastive Policy Learning for Conversational Recommendation

Sen Zhao
CCIIP Laboratory, Huazhong University of Science and Technology
Joint Laboratory of HUST and Pingan Property & Casualty Research (HPL)
China
senzhao@hust.edu.cn

Wei Wei*
CCIIP Laboratory, Huazhong University of Science and Technology
Joint Laboratory of HUST and Pingan Property & Casualty Research (HPL)
China
weiw@hust.edu.cn

Xian-Ling Mao
Beijing Institute of Technology
China
maoxl@bit.edu.cn

Shuai Zhu
Ant Group
China
zs261988@antgroup.com

Minghui Yang
Ant Group
China
minghui.ymh@antgroup.com

Zujie Wen
Ant Group
China
zujie.wzj@antgroup.com

Dangyang Chen
Ping An Property & Casualty Insurance Company of China, Ltd
China
chendangyang273@pingan.com.cn

Feida Zhu
Singapore Management University
Singapore
fdzhu@smu.edu.sg

## ABSTRACT

Conversational recommendation systems (CRS) aim to interactively acquire user preferences and accordingly recommend items to users. Accurately learning the dynamic user preferences is of crucial importance for CRS. Previous works learn the user preferences with pairwise relations from the interactive conversation and item knowledge, while largely ignoring the fact that factors for a relationship in CRS are multiplex. Specifically, the user likes/dislikes the items that satisfy some attributes (*Like/Dislike view*). Moreover social influence is another important factor that affects user preference towards the item (*Social view*), while is largely ignored by previous works in CRS. The user preferences from these three views are inherently different but also correlated as a whole. The user preferences from the same views should be more similar than that from different views. The user preferences from *Like View* should be similar to *Social View* while different from *Dislike View*. To this end, we propose a novel model, namely Multi-view Hypergraph Contrastive Policy Learning (MHCPL). Specifically, MHCPL timely chooses useful social information according to the interactive history and builds a dynamic hypergraph with three types of multiplex relations from different views. The multiplex relations in each view are successively connected according to their generation order in the interactive conversation. A hierarchical hypergraph neural network is proposed to learn user preferences by integrating information of the graphical and sequential structure from the dynamic hypergraph. A cross-view contrastive learning module is proposed to maintain the inherent characteristics and the correlations of user preferences from different views. Extensive experiments conducted on benchmark datasets demonstrate that MHCPL outperforms the state-of-the-art methods.

## CCS CONCEPTS

• **Information systems → Recommender systems**.

## KEYWORDS

Conversational Recommendation, Reinforcement Learning, Graph Representation Learning

**ACM Reference Format:**
Sen Zhao, Wei Wei*, Xian-Ling Mao, Shuai Zhu, Minghui Yang, Zujie Wen, Dangyang Chen, and Feida Zhu. 2023. Multi-view Hypergraph Contrastive Policy Learning for Conversational Recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '23), July 23–27, 2023, Taipei, Taiwan.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3539618.3591737

*Corresponding author.


*SIGIR '23, July 23–27, 2023, Taipei, Taiwan*

ACM ISBN 978-1-4503-9408-6/23/07…$15.00
https://doi.org/10.1145/3539618.3591737

## 1 INTRODUCTION

Recommendation systems [12, 39, 41, 45, 50] are emerging as an efficient tool to help users find items of potential interest. They conventionally learn user preferences from their historical actions [16, 32], while hardly acquiring dynamic user preferences which often drift with time. To this end, conversational recommendation systems (CRS) [23] are proposed to dynamically acquire user preferences and accordingly make recommendations through interactive conversations. Different settings [6, 7, 34] of CRS are explored and

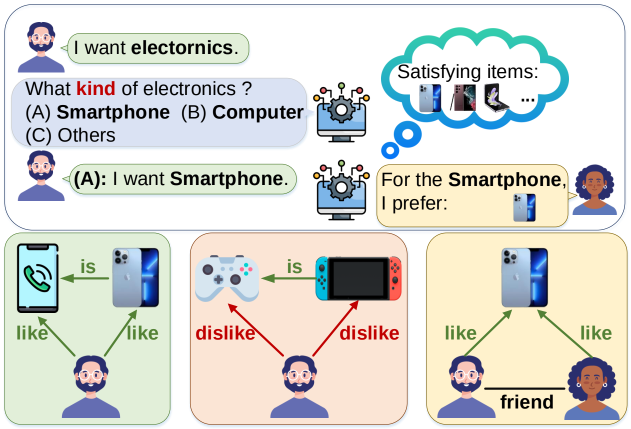


**Figure 1: Common types of multiplex user relations from different views in the scenario of conversational recommendation.**

we focus on the Multi-Interest Multi-round Conversational Recommendation (MMCR) [49] in which users could accept multiple items and CRS needs to strategically ask multi-choice questions about user-preferred attributes and accordingly recommend items, reaching success in the limited turns.

Learning the dynamic user preferences for the candidate attributes and items accurately is of crucial importance for CRS. CRM [34] and EAR [22] develop factorization-based methods to learn user preferences from pairwise interactions, but they fail to capture multi-hop information from the connectivity graph. SCPR [24] learns user preferences by reasoning the path on the user-item-attribute graph. Unicorn [9] and MCMIPL [49] further apply graph neural networks to learn user preferences from the graph structure that captures rich correlations among different types of nodes (*i.e.,* user, attribute, and item). Despite effectiveness, previous works learn user preferences with pairwise relations from the interactive conversation (*i.e.,* user-item and user-attribute relations) and the item knowledge (*i.e.,* item-attribute relations), while largely ignoring the fact that factors for a relationship in CRS are multiplex. For the example in Fig.1, the user dislikes Switch because of its attribute named "game console" rather than its other attributes like "electronics". Moreover, social influence is also an important factor that affects user preferences towards the item, since people with social connections will influence each other, leading to similar interests [8, 13]. However, in the field of CRS, social information is seldom explored. Inspired by the advantage of hypergraph [11, 46] in modeling the multiplex relations (*i.e.,* relations that connect more than two nodes), we investigate the potential of hypergraph modeling with the integration of interactive conversation, item knowledge, and social influence for learning dynamic user preferences in CRS.

Actually, it's non-trivial to build a hypergraph for learning dynamic user preferences in CRS, due to three challenges: 1) The first challenge is the dynamic filtering and utilizing of social information. The social information conventionally contains all the historical interactions of the user's friends, which could be noisy for the dynamic user preferences in the current conversation, since only friend preferences that satisfy the current conversation are helpful for dynamic user preferences learning. For the example in Fig.1, only the friends' preferences for "smartphone" are helpful for learning the dynamic user preferences. 2) The second challenge is hypergraph formulation. In the scenario of CRS (as illustrated in Fig.1), there mainly remain three multiplex relation patterns, that is, the user likes/dislikes the items that satisfy some attribute (*Like/Dislike view*) and the user shares the preferences for items with some friend (*Social view*). Each relation pattern corresponds to a kind of hyperedges, which are successively generated during the interactive conversation. 3) The third challenge is the aggregation of user preferences learned from different views, which might obscure the inherent characteristics of preference distributions from different views and the correlation between them. Specifically, user preferences from the same views should be more similar than user preferences from different views. And the user preferences from *Like View* should be similar to *Social View* while different from *Dislike View*. Contrastive learning [15, 38, 42], one successful self-supervised learning paradigm, which aims to learn discriminative representations by contrasting positive and negative samples, paves a way to maintain the inherent characteristics and the correlation of user preferences learned from different views.

To this end, we propose a novel hypergraph-based model, namely Multi-view Hypergraph Contrastive Policy Learning (MHCPL). Specifically, MHCPL dynamically filters social information according to the interactive conversation and builds a dynamic multi-view hypergraph with three types of multiplex relations from different views: the user likes/dislikes the items that satisfy some attribute (*Like/Dislike view*) and the user shares the preferences for items with some friend (*Social view*). The multiplex relations in each view are successively connected according to their generation order in the interactive conversation. A hierarchical hypergraph neural network is proposed to learn user preferences by integrating information of the graphical and sequential structure from the dynamic hypergraph. Furthermore, a cross-view contrastive learning module is proposed with two terms to maintain the inherent characteristics and the correlations of user preferences from different views. Extensive experiments conducted on Yelp and LastFM demonstrate that our proposed model MHCPL outperforms the state-of-the-art methods.

**Our contributions** of this work are summarized as follows:

- **General Aspects:** We emphasize the importance of multiplex relations and investigate three views of multiplex relations that integrate interactive conversation, item knowledge, and social influence for dynamic user preference learning in CRS.
- **Novel Methodologies:** We propose the model MHCPL, which builds a multi-view hypergraph contrastive policy learning framework for CRS. MHCPL timely filters social information according to the interactive conversation and learns dynamic user preferences with three types of multiplex relations from different views. Moreover, a cross-view contrastive learning module is proposed to maintain the inherent characteristics and the correlations of user preferences from different views.
- **Multifaceted Experiments:** We conduct extensive experiments on two benchmark datasets. The results demonstrate the advantage of our MHCPL in better dynamic user preference learning, which shows the effectiveness of our MHCPL for conversational recommendation.

# 2 RELATED WORKS

## 2.1 Conversational Recommendation

Conversational recommendation systems (CRS) [23, 31, 47, 51] aim to communicate with the user and recommend items based on the attributes explicitly asked during the conversation. Due to its ability to dynamically get the user's feedback, CRS has become an effective solution for capturing dynamic user preferences and solving the explainability problem. Various efforts have been conducted to explore the challenges in CRS which can mainly be categorized into two tasks: dialogue-biased CRS studies the dialogue understanding and generation [5, 19, 25, 26], and recommendation-biased CRS explores the strategy to consult and recommend [6, 7, 22, 34]. This work focuses on the recommendation-biased CRS.

Early works on the recommendation-biased CRS [6, 7, 34] only consider the conversational recommendation under simplified settings. For example, Christakopoulou *et al.* [7] consider the situation that CRS only needs to recommend without asking the user about his/her preferred attributes. The Q&A work [6] proposes to explore the situation that CRS jointly asks attributes and recommends items, but restricts the conversational recommendation to two turns: one to ask attributes and the other to recommend items. To explore a more realistic scenario of the recommendation-biased CRS, further efforts [22, 24] based on the reinforcement learning (RL) are conducted to explore the problem of multi-round conversational recommendation (MCR) which aims to strategically ask users binary questions towards attributes and recommend items in multiple rounds, achieving success in the limited turns. Zhang *et al.* [49] further explore the setting of multi-interest MCR (MMCR) where users may have multiple interests in attribute combinations and allows CRS to ask multi-choice questions towards the user-preferred attributes.

The main challenge of MCR is how to dynamically learn user preferences, and accordingly choose actions that satisfy user preferences. CRM [34] and EAR [22] learn user preferences with a factorization-based method under the pairwise Bayesian Personalized Ranking (BPR) framework [33]. SCPR [24] learns user preferences by reasoning the path on the user-item-attribute graph and strictly chooses actions on the path. Unicorn [9] builds a weighted graph to model the dynamic relationship between the user and the candidate action space and proposes a graph-based Markov Decision Process (MDP) environment to learn dynamic user preferences and choose actions from the candidate action space. MCMIPL [49] further considers the multiple interests of the user and develops a multi-interest policy learning module that combines the graph-based MDP with the multi-attention mechanism. Despite effectiveness, previous works model user preferences with binary relations, while hardly capturing the multiplex relations which are important in modeling dynamic user preferences for CRS. Furthermore, previous methods ignore the influence of social relations on user preferences.

## 2.2 Social Recommendation

Social recommendation [13, 18, 20] aims to exploit social relations to enhance the recommender system. According to the social science theories [1, 3, 29], user decisions are influenced by their social relations, leading to similar preferences among social neighbors. Following this assumption, SoRec [28] jointly factorizes the user-item matrix and the user-user social relation matrix by sharing the same user preference latent factor. STE [27] learns user preferences by linearly combing the preference latent factor of the user and his/her social neighbors. SocialMF [17] forces the user preference latent factor to be similar to that of his/her social neighbors by adding regularization to the user-item matrix factorization. These works only leverage first-order social neighbors for recommendation and ignore the fact that the social influence could diffuse recursively through social networks.

To model the high-order social influence, graph neural networks (GNNs) [21] are introduced to social recommendation due to their superiority in learning the graph structure. GraphRec [10] applies GNNs to capture the heterogeneous graph information from the user-item interactions and social relations. DiffNet [44] and its extension DiffNet++ [43] develop a layer-wise influence propagation structure to model the recursive social diffusion in social recommendation. These works model user preferences with pairwise relations and fail to capture the complex multiplex user relation patterns (*i.e.,* user-friend-item). MHCN [48] constructs hypergraphs by unifying nodes that form specific triangular relations and applies hypergraph neural network [11, 46] to model user preferences with hypergraphs. Despite effectiveness, previous works treat social relations as static information to enhance the learning of user preferences, while ignoring the dynamic characteristic of user preferences and failing to dynamically choose helpful social information for the learning of user preferences.

# 3 DEFINITION AND PRELIMINARY

In this section, we formulate the problem of multi-interest Multi-round Conversational Recommendation (MMCR) [49].

Specifically, we define the set of items $\mathcal{V}$, attributes $\mathcal{P}$, and attributes types $C$. Each item $v \in \mathcal{V}$ is associated with a set of attributes $\mathcal{P}_v \subseteq \mathcal{P}$ and each attribute $p$ has its corresponding type $c_p \in C$. In each episode, there exists an item set $\mathcal{V}_u$ that is acceptable for the user. Then CRS screens out candidate items $\mathcal{V}_{cand} \subseteq \mathcal{V}$ that contains the user-preferred attribute $p_0$ and candidate attributes $\mathcal{P}_{cand} \subseteq \mathcal{P}$ that are associated to the candidate items. Then in each turn $t$ ($t = 1, 2, \cdots, T$; $T$ is the max turn of the session), the CRS can either *ask* $K_p$ attribute $\tilde{\mathcal{P}}_c \in \mathcal{P}_{cand}$ corresponding to the same attribute type $c$, or *recommend* $K_v$ items $\tilde{\mathcal{V}} \in \mathcal{V}_{cand}$:

- If the CRS chooses to *ask*, the user gives feedback according to whether $\mathcal{P}_c^*$ is associated with one of the items in the target item set $\mathcal{V}_u$.
- If the CRS chooses to *recommend*, the user chooses to accept or not according to whether one of the items in the target item set $\mathcal{V}_u$ is listed in the recommended items $\tilde{\mathcal{V}}$.

The session of MMCR terminates if the user accepts the recommended items or leaves impatiently when the max turn accesses.

# 4 FRAMEWORK

In this section, we propose a novel Multi-view Hypergraph Contrastive Policy Learning (MHCPL) illustrated in Figure 2 that learns user preferences from the hypergraph integrating the interactive conversation, item knowledge and social information, and accordingly chooses actions. The Markov Decision Process (MDP) [35] formulation of our framework contains four components: multi-view user preference modeling, action, transition, and reward.

## 4.1 Multi-view User Preference Modeling

We first encode the state $s_t$, which contains the interactive conversation information $\mathcal{I}_u$ between the user and CRS, and the social information $\mathcal{F}_u$ that helps learn user preferences:

$$s_t = [\mathcal{I}_u^{(t)}, \mathcal{F}_u^{(t)}], \tag{1}$$

where $\mathcal{I}_u^{(t)} = [\mathcal{P}_{acc}^{(t)}, \mathcal{P}_{rej}^{(t)}, \mathcal{V}_{rej}^{(t)}, \mathcal{P}_{cand}^{(t)}, \mathcal{V}_{cand}^{(t)}]$ records the interactive history, and $\mathcal{F}_u^{(t)}$ denotes user's friends who have preferred items satisfying the interactive history $\mathcal{I}_u^{(t)}$, which is updated by:

$$\mathcal{F}_u^{(t)} = \{f \mid f \in \mathcal{F}_u \text{ and } \mathcal{V}_f^{(t)} \neq \emptyset\}, \tag{2}$$

where $\mathcal{F}_u$ denotes the friends of the user, $\mathcal{V}_f^{(t)} = \mathcal{V}_f \cap \mathcal{V}_{cand}^{(t)}$ indicates the set of items that are acceptable for the friend $f$ and satisfy the interactive history. To this end, we build a dynamic hypergraph that integrates the interactive conversation, item knowledge, and social information to learn the user preference representation. Moreover, we develop a hypergraph-based state encoder to learn user preferences with multiplex relations from different views.

## 4.2 Action

According to the state $s_t$, the CRS agent chooses an action $a_t$ from the action space $\mathcal{A}_t$. The action space $\mathcal{A}_t$ contains candidate attributes $\mathcal{P}_{cand}^{(t)}$ and candidate items $\mathcal{V}_{cand}^{(t)}$, which are updated by:

$$\mathcal{V}_{\text{cand}}^{(t)} = \Big\{v \mid v \in \mathcal{V}_{p_0} - \mathcal{V}_{rej}^{(t)} \text{ and } \mathcal{P}_v \cap \mathcal{P}_{acc}^{(t)} \neq \emptyset \text{ and } \mathcal{P}_v \cap \mathcal{P}_{rej}^{(t)} = \emptyset\Big\}, \tag{3}$$

$$\mathcal{P}_{cand}^{(t)} = \mathcal{P}_{\mathcal{V}_{cand}^{(t)}} - \mathcal{P}_{acc}^{(t)} \cup \mathcal{P}_{rej}^{(t)}, \tag{4}$$

where $\mathcal{V}_{p_0}$ denotes the items that satisfy the initial attribute $p_0$ of the user and $\mathcal{P}_{\mathcal{V}_{cand}^{(t)}}$ indicates attributes that belong to at least one of the candidate items $\mathcal{V}_{cand}^{(t)}$. When the CRS agent chooses to recommend, the agent chooses the top-K items $\tilde{\mathcal{V}}^{(t)}$ from $\mathcal{A}_t$. If the CRS agent decides to consult, the agent chooses $K_a$ attributes $\tilde{\mathcal{P}}_c^{(t)}$ that belong to the same attribute type $c$ from $\mathcal{A}_t$.

## 4.3 Transition

After the CRS agent chooses the action $a_t$, the state $s_t$ will transition to the next state $s_{t+1}$. Specifically, if the agent chooses to consult, the attribute the user accepts and rejects in the current turn can be defined as $\mathcal{P}_{cur\_acc}^{(t)}$ and $\mathcal{P}_{cur\_rej}^{(t)}$. Then the state is updated by $\mathcal{P}_{acc}^{(t+1)} = \mathcal{P}_{acc}^{(t)} \cup \mathcal{P}_{cur\_acc}^{(t)}$ and $\mathcal{P}_{rej}^{(t+1)} = \mathcal{P}_{rej}^{(t)} \cup \mathcal{P}_{cur\_rej}^{(t)}$. When the agent chooses to recommend items $\tilde{\mathcal{V}}^{(t)}$ and the user rejects all the items, the state is updated by $\mathcal{V}_{rej}^{(t+1)} = \mathcal{V}_{rej}^{(t)} \cup \tilde{\mathcal{V}}^{(t)}$. Otherwise, this session ends with success.

## 4.4 Reward

In this work, we design five kinds of rewards following previous works [49]: (1) $r_{rec\_suc}$, a strong reward when recommending successfully; (2) $r_{rec\_fail}$, a weak penalty when the user rejects the recommended items; (3) $r_{ask\_suc}$, a weak reward when the user accepts the asked attributes; (4) $r_{ask\_fail}$, a weak penalty when the user rejects the asked attributes; (5) $r_{quit}$, a strong penalty when the session quits without success. The reward on the multi-choice question is designed as $r_t = \sum_{\mathcal{P}_{cur-acc}^{(t)}} r_{ask-suc} + \sum_{\mathcal{P}_{currej}^{(t)}} r_{ask-rej}$.

# 5 MULTI-VIEW HYPERGRAPH CONTRASTIVE POLICY LEARNING

In this section, we detail the design of the Multi-view Hypergraph Contrastive Policy Learning (MHCPL). As shown in Figure 2, to model the dynamic user preferences, we build a hypergraph with three types of multiplex relations from different views to integrate information from the interactive conversation, item knowledge, and social information. To comprehensively learn user preferences, we develop a hypergraph-based state encoder, that captures the graph structure and the sequential modeling in the dynamic hypergraph, and propose a cross-view contrastive learning module to maintain the inherent characteristics and the correlation of user preferences from different views. Moreover, we develop an action decision policy to decide the next action based on the learned dynamic user preferences.

## 5.1 Multi-view Hypergraph Construction

As illustrated in Figure 2, we model the user preference at timestep $t$ with a multi-view dynamic hypergraph which can be formulated as $\mathcal{G}_u^{(t)} = (\mathcal{N}^{(t)}, \mathcal{H}^{(t)}, \mathrm{A}^{(t)})$, including: (1) a node set $\mathcal{N}^{(t)} = \{u\} \cup \mathcal{P}_{rej}^{(t)} \cup \mathcal{P}_{acc}^{(t)} \cup \mathcal{F}_u^{(t)} \cup \mathcal{V}_{p_0}$, where $\mathcal{V}_{p_0}$ indicates the items satisfying the initial attribute $p_0$ of the user $u$, and $\mathcal{F}_u^{(t)}$ denotes the filtered friends that have preferring items that satisfy the interactive history $\mathcal{I}_u^{(t)}$; (2) a hyperedge set $\mathcal{H}^{(t)} = \mathcal{H}_{like}^{(t)} \cup \mathcal{H}_{dis}^{(t)} \cup \mathcal{H}_f^{(t)}$, where $\mathcal{H}_{like}^{(t)}$ denotes the user like items that satisfy the attribute (*Like View*), $\mathcal{H}_{dis}^{(t)}$ indicates the user dislike items that satisfy the attribute (*Dislike View*), and $\mathcal{H}_f^{(t)}$ denotes the user shares preferences to the items with the friend (*Social View*). Each hyperedge $h \in \mathcal{H}^{(t)}$ is corresponding to an attribute $p_h$ or friend $f_h$; (3) a $|\mathcal{N}^{(t)}| \times |\mathcal{H}^{(t)}|$ adjacent matrix $\mathrm{A}^{(t)}$ which denotes the weighted edge between each node and hyperedge, with entries denoted as:

$$A_{i,j}^{(t)} = \begin{cases} 1, & \text{if } n_i = u, h_j \in \mathcal{H}_{like}^{(t)} \cup \mathcal{H}_f^{(t)} \\ -1, & \text{if } n_i = u, h_j \in \mathcal{H}_{dis}^{(t)} \\ \frac{1}{|\mathcal{V}_{h_j}^{(t)}|}, & \text{if } n_i \in \mathcal{V}_{h_j}^{(t)}, h_j \in \mathcal{H}_{like}^{(t)} \cup \mathcal{H}_{dis}^{(t)} \\ \frac{1}{|\mathcal{V}_{h_j}^{(t)}|}, & \text{if } n_i \in \mathcal{V}_{h_j}^{(t)}, h_j \in \mathcal{H}_f^{(t)} \\ 1, & \text{if } h_j \in \mathcal{H}_{like}^{(t)} \cup \mathcal{H}_{dis}^{(t)}, n_i = p_{h_j} \\ 1, & \text{if } h_j \in \mathcal{H}_f^{(t)}, n_i = f_{h_j} \\ 0, & \text{otherwise} \end{cases}, \tag{5}$$

where $\mathcal{V}_{h_j}^{(t)}$ denotes items connected to the hyperedge $h_j$. Specifically, when $h_j \in \mathcal{H}_{like}^{(t)} \cup \mathcal{H}_{dis}^{(t)}$, $\mathcal{V}_{h_j}^{(t)}$ means items that satisfy the corresponding attribute $p_{h_j}$. And when $h_j \in \mathcal{H}_f^{(t)}$, it means the friend $f_{h_j}$'s acceptable items that satisfy the interactive history $\mathcal{I}_u^{(t)}$. We filter out the noise in friends' acceptable items with the interactive history to help learn the user's current dynamic preferences.

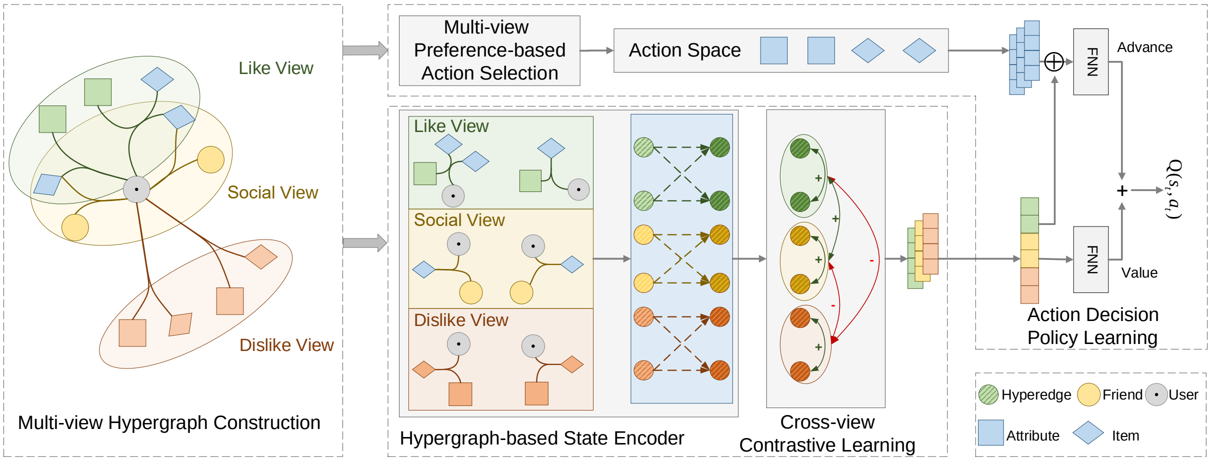


**Figure 2: The overview of Multi-view Hypergraph Contrastive Policy Learning (MHCPL). It mainly contains four modules: (a) Multi-view Hypergraph Construction, which dynamically captures multiplex relations from three views. (b) Hypergraph-based State Encoder, which captures the graph structure and sequential modeling in the dynamic hypergraph. (c) Cross-view Contrastive Learning, which maintains the inherent characteristics and correlations of user preferences from different views, and (d) Action Decision Policy Learning to decide the next action based on the learned dynamic user preferences. (Best view in color.)**

## 5.2 Hypergraph-based State Encoder

*5.2.1 Hypergraph Message Passing Paradigm.* Motivated by the strength of hypergraph [11, 46] for generalizing the concept of edge to connect more than two nodes, we endow our MHCPL to capture multiplex relations under a hypergraph message passing architecture, where the hyperedges are treated as intermediate hubs for message passing across different nodes without the hop limitation. The formal representation of our hypergraph message passing is formulated as:

$$\mathbf{\Gamma} = \mathrm{ReLU}(\mathbf{A} \cdot \mathbf{H}) = \mathrm{ReLU}\left(\mathbf{A} \cdot \mathbf{A}^\top \cdot \mathbf{E}\right), \tag{6}$$

where $\mathbf{E} \in \mathbb{R}^{|\mathcal{N}^{(t)}| \times d}$ denotes the initial embedding of nodes $\mathcal{N}^{(t)}$ in the hypergraph, $\mathbf{H} \in \mathbb{R}^{|\mathcal{H}^{(t)}| \times d}$ indicates the hyperedge representations aggregated from the node representations, and $\mathrm{ReLU}(\cdot)$ denotes the LeakyReLU mapping. $\mathbf{\Gamma}$ denotes the hyper embedding of the nodes in the hypergraph representation space. With the help of hypergraph message passing, our MHCPL is capable to capture the multiplex collaborative relations that specify the attribute/friend that motivates/discourages the user's interest in the items.

*5.2.2 Hierarchical Hypergraph State Encoder.* During the conversation, the hyperedges are successively generated when the user accepts or rejects the asked attribute. Moreover, the higher-level interactions between different hyperedges are also important in learning user preferences. Although the aforementioned hypergraph message passing paradigm is capable to capture the multiplex relations, it fails to model sequential information and hyperedge-wise feature interactions. Inspired by the success of the Transformer encoder [37] in capturing sequential information and feature interactions, we employ the Transformer encoder to realize high-level hyperedge-wise message passing. Specifically, with the representation of hyperedges $\boldsymbol{H}$ that aggregate information from neighbor nodes, higher-level hypergraph layers further pass messages through the interactions between hyperedges under the same view as:

$$\bar{\mathrm{H}} = \psi^l(\boldsymbol{H}), \boldsymbol{H} = \mathrm{A}^\top \mathrm{E}, \tag{7}$$

where $\mathbf{E} \in \mathbb{R}^{|\mathcal{N}^{(t)}| \times d}$ denotes the initial embedding of nodes $\mathcal{N}^{(t)}$ in the hypergraph. $\psi^l(\cdot)$ indicates the high-level hypergraph layers. $l$ denotes the layer number of high-level hypergraph layers. Hyperedges $\mathcal{H}_o^{(t)}$ of the same view $o \in \{like, dis, f\}$ are successively connected according to their generation order in the interactive conversation. To realize this, we apply the Transformer encoder $MHSA_o(\cdot)$ on hyperedges $\mathcal{H}_o^{(t)}$ of each view $o$ as:

$$\psi(\boldsymbol{H}_o^l) = \mathrm{MHSA}_o(\boldsymbol{H}_o^{l-1}, \boldsymbol{H}_o^{l-1}, \boldsymbol{H}_o^{l-1}). \tag{8}$$

After the high-level hyperedge message passing, we aggregate the information from hyperedges to refine the node representations as:

$$\Gamma_l = \mathrm{ReLU}(\mathrm{A} \cdot \bar{\mathrm{H}}) = \mathrm{ReLU}\left(\mathcal{H} \cdot \psi^l\left(\mathrm{A}^\top \cdot \mathrm{E}\right)\right), \tag{9}$$

where $\psi^l$ denotes $l$ high-level hypergraph layers. The hyper representation from different layers of the user is summed to get the representation of state $s_t$:

$$\mathbf{q}_t = \sum_l \Gamma_l(u) \tag{10}$$

## 5.3 Cross-view Contrastive Learning

Different types of multiplex relations present user preferences from various views (*i.e., Like View*, *Dislike View*, *Social View*). Actually, it is still non-trivial to sufficiently integrate user preferences from different views, since it might obscure the inherent characteristics of preference distributions from different views and the correlation between them. Specifically, the user preferences from the same view should be more similar than those from different views. Also, user preferences from *Like View* should be similar to *Social View* while different from *Dislike View*. To capture these two correlations and better integrate user preferences from different views, we develop cross-view contrastive learning based on InfoNCE[30] as:

$$\mathcal{L}^{SSL} = -\sum_{o}\sum_{i\in\mathcal{H}_o}\log\frac{\sum_{i^+\in\mathcal{H}_o}\exp(s(\mathbf{H}_i,\mathbf{H}_{i^+})/\tau)}{\underbrace{\sum_{i^+\in\mathcal{H}_o}\exp(s(\mathbf{H}_i,\mathbf{H}_{i^+})/\tau)}_{\text{positive pairs}}+\underbrace{\sum_{i^-\in\mathcal{H}-\mathcal{H}_o}\exp(s(\mathbf{H}_i,\mathbf{H}_{i^-})/\tau)}_{\text{negative pairs}}} -\sum_{o}\sum_{i\in\mathcal{H}_o}\log\frac{\sum_{i^+\in\mathcal{H}_{o^+}}\exp(s(\mathbf{H}_i,\mathbf{H}_{i^+})/\tau)}{\underbrace{\sum_{i^+\in\mathcal{H}_{o^+}}\exp(s(\mathbf{H}_i,\mathbf{H}_{i^+})/\tau)}_{\text{positive pairs}}+\underbrace{\sum_{i^-\in\mathcal{H}_{o^-}}\exp(s(\mathbf{H}_i,\mathbf{H}_{i^-})/\tau)}_{\text{negative pairs}}}, \quad (11)$$

where $o\in\{like, dis, f\}$ denotes three views, $\mathcal{H}=\mathcal{H}_{like}\cup\mathcal{H}_{dis}\cup\mathcal{H}_f$ indicates the set of hyperedges, $\mathbf{H}$ denotes the representations of hyperedges, and $s(\cdot)$ is the cosine similarity function. In Eq.11, the first term is designed to maintain the intrinsic characteristics of user preferences from each view, which treats the hyperedges of the same view as positive pairs, while the different-view hyperedges as negative pairs. The second term of Eq.11 is designed to maintain the correlation of user preferences from different views, where the hyperedges in $\mathcal{H}_{like}$ and $\mathcal{H}_f$ are treated as positive pairs to each other, while treated as negative pairs with the hyperedges in $\mathcal{H}_{dis}$.

## 5.4 Action Decision Policy Learning

A large action search space reduces the efficiency of policy learning. Following [9, 49], we select top-$\mathcal{K}_v$ candidate items and top-$\mathcal{K}_p$ candidate attributes to form the action space $\mathcal{A}_t$. To this end, we develop a multi-view action selection strategy, which selects items/attributes according to user preferences from three views. Specifically, we rank them based on their similarity with accepted attributes in the interactive conversation and filtered social information and their difference with rejected attributes in the interactive conversation as:

$$w_v^{(t)}=\sigma\left(\mathbf{e}_u^T\mathbf{e}_v+\sum_{p\in\mathcal{P}_{acc}^{(t)}}\mathbf{e}_v^T\mathbf{e}_p+\sum_{f\in\mathcal{F}_u^{(t)}}\mathbf{e}_v^T\tilde{\mathbf{e}}_f-\sum_{p\in\mathcal{P}_{rej}^{\prime(t)}}\mathbf{e}_v^T\mathbf{e}_p\right), \quad (12)$$

$$w_p^{(t)}=\sigma\left(\mathbf{e}_u^T\mathbf{e}_p+\sum_{p'\in\mathcal{P}_{acc}^{(t)}}\mathbf{e}_p^T\mathbf{e}_{p'}+\sum_{f\in\mathcal{F}_u^{(t)}}\mathbf{e}_p^T\tilde{\mathbf{e}}_f-\sum_{p'\in\mathcal{P}_{rej}^{\prime(t)}}\mathbf{e}_p^T\mathbf{e}_{p'}\right), \quad (13)$$

where $\mathbf{e}_u$, $\mathbf{e}_v$, $\mathbf{e}_p$ and $\mathbf{e}_f$ are embeddings of the user, item, attribute, and friend. $\tilde{\mathbf{e}}_f=\sum_{v'\in\mathcal{V}_f^{(t)}}\mathbf{e}_v$ represents friend preferences that satisfy the interactive history, $\sigma(\cdot)$ denotes the sigmoid function.

With the action space $\mathcal{A}_t$ and the state representation $\mathbf{q}_t$, we introduce the dueling Q-networks [40] to determine the next action and calculate the Q-value as:

$$Q(s_t,a_t)=f_{\theta_V}(\mathbf{q}_t)+f_{\theta_A}(\mathbf{q}_t,a_t), \quad (14)$$

where the value function $f_{\theta_V}(\cdot)$ and the advantage function $f_{\theta_A}(\cdot)$ are two separate multi-layer perceptions with $\theta_V$ and $\theta_A$ denote the parameters, respectively. The optimal Q-function $Q^*(\cdot)$, which has the maximum expected reward achievable by the optimal policy $\pi^*$, follows the Bellman equation [2] as:

$$Q^*(s_t,a_t)=\mathbb{E}_{s_{t+1}}\left[r_t+\gamma\max_{a_{t+1}\in\mathcal{A}_{t+1}}Q^*(s_{t+1},a_{t+1}\mid s_t,a_t)\right], \quad (15)$$

where $\gamma$ denotes the discount factor for the delayed rewards.

In each turn, the CRS agent will get the reward $r_t$, and we can update the state $s_{t+1}$ and the action space $\mathcal{A}_{t+1}$ according to the user's feedback. Following Deng *et al.* [9], we define a replay buffer $\mathcal{D}$ to store the experience $(s_t,a_t,r_t,s_{t+1},\mathcal{A}_{t+1})$. For training of the DQN, we sample mini-batches from the buffer and minimize the following loss:

$$\mathcal{L}^{DQN}=\mathbb{E}_{(s_a,a_t,r_t,s_{t+1},\mathcal{A}_{t+1})\sim\mathcal{D}}\left[\left(y_t-Q\left(s_t,a_t;\theta_Q,\theta_S\right)\right)^2\right], \quad (16)$$

where $\theta_S$ is the set of parameters in the module for hypergraph-based representation learning, $\theta_Q=\{\theta_V,\theta_A\}$, and $y_t$ is the target value based on the currently optimal $Q^*$:

$$y_t=r_t+\gamma\max_{a_{t+1}\in\mathcal{A}_{t+1}}Q\left(s_{t+1},a_{t+1};\theta_Q,\theta_S\right). \quad (17)$$

To deal with the overestimation bias in the original DQN, we apply the double DQN [36], which copies a target network $Q^{'}$ as a periodic from the online network to train the model. During training, the action decision policy learning in Eq.16, and the cross-view contrastive learning in Eq.11 are alternatively optimize.

# 6 EXPERIMENTS

To fully demonstrate the superiority of our proposed MHCPL, we conduct experiments on two public datasets to explore the following questions:

- **RQ1:** How does MHCPL perform compared with the state-of-the-art methods?
- **RQ2:** How do different components (social influence, hypergraph based state encoder, and cross-view contrastive learning) affect the results of MHCPL?
- **RQ3:** How do parameters (the layer number of Hypergraph based State Encoder) influence the results of MHCPL?
- **RQ4:** Can our MHCPL effectively leverage the interactive conversation, item knowledge, and social influence to learn the dynamic user preferences?

## 6.1 Datasets

To evaluate the proposed method, we adapt two existing MCR benchmark datasets, named Yelp and LastFM. The statistics of these datasets are presented in Table 1.

- **LastFM** [22]: LastFM dataset is the music listening dataset collected from Last.fm online music systems. As Zhang *et al.* [49], We define the 33 coarse-grained groups as attribute types for the 8,438 attributes.
- **Yelp** [22]: Yelp dataset is adopted from the 2018 edition of the Yelp challenge. Following Zhang *et al.* [49], we define the 29 first-layer categories as attribute types, and 590 second-layer categories as attributes.

Following Zhang *et al.* [49], we sample two items with partially overlapped attributes as the user's acceptable items for each conversation episode.

| Dataset | Yelp | LastFM |
|---|---|---|
| Users | 27,675 | 1,801 |
| Items | 70,311 | 7,432 |
| Attributes | 590 | 8,438 |
| Attribute types | 29 | 34 |
| User-Item | 1,368,606 | 76,693 |
| User-User | 688,209 | 23,958 |
| Item-Attribute | 477,012 | 94,446 |

**Table 1: Statistics of two utilized datasets**

## 6.2 Experiments Setup

*6.2.1 User Simulator.* MMCR is a system that is trained and evaluated based on interactive conversations with users. Following the user simulator adopted in [49], we simulate a interactive session for each user-item set interaction pair $(u, \mathcal{V}_u)$. Each item in the item set $v \in \mathcal{V}_u$ is treated as an acceptable item for the user. Each session is initialized with a user $u$ specifying an attribute $p_0 \in \mathcal{P}_{joint}$, where $\mathcal{P}_{joint}$ is the set of attributes that are shared by the items in $\mathcal{V}_u$. Then the session follows the process of "System Ask or Recommend, User response" [49] as described in Section 3.

*6.2.2 Baselines.* To demonstrate the effectiveness of the proposed MHCPL, the state-of-the-art methods are chosen for comparison as follows:

- **Max Entropy.** This method employs a rule-based strategy to ask and recommend. It chooses to select an attribute with maximum entropy based on the current state, or recommends the top-ranked item with certain probabilities [22].
- **Greedy[7].** This method only makes item recommendations and updates the model based on the feedback. It keeps recommending items until the successful recommendation is made or the pre-defined round is reached.
- **CRM[34].** A reinforcement learning-based method that records the users' preferences into a belief tracker and learns the policy deciding when and which attributes to ask based on the belief tracker.
- **EAR[22].** This method proposes a three-stage solution to enhance the interaction between the conversational component and the recommendation component.
- **SCPR[24].** This method learns user preferences by reasoning the path on the user-item-attribute graph via the user's feedback and accordingly chooses actions.
- **UNICORN[9].** This work builds a weighted graph to model the dynamic relationships between the user and the candidate action space and proposes a graph-based Markov Decision Process (MDP) environment to learn the user's dynamic preferences and chooses actions from the candidate action space.
- **MCMIPL[49].** This approach proposes a multi-interest policy learning framework that captures the multiple interests of the user to decide the next action.
- **S*-UNICORN and S*-MCMIPL.** For a more comprehensive and fair performance comparison, we adapt UNICORN and MCMIPL by timely selecting helpful social information and incorporating it into the weighted graph of the model. We name the two adapted methods S*-UNICORN and S*-MCMIPL.

*6.2.3 Parameters Setting.* Following [49], we recommend top $K = 10$ items or ask $K_a = 2$ attributes in each turn. We employ the Adam optimizer with a learning rate of $1e-4$. Discount factor $\gamma$ is set to be 0.999. Following [9], we adopt TransE [4] via OpenKE [14] to pretrain the node embeddings with 64 dimensions in the constructed KG with the training set. We make use of Nvidia Titan RTX graphics cards equipped with AMD r9-5900x CPU (32GB Memory). For the action space, we select $K_p = 10$ attributes and $K_v = 10$ items. To maintain a fair comparison, we adopt the same reward settings as previous works [9, 22, 24, 49]: $r_{rec\_suc} = 1, r_{rec\_fail} = -0.1, r_{ask\_suc} = 0.01, r_{ask\_fail} = 0.1, r_{quit} = -0.3$. For MHCPL, we select the number of layers from 1, 2, 3, 4.

*6.2.4 Evaluation Metrics.* Following previous works [9, 22, 24], we adopt success rate (SR@t) to measure the cumulative ratio of successful recommendations by the turn t, average turns (AT) to evaluate the average number of turns for all sessions, and hDCG@(T, K) to additionally evaluate the ranking performance of recommendations. Therefore, the higher SR@t and hDCG@(T, K) indicate better performance, while the lower AT means an overall higher efficiency.

## 6.3 Performance Comparison (RQ1)

*6.3.1 Overall Performance.* The comparison experimental results of the baseline models and our models are shown in Table 2. Based on the comparison in the table, we can summarize our observations as follows:

- **Our proposed MHCPL achieves the best performance.** MHCPL significantly outperforms all the baselines on the metrics of SR@15, AT and hDCG by over 4.47%, 5.23% and 13.48%, respectively. We attribute the improvements to the following reasons: 1) The proposed dynamic multi-view hypergraph could effectively capture multiplex relations from three views. And the proposed hierarchical hypergraph neural network is able to well learn dynamic user preferences by integrating the information of graph structure and sequential modeling from the dynamic multi-view hypergraph; 2) MHCPL timely selects helpful social information and effectively integrates the interactive conversation, item knowledge, and social influence for better dynamic user preference learning; 3) MHCPL designs a cross-view contrastive learning method to help maintain the inherent characteristics and the correlations of user preferences from different views.
- **The learning of the dynamic user preferences is crucial for conversational recommendation.** The graph-based methods (MHCPL, MCMIPL, UNICORN, SCPR) outperforms the factorization-based methods (EAR, CRM) since they learn user preferences from the collaborative information in the graph. MCMIPL achieves the best performance among the graph-base baselines since it further considers the multiple interests of the user preferences. Our proposed MHCPL further outperforms these methods since we leverage multiplex relations to integrate interactive conversation, item knowledge, and social influence to help learn the dynamic user preferences.
- **Social influence is effective in helping learn dynamic user preferences for conversational recommendation when well**

| Models | Yelp | | | | | LastFM | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | SR@5 | SR@10 | SR@15 | AT | hDCG | SR@5 | SR@10 | SR@15 | AT | hDCG |
| Abs Greedy | 0.078 | 0.124 | 0.150 | 13.65 | 0.065 | 0.292 | 0.436 | 0.512 | 10.10 | 0.237 |
| Max Entropy | 0.046 | 0.200 | 0.390 | 12.97 | 0.117 | 0.280 | 0.560 | 0.680 | 9.34 | 0.263 |
| CRM | 0.026 | 0.100 | 0.188 | 13.99 | 0.059 | 0.092 | 0.240 | 0.372 | 12.56 | 0.130 |
| EAR | 0.120 | 0.198 | 0.240 | 12.91 | 0.094 | 0.298 | 0.436 | 0.508 | 10.08 | 0.237 |
| SCPR | 0.146 | 0.188 | 0.436 | 12.29 | 0.169 | 0.322 | 0.630 | 0.764 | 8.47 | 0.322 |
| UNICORN | 0.200 | 0.338 | 0.430 | 11.33 | 0.175 | 0.444 | 0.774 | 0.846 | 7.10 | 0.348 |
| MCMIPL | 0.162 | 0.366 | 0.522 | 11.25 | 0.184 | 0.448 | 0.809 | 0.884 | 6.87 | 0.353 |
| S*-UNICORN | 0.120 | 0.478 | 0.696 | 10.59 | 0.223 | 0.412 | 0.850 | 0.912 | 6.69 | 0.363 |
| S*-MCMIPL | 0.126 | 0.490 | 0.722 | 10.51 | 0.230 | 0.442 | 0.872 | 0.940 | 6.43 | 0.368 |
| MHCPL | 0.142 | **0.592** | **0.854** | **9.96** | **0.261** | **0.470** | **0.938** | **0.982** | **5.87** | **0.427** |
| Improv. | - | 20.82% | 18.28% | 5.23% | 13.48% | 4.91% | 7.57% | 4.47% | 8.71% | 16.03% |

**Table 2: Performance comparison of different models on the two datasets. The bold number represents the improvement of our model over baselines is statistically significant with p-value $< 0.01$. hDCG stands for hDCG@$(15, 10)$.**

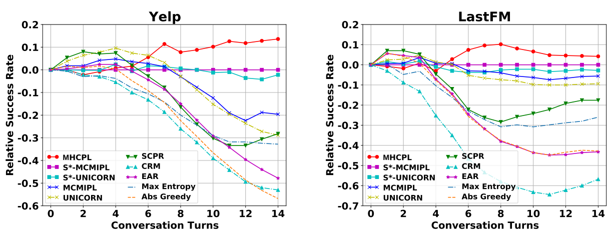


**Figure 3: Comparisons at Different Conversation Turns.**

| Models | Yelp | | | LastFM | | |
|---|---|---|---|---|---|---|
| | SR@15 | AT | hDCG | SR@15 | AT | hDCG |
| Ours | **0.854** | **9.96** | **0.261** | **0.982** | **5.87** | **0.427** |
| -w/o social | 0.592 | 10.80 | 0.208 | 0.908 | 6.63 | 0.365 |
| -w/o hypergraph | 0.726 | 10.68 | 0.346 | 0.938 | 6.58 | 0.382 |
| -w/o contrastive | 0.762 | 10.37 | 0.237 | 0.962 | 6.17 | 0.403 |

**Table 3: Results of the Ablation Study.**

**filtered.** The socially adapted methods (*i.e.,* S*-UNICORN and S*-MCMIPL) outperform their original versions in the final performances. We attribute this to the reason that social influence is an important factor that affects user preferences and could help learn dynamic user preferences with friends' preferences that satisfy the interactive conversation. But the socially adapted methods perform worse than their original version in the early turns (*e.g.,* SR@5). This happens because the information in the interactive conversation is not sufficient to filter out the noise from the social information in the early turn of the conversation.

*6.3.2 Comparison at Different Conversation Turns.* Besides the performance in the final turn, we also present success rates at different turns in Figure 3. In order to better observe the differences among different models, we use the relative success rate compared with the most competitive baseline S*-MCMIPL, where the blue line of S*-MCMIPL is set to zero in the figures. From the Figure 3, we following observations:

- The proposed MHCPL outperforms these baseline methods across all the datasets and almost all the turns in the conversational recommendation. This is because our proposed MHCPL could better learn dynamic user preferences with multiplex relations that integrate interactive conversation, item knowledge, and social influence.
- The recommendation success rate of the proposed socially-aware methods (*i.e.,* MHCPL, S*-MCMIPL, and S*-UNICORN) could not surpass all the baselines in the early turns of the conversational recommendation, especially on the dataset Yelp with a larger candidate space of items and attributes. This is because the information in the interactive conversation is not sufficient to filter out the noise from the social information at the early turn of the conversation. Furthermore, socially-aware methods prefer to ask rather than recommend in the early turns when the user's preference is not certain enough. This will effectively reduce the action space and better learn user preferences, but lead to a lower recommendation success rate in the early turns.

## 6.4 Ablation Studies (RQ2)

To investigate the underline mechanism of our proposed MHCPL, we conduct a series of ablation experiments on the Yelp and LastFM datasets with three ablated methods including: MHCPL$_{w/o\ social}$ that ablates the social influence, MHCPL$_{w/o\ hypergraph}$ that replaces the hypergraph neural networks with graph neural networks, and MHCPL$_{w/o\ contrastive}$ that ablates the cross-view contrastive learning. From results are shown in Table 3, we have the following observations:

- MHCPL$_{w/o\ social}$ is the least competitive. This demonstrates the importance of social influence in alleviating the data sparsity problem and helping learn dynamic user preferences. And it is effective to accordingly choose helpful social information based on interactive conversation. MHCPL$_{w/o\ social}$ still outperforms all the baselines that ignore the social information in Table 2, which proves the effectiveness of our multi-view hypergraph contrastive policy learning strategy in learning dynamic user preferences with multiplex relations.

- MHCPL outperforms MHCPL$_{w/o\ hypergraph}$. We contribute this to the importance of multiplex relations in learning dynamic user preferences. This also proves the effectiveness of our proposed multi-view hypergraph-based state encoder in learning user preferences by integrating the information of graph structure and sequential modeling from the dynamic multi-view hypergraph.
- MHCPL outperforms MHCPL$_{w/o\ contrastive}$. This demonstrates the effectiveness of the cross-view contrastive learning module in helping maintain the inherent characteristics and correlations of user preferences from different views.

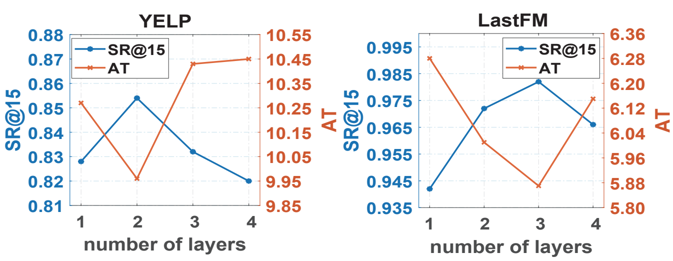


**Figure 4: Impact of Layer Number(L)**

## 6.5 Hyper-parameter Sensitivity Analysis (RQ3)

*6.5.1 Impact of Layer Number.* The hypergraph-based state encoder learns the dynamic user preference from the multiplex relations in the hypergraph. By stacking more layers, collaborative information from multi-hop neighbors is distilled. We investigate how the number of layers influences the performance of MHCPL. Specifically, we conduct experiments with layer number $L$ in the range $\{1, 2, 3, 4\}$, and the results are shown in Figure 4. There are some observations:

- Increasing the number of layers can improve the performance of our model. MHCPL-2 highly outperforms MHCPL-1. The reason is that MHCPL-1 only gains information from the one-hop neighbors and neglects high-order collaborative information.
- When increasing the layer of number, the performance does not always improve. MHCPL-3 outperforms MHCPL-4 on data LastFM. This can be attributed to the noise which increases along with the hop of neighbors.

## 6.6 Case Study (RQ4)

To show the effectiveness of our proposed MHCPL in leveraging multiplex relations to integrate interactive conversation, item knowledge, and social influence to learn dynamic user preferences, we present a case of conversational recommendation generated by our framework in Figure 5. As illustrated in the figure, by integrating the information from the interactive conversation, item knowledge, and social information with multiplex relations from different views,

| Acceptable items | Category | City | Star | Price range | Nightlife | Arts & Entertainment | Active Life |
|---|---|---|---|---|---|---|---|
| Item33510 | Restaurant | City8 | Star4 | Price2 | Nightlife | Arts & Entertainment Music Venues Festivals | - |
| Item20978 | Restaurant | City8 | Star3 | Price2 | Arcades | Arts & Entertainment | Active Life Bowling |

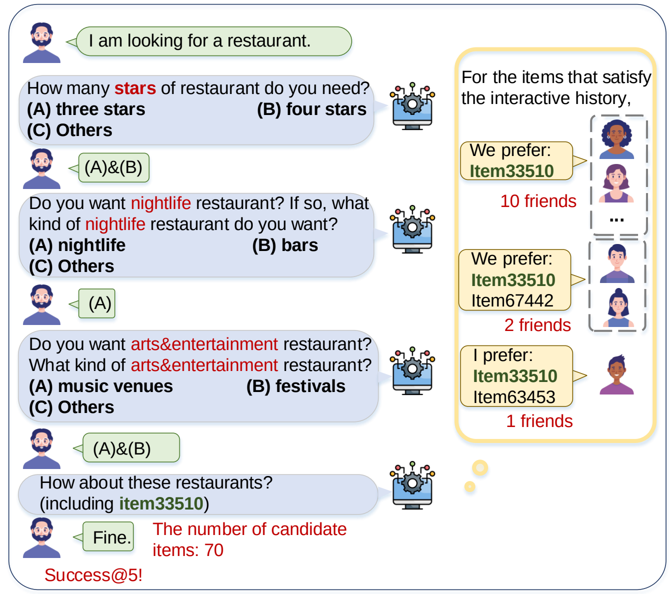


**Figure 5: A case of conversation recommendation generated by our proposed MHCPL.**

MHCPL is able to effectively ask attributes and recommend user-preferred items, reaching success in five turns. Furthermore, the social information selected according to the interactive conversation is helpful in learning dynamic user preferences. With the help of selected social information, MHCPL could accurately select the target item when the information from the interactive history is limited in distinguishing user preferences towards the seventy candidate items.

# 7 CONCLUSION

In this work, we explore multiplex relations to integrate interactive conversation, item knowledge, and social influence in helping learn the dynamic user preferences for conversational recommendation. We propose a novel hypergraph-based model, namely Multi-view Hypergraph Contrastive Policy Learning (MHCPL), which timely selects useful social information according to the interactive history and builds a dynamic hypergraph with three types of multiplex relations from different views. A hierarchical hypergraph neural network is proposed to learn user preferences by integrating information of the graph structure and sequential modeling from the dynamic multi-view hypergraph. Furthermore, a cross-view contrastive learning module is proposed with two terms to maintain the inherent characteristics and the correlations of user preferences from different views. Extensive experiments on two popular benchmarks demonstrate the superiority of our proposed method, as compared to the state-of-the-art baselines.

## ACKNOWLEDGMENTS

This work was supported in part by the National Natural Science Foundation of China under Grant No.62276110, Grant No.61772076, in part by CCF-AFSG Research Fund under Grant No.RF20210005, and in part by the fund of Joint Laboratory of HUST and Pingan Property & Casualty Research (HPL). The authors would also like to thank the anonymous reviewers for their comments on improving the quality of this paper.

## REFERENCES

[1] Aris Anagnostopoulos, Ravi Kumar, and Mohammad Mahdian. 2008. Influence and correlation in social networks. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. 7–15.

[2] Richard Bellman and Robert Kalaba. 1957. On the role of dynamic programming in statistical communication theory. *IRE Transactions on Information Theory* 3, 3 (1957), 197–203.

[3] Robert M Bond, Christopher J Fariss, Jason J Jones, Adam DI Kramer, Cameron Marlow, Jaime E Settle, and James H Fowler. 2012. A 61-million-person experiment in social influence and political mobilization. *Nature* 489, 7415 (2012), 295–298.

[4] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems* 26 (2013).

[5] 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)*. 1803–1813.

[6] 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 the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 139–148.

[7] Konstantina Christakopoulou, Filip Radlinski, and Katja Hofmann. 2016. Towards conversational recommender systems. , 815–824 pages.

[8] Robert B Cialdini and Noah J Goldstein. 2004. Social influence: Compliance and conformity. *Annual review of psychology* 55, 1 (2004), 591–621.

[9] 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.

[10] Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. 2019. Graph neural networks for social recommendation. In *The world wide web conference*. 417–426.

[11] Yifan Feng, Haoxuan You, Zizhao Zhang, Rongrong Ji, and Yue Gao. 2019. Hypergraph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 3558–3565.

[12] Maurizio Ferrari Dacrema, Paolo Cremonesi, and Dietmar Jannach. 2019. Are we really making much progress? A worrying analysis of recent neural recommendation approaches. In *Proceedings of the 13th ACM conference on recommender systems*. 101–109.

[13] Guibing Guo, Jie Zhang, and Neil Yorke-Smith. 2015. Trustsvd: Collaborative filtering with both the explicit and implicit influence of user trust and of item ratings. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 29.

[14] 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.

[15] Kaveh Hassani and Amir Hosein Khasahmadi. 2020. Contrastive multi-view representation learning on graphs. In *International conference on machine learning*. PMLR, 4116–4126.

[16] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*. 173–182.

[17] Mohsen Jamali and Martin Ester. 2009. Trustwalker: a random walk model for combining trust-based and item-based recommendation. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. 397–406.

[18] Meng Jiang, Peng Cui, Fei Wang, Wenwu Zhu, and Shiqiang Yang. 2014. Scalable recommendation with social contextual information. *IEEE Transactions on Knowledge and Data Engineering* 26, 11 (2014), 2789–2802.

[19] Dongyeop Kang, Anusha Balakrishnan, Pararth Shah, Paul Crook, Y Lan Boureau, and Jason Weston. 2020. Recommendation as a communication game: Self-Supervised Bot-Play for Goal-oriented Dialogue. In *2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019*. Association for Computational Linguistics, 1951–1961.

[20] Henry Kautz, Bart Selman, and Mehul Shah. 1997. Referral Web: combining social networks and collaborative filtering. *Commun. ACM* 40, 3 (1997), 63–65.

[21] Thomas N Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *ICLR*.

[22] 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 the 13th International Conference on Web Search and Data Mining*. 304–312.

[23] Wenqiang Lei, Xisen Jin, Min-Yen Kan, Zhaochun Ren, Xiangnan He, and Dawei Yin. 2018. Sequicity: Simplifying task-oriented dialogue systems with single sequence-to-sequence architectures. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 1437–1447.

[24] 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*. 2073–2083.

[25] Raymond Li, Samira Ebrahimi Kahou, Hannes Schulz, Vincent Michalski, Laurent Charlin, and Chris Pal. 2018. Towards deep conversational recommendations. *Advances in neural information processing systems* 31 (2018).

[26] 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 the 58th Annual Meeting of the Association for Computational Linguistics*. 1036–1049.

[27] Hao Ma, Irwin King, and Michael R Lyu. 2009. Learning to recommend with social trust ensemble. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*. 203–210.

[28] Nan Ma, Ee-Peng Lim, Viet-An Nguyen, Aixin Sun, and Haifeng Liu. 2009. Trust relationship prediction using online product review data. In *Proceedings of the 1st ACM international workshop on Complex networks meet information & knowledge management*. 47–54.

[29] Miller McPherson, Lynn Smith-Lovin, and James M Cook. 2001. Birds of a feather: Homophily in social networks. *Annual review of sociology* (2001), 415–444.

[30] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748* (2018).

[31] Bilih Priyogi. 2019. Preference elicitation strategy for conversational recommender system. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*. 824–825.

[32] Steffen Rendle. 2010. Factorization machines. In *2010 IEEE International conference on data mining*. IEEE, 995–1000.

[33] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*. 452–461.

[34] Yueming Sun and Yi Zhang. 2018. Conversational recommender system. In *The 41st international acm sigir conference on research & development in information retrieval*. 235–244.

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

[36] Hado Van Hasselt, Arthur Guez, and David Silver. 2016. Deep reinforcement learning with double q-learning. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 30.

[37] 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).

[38] Petar Velickovic, William Fedus, William L Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. 2019. Deep Graph Infomax. *ICLR (Poster)* (2019), 4.

[39] Yifan Wang, Suyao Tang, Yuntong Lei, Weiping Song, Sheng Wang, and Ming Zhang. 2020. DisenHAN: Disentangled Heterogeneous Graph Attention Network for Recommendation. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 1605–1614.

[40] Ziyu Wang, Tom Schaul, Matteo Hessel, Hado Hasselt, Marc Lanctot, and Nando Freitas. 2016. Dueling network architectures for deep reinforcement learning. In *International conference on machine learning*. PMLR, 1995–2003.

[41] Wei Wei, Sen Zhao, and Ding Zou. 2023. Recommendation System: A Survey and New Perspectives. *World Scientific Annual Review of Artificial Intelligence* (2023).

[42] Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. 2021. Self-supervised graph learning for recommendation. In *SIGIR*. 726–735.

[43] Le Wu, Junwei Li, Peijie Sun, Richang Hong, Yong Ge, and Meng Wang. 2020. Diffnet++: A neural influence and interest diffusion network for social recommendation. *IEEE Transactions on Knowledge and Data Engineering* (2020).

[44] Le Wu, Peijie Sun, Yanjie Fu, Richang Hong, Xiting Wang, and Meng Wang. 2019. A neural influence diffusion model for social recommendation. In *Proceedings of the 42nd international ACM SIGIR conference on research and development in*

*information retrieval*. 235–244.

[45] Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. 2019. Session-based recommendation with graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 346–353.

[46] Lianghao Xia, Chao Huang, Yong Xu, Jiashu Zhao, Dawei Yin, and Jimmy Huang. 2022. Hypergraph contrastive collaborative filtering. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 70–79.

[47] Zhihui Xie, Tong Yu, Canzhe Zhao, and Shuai Li. 2021. Comparison-based Conversational Recommender System with Relative Bandit Feedback. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1400–1409.

[48] Junliang Yu, Hongzhi Yin, Jundong Li, Qinyong Wang, Nguyen Quoc Viet Hung, and Xiangliang Zhang. 2021. Self-supervised multi-channel hypergraph convolutional network for social recommendation. In *Proceedings of the Web Conference 2021*. 413–424.

[49] 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.

[50] Sen Zhao, Wei Wei, Ding Zou, and Xianling Mao. 2022. Multi-view intent disentangle graph networks for bundle recommendation. In *Proceedings of the 36th AAAI Conference on Artificial Intelligence*.

[51] 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*. 1006–1014.