LLM-based Conversational Recommendation Agents with Collaborative Verbalized Experience

Yaochen Zhu[♠], **Harald Steck**[♦], **Dawen Liang**[♦], **Yinhan He**[♠], **Nathan Kallus**[♣,♦], **Jundong Li**[♠]

[♠]University of Virginia, [♦]Netfli
{uqp4qh,nee7ne,jun
{hsteck,dliang,nka

Abstract

Large language models (LLM) have shown impressive zero-shot capabilities in conversational recommender systems (CRS). However, effectively utilizing historical conversations remains a significant challenge. Current approaches either retrieve few-shot examples or extract global rules to augment the prompt, which fail to capture the implicit and preference-oriented knowledge. To address the above challenge, we propose LLM-based **C**onversational **R**ecommendation **A**gents with Collaborative **V**erbalized **E**xperience (**CRAVE**). CRAVE starts by sampling trajectories of LLM-based CRS agents on historical queries and establishing verbalized experience banks by reflecting the agents' actions on user feedback. Additionally, a collaborative retriever network fine-tuned with item content-parameterized multinomial likelihood on query-item pairs is introduced to retrieve preference-oriented verbal experiences for new queries. Furthermore, we developed a debater-critic agent (DCA) system where each agent maintains an independent collaborative experience bank and works together to enhance the CRS recommendations. We demonstrate that the open-end debate and critique nature of DCA benefits significantly from the collaborative experience augmentation with CRAVE. The code is available at https://github.com/yaochenzhu/CRAVE.

1 Introduction

Conversational recommender systems (CRS) aim to recommend items by making dialogues with users (Jannach et al., 2021). Compared with traditional recommender systems (RS) that leverage historical interactions or item content to suggest new items (Lu et al., 2025; Shi et al., 2025; Zhu and Chen, 2022), CRSs engage users to express their preferences in natural language, which attracts more attention in both academia (He et al., 2023; Surana et al., 2025) and industry (Zhu et al., 2025).

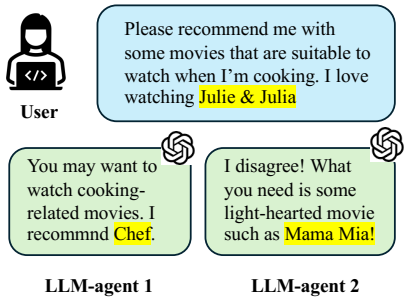


Figure 1: An exemplar conversational recommendation agent system with two debaters that offer different perspectives on user preference based on dialogue.

Traditional CRS methods train sequential models such as RNNs (Chung et al., 2014) or transformers (Vaswani et al., 2017) on historical conversations with groundtruth items to extract user preference, where external databases (e.g., item/word level knowledge graphs) are often introduced as the prior knowledge (Chen et al., 2019; Zhou et al., 2020). Afterward, efforts have been devoted to finetuning pretrained language models (PLM)[1], e.g., GPT-2, Llama-3, such that knowledge gained from external corpora can be utilized for better item/dialogue understanding (Wang et al., 2022; Feng et al., 2023). However, these PLMs are small in scale, whose reasoning ability is limited. Recently, CRSs built on large language models (LLM) with hundreds of billions of parameters, such as GPT-4o (OpenAI, 2024), have gained more attention. These models encompass substantial knowledge and show unprecedented reasoning ability over user preference based on dialogues (He et al., 2023), which substantially outperform finetuned PLM-based CRSs even in a zero-shot manner.

Despite the success of LLMs as zero-shot CRSs, utilizing the knowledge in historical conversations and user feedback still remains a great challenge. First, most LLMs are large black-box models,

[1]Following Zhao et al. (2023), we refer to large pretrained transformers with ***emergent*** zero-shot CRS ability compared with traditional CRS methods as LLMs, and refer to others that need finetuning to emulate traditional methods as PLMs.

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 2207–2220
November 4-9, 2025

which preclude model finetuning with historical conversations (He et al., 2023). One naive strategy is to retrieve conversation-feedback pairs as few-shot demonstrations in the prompt (Gao et al., 2023). However, the *semantic gap* between the conversation and user preference is substantial, making it difficult (even for the LLMs) to derive generalizable recommendations for new conversations in an in-context manner (Dong et al., 2024). To bridge the semantic gap, summarizing the implicit knowledge by reflecting on historical recommendations, as proposed in verbal reinforcement learning (VRL), (Yao et al., 2023; Kim et al., 2024; Shinn et al., 2024; Zhao et al., 2024) appears to be a promising strategy. However, unlike typical VRL tasks such as question answering (QA) that assume LLMs can have multiple attempts for each question with external feedback, recommendations for each user query are usually one-time. Consequently, Xi et al. (2024) introduced a general memory part in MemoCRS, which adapted VRL to LLM-based CRSs by summarizing global rules via reflecting their reasoning and recommendations for all the training samples, which are shared across all test queries. Nevertheless, these global rules often fail to account for the personalized preferences of different user queries, which are critical for CRSs.

Furthermore, all the methods introduced above focus on a single LLM with chain-of-thought (COT) reasoning (Wei et al., 2022). Nevertheless, Wang et al. (2023) found that LLMs tend to think convergently with COT (e.g., see Fig. 1, agent 1), but for CRS, since user queries are usually vague (otherwise users will directly search for the item instead of exploring with the system), it is important to promote *divergent* thoughts on different aspects of user preference. Recently, LLM debate has been introduced to address such limitations (Chan et al., 2024). However, the tasks that LLM debate focuses on (e.g., QA) usually have clear answers derived with logic/math reasoning from the question, whereas the answers for CRS are more vague and personalized, resembling an open-end debate without definite answers. Therefore, it is especially crucial to derive valuable experiences from historical conversations and user feedback to guide each debater to contribute new perspectives on user preferences. In addition, since there are no definite answers nor multiple attempts for CRS debaters to generate the final answers, it would be challenging to comprehensively evaluate the reasoning and recommendations of different debaters.

To address the challenges, we propose **CRAVE**, i.e., LLM-based **C**onversational **R**ecommendation **A**gents with Collaborative **V**erbalized **E**xperience. Specifically, CRAVE starts by sampling trajectories of LLM-based CRS agents on historical conversations and establishing verbalized experience banks by reflecting the agents' actions on the user feedback. Afterward, a collaborative retriever network finetuned on query-item pairs with item content-parameterized multinomial likelihood is introduced to retrieve the agent-specific preference-oriented experiences for each new query. Furthermore, we develop a debater-critic agent (DCA) system to tackle the convergent thinking issue of COT-based CRS agents, where each agent maintains an independent collaborative experience bank and works together to enhance the CRS recommendations. We find that the open-end debate nature of the DCA system benefits significantly from the collaborative experience augmentation with CRAVE.

2 Related Work

2.1 LLM Agent with Verbalized Experience

The simplest form of verbal experience for LLM agents is memorization, i.e., documents (Gao et al., 2023; Lei et al., 2025) or few-shot examples (Dong et al., 2024; Shi et al., 2024) retrieved based on semantic relevancy. However, in-context learning can be challenging for tasks like CRSs that require complex reasoning. To bridge the semantic gap, verbal reinforcement learning (VRL) was proposed to self-reflect on LLM agents' actions based on external feedback. For instance, RCI (Kim et al., 2024) iteratively prompts the LLM to critique and improve its previous output until the answer is correct, whereas Reflexion (Shinn et al., 2024) gains experience by reflecting on the failures. These methods typically require multiple attempts on a single test query, which is not feasible for CRS. Recently, EXPEL (Zhao et al., 2024) was proposed to retrieve experiences across tasks based on task-task semantic similarity. However, naive semantic similarity cannot be directly applied to CRS due to the significant semantic gap between the conversation and user preferences. The global memory introduced in Xi et al. (2024) reflects the reasoning and recommendations of an LLM-based CRS on the training samples to learn global rules to support future recommendations. However, the shared rules could struggle to account for the personalized preferences of different user queries.

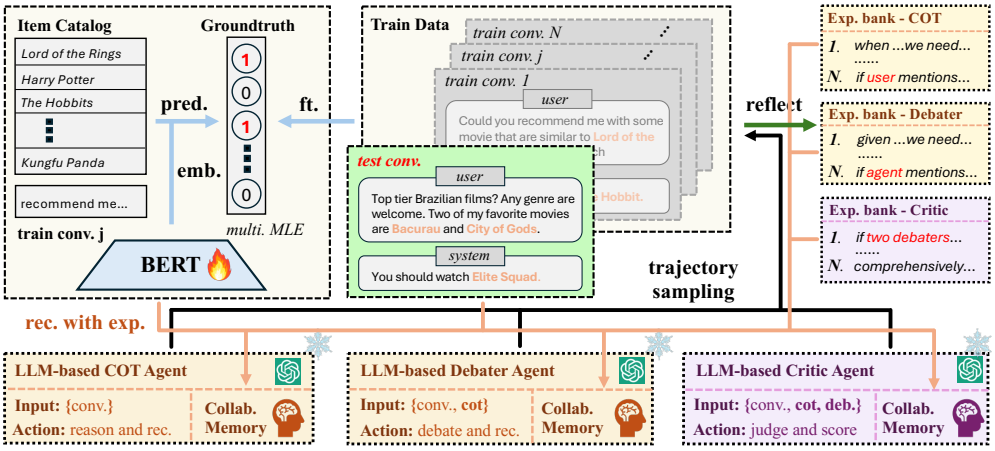


Figure 2: Overview of CRAVE for CRS and its three components: ***(i)*** conversational recommendation agents (the bottom part), ***(ii)*** collaborative experience retrieval (the left and right parts and all the lines except the light brown ones), and ***(iii)*** experience augmented generation (light brown lines).

2.2 Conversational Recommender System

Generally, CRS consists of two main modules: ***(i)*** conversation and ***(ii)*** recommendation (Jannach et al., 2021; Fang et al., 2024). *This paper focuses on the recommendation aspect of CRS*, i.e., suggesting new items to users based on their previous conversations with the system. Traditional methods (Zhou et al., 2020; Wang et al., 2022) rely on training sequential models, such as RNNs (Chung et al., 2014) or transformers (Vaswani et al., 2017), to understand the conversations and integrate them with the item information learned from recommendation models (Rendle, 2010; Vincent et al., 2008). These approaches often incorporate external knowledge databases, such as DBPedia (Auer et al., 2007) and ConceptNet (Speer et al., 2017), as the item/word prior knowledge. Afterward, pretrained language models (PLM) (Zhu et al., 2024) have gained more attention in CRS research, as they encapsulate prior knowledge of both natural language and items through pretraining on external corpora, which is beneficial for both conversation and item modeling in CRS (Wang et al., 2022). Recently, large language models (LLM) with hundreds of billions of parameters, e.g., GPT-4o (OpenAI, 2024), have emerged as the strongest baseline in CRS. He et al. (2023) demonstrated that these LLMs achieve excellent zero-shot recommendations, significantly outperforming both traditional and finetuned PLM-based methods. The aim of this paper is to further enhance the strongest zero-shot LLMs by developing a collaborative LLM-based CRS agent system that incorporates collaborative experience gained through self-reflection on historical conversations.

3 Problem Formulation

Let $\mathcal{I}$ denote the set of items in the system. We use $\{(u_t, s_t, \mathcal{I}_t)\}_{t=1}^{T}$ to denote a conversation between a user and the system, where at the t-th turn, $u_t \in \{\texttt{User}, \texttt{System}\}$ generates an utterance s_t, and $\mathcal{I}_t \subseteq \mathcal{I}$ denotes the set of mentioned items. When $u_T = \texttt{System}$, we have the groundtruth items $\mathcal{I}_T^{gt}$ (i.e., items with positive feedback) for the previous conversation $c = \{(u_t, s_t, \mathcal{I}_t)\}_{t=1}^{T-1}$. We denote all the historical conversations as $\mathcal{C}_{train} = \{c_1, c_2, \ldots, c_{N_{train}}\}$. For a test conversation $c^{te} = \{(u_t^{te}, s_t^{te}, \mathcal{I}_t^{te})\}_{t=1}^{T'-1}$, the aim of this paper is to develop an LLM-based CRS agent system that generates a ranked item list $\hat{\mathcal{I}}_{T'}^{te}$ from the catalog $\mathcal{I}$ with personalized verbal experience obtained from $\mathcal{C}_{train}$, such that $\hat{\mathcal{I}}_{T'}^{te}$ best matches the groundtruth items in $\mathcal{I}_{T'}^{te}$ (if $\mathcal{I}_{T'}^{te} \neq \emptyset$ and $u_{T'} = \texttt{System}$).

4 Methodology

The overall framework of CRAVE is illustrated in Fig. 2, which is composed of three components: ***(i)*** conversational recommendation agents, ***(ii)*** collaborative experience retrieval, and ***(iii)*** experience augmented generation. The details of CRAVE will be introduced in the following subsections.

4.1 Conversation Recommendation Agents

The conversational recommendation agents (CRA) that we study in this paper are a set of agents $\mathcal{A} = \{A_1, \ldots, A_{N_A}\}$ that work together to generate recommendations based on a user's dialogue with the system that seeks recommendations. Specifically, each agent $A_i \in \mathcal{A}$ is associated with an LLM-

based policy $\pi_i(a|c, r_{-i})$, where c denotes the conversation, r_{-i} denotes the responses/actions from the agents in $\mathcal{A}$ before agent i takes action, and a is the action taken by agent i based on c and r_{-i} (typically involving reasoning and recommendations).

4.1.1 Chain-of-Thought Agent

The simplest form of CRA is composed of only one chain-of-thought (COT) agent. In this case, $\mathcal{A} = \{A_{cot}\}$ and the policy $\pi_{cot}(a|c, r_{-cot})$ reduces to $\pi_{cot}(a|c)$ where action a reasons with the user's preference based on the conversation c and makes recommendations accordingly. Despite the efficiency, COT can lead to the convergent thinking issue as introduced in Wang et al. (2023), which may not be able to provide good and diverse recommendations that fully cover the user preference.

4.1.2 Debater-Critic Agent System

To address the limitation of the COT agent, we introduce a debater-critic agent (DCA) system for CRA that encourages divergent thinking on the user queries and generates recommendations that better cover the user preferences. In DCA, LLM agents in $\mathcal{A}$ are divided into two parts, i.e., the *debaters* $\mathcal{D} = \{A_1, \ldots, A_{N_A-1}\}$ that sequentially evaluates the reasoning and recommendations of the previous debaters, and the *critic* $Q = \{A_{N_A}\}$ that judges the reasoning and recommendations of all the debaters and provides the final recommendation list. Here, $A_1 \in \mathcal{D}$ is a COT agent that starts the debate.

Given that CRS resembles an open-end debate with no definite answers, since both the debaters $\mathcal{D}$ and the critic Q have no experiences on the actions that lead to good recommendations, zero-shot debaters may struggle to provide meaningful debates that maximally cover the user preferences, and the critic may fail to effectively evaluate the reasoning and recommendations from the debaters. To address this challenge, CRAVE leverages the training set $\mathcal{C}_{train}$ to gain verbalized experience that can be retrieved based on the implicit preference in the conversation c, thereby guiding $\mathcal{D}$ and Q to more effective debating and critiquing, respectively.

4.2 Collaborative Experience Retrieval

To leverage the historical conversations $\mathcal{C}_{train}$ to improve the actions of the agents, traditional reinforcement learning (RL) generally parameterizes each policy $\pi_i(a|c, r_{-i})$ with a learnable neural network and optimizes it with gradient-based methods such as policy gradient (Silver et al., 2014). However, since π_i in CRA is based on a blackbox LLM that inputs and outputs only natural language and precludes gradient-based updates, we adopt verbal reinforcement learning (VRL) and adaptively augment the policy $\pi_i(a|c, r_{-i})$ with an agent-specific retrieved experience $e(c; \mathcal{E}_i)$ as follows:

$$a \sim \pi_i\left(a|c, r_{-i}; e(c, \mathcal{E}_i)\right), \tag{1}$$

where $\mathcal{E}_i$ is the *experience bank* for agent i that stores the verbalized experience gained from each of the training conversations in $\mathcal{C}_{train}$ in natural language, and e is the *collaborative retrieval network* that selects the verbalized experiences in $\mathcal{E}_i$ based on the **user preference similarity** of the conversation c with the training conversations.

4.2.1 CRA Trajectory Sampling

To establish the experience bank $\mathcal{E}_i$ for agent i, we first collect its trajectories with policy $\pi_i(a|c, r_{-i})$ on the training conversations $\mathcal{C}_{train}$. Existing VRL methods typically require sampling multiple trajectories for each sample until the task succeeds (Yao et al., 2023). However, since it is challenging to exactly identify all the groundtruth items for CRS, we sample the trajectory only once for each conversation in $\mathcal{C}_{train}$, which we empirically show can already establish a good experience bank. Specifically, for the j-th training sample c_j, the trajectory for the COT agent A_{cot} can be sampled as follows:

$$r_{j,cot} \sim \pi_{cot}(a|c_j) = \mathbf{\Phi}\left(T^f_{cot}, F^f_{cot}, c_j\right). \tag{2}$$

Here, T^f_{cot} is the task-specific prompt that instructs the LLM (which we denote as $\mathbf{\Phi}$) to reason with user preference based on the conversation c_j and make recommendations accordingly, and F^f_{cot} is the format instruction that guides the LLM agent to output its reasoning and recommendations that can be easily processed with string split functions[2].

Similarly, the trajectory sampling process for the DCA system can be formulated as follows:

$$r_{j,i} \sim \pi_i(a|s_j, r_{j,<i}) = \mathbf{\Phi}\left(T^f_i, F^f_i, r_{j,<i}, c_j\right), \tag{3}$$

where $r_{j,1} = r_{j,cot}$ is the action of the COT agent that starts the debate, and r_{j,N_A} is the judgment provided by the critic Q. DCA supports an arbitrary number of debaters with arbitrary debate rounds. However, due to the efficiency constraint of CRS (as we cannot ask the user to wait too long for the

[2] See Appendix A for the prompts used in the main paper.

LLMs' debate), we consider only a one-round two-debater system and empirically show that it can already substantially improve over the COT agent.

In Eq. (3), the task-specific prompt T_2 instructs the second debater A_2 to find issues in the reasoning and recommendations of the first COT agent A_{cot} and address the problems by first providing the correct reasoning on user preference and based on it making new recommendations. T_3 instructs the critic Q to comprehensively evaluate the reasoning of the two debaters and provide numerical scores (in the range of $[-2, 2]$) to judge the quality of the items recommended by the two debaters.

4.2.2 Verbalized Experience Collection

It is extremely challenging for the COT agent and the DCA system to reason over user preference based on conversations *in a zero-shot manner* as both have no prior experience on the actions that lead to good recommendations. Fortunately, items with positive feedback, i.e., $\mathcal{I}_j^{gt}$, are available for the historical conversations $c_j \in \mathcal{C}_{train}$, which provide external guidance for the agents to reflect on their own actions. This allows for the summarization of useful experiences to guide their future actions when seeing similar conversations. For the COT agent A_{cot}, the reflection-based experience collection process can be formulated as follows:

$$e_{j,cot} = \mathbf{\Phi}\left(T_{cot}^b, F_{cot}^b, c_j, r_{j,cot}, \mathcal{I}_j^{gt}\right), \quad (4)$$

where the task-specific prompt T_{cot}^b instructs the LLM $\mathbf{\Phi}$ to reflect on the action $r_{j,cot}$ based on both the conversation c_j and the groundtruth items $\mathcal{I}_j^{gt}$. Specifically, the LLM is asked to assess the correctness of the reasoning and recommendations in $r_{j,cot}$, provide the rationale for the assessment, and finally offer useful experiences that can be applied to similar conversations in the future. Similarly, the experience collection process for agent i in the DCA system can be formulated as follows:

$$e_{j,i} = \mathbf{\Phi}\left(T_i^b, F_i^b, r_{j,<i}, c_j, r_{j,i}, \mathcal{I}_j^{gt}\right), \quad (5)$$

where the prompt T_i^b is similar to that in Eq. (4). Compared to Eq. (4), Eq. (5) also considers the trajectory of previous agents, i.e., $r_{j,<i}$ when reflecting on the action $r_{j,i}$ with conversation c_j and groundtruth items $\mathcal{I}_j$. The final experience bank for DCA can be established as $\mathcal{E}_i = \{e_{j,i}\}_{j=1}^{N_{train}}$.

4.2.3 Collaborative Retrieval Network

After establishing the experience bank $\mathcal{E}_i$, we introduce the collaborative retrieval model $e(c, \mathcal{E}_i)$ defined in Eq. (1). The backbone model for e is a Sentence-BERT (Reimers, 2019), which finetunes a BERT model $\mathbf{\Psi}$ on document pair-wise similarity. However, user queries with good semantic similarity do not necessarily share similar preferences. Therefore, we further finetune $\mathbf{\Psi}$ with *collaborative similarity*, such that the embeddings of conversations that share similar preferences become similar to each other. The user preference similarity of two queries c_i and c_j can be measured by the normalized overlap of groundtruth items as:

$$O_{i,j} = |\mathcal{I}_i^{gt} \cap \mathcal{I}_j^{gt}| / |\mathcal{I}_i^{gt} \cup \mathcal{I}_j^{gt}|. \quad (6)$$

However, $O_{i,j}$ simply counts the number of overlaps and ignores the item content, which is critical for conversational recommendations. Therefore, we propose a novel *collaborative finetuning* strategy that encourages the embeddings of c_j to be simultaneously similar to the content of all the groundtruth items in $\mathcal{I}_j^{gt}$. This is achieved by maximizing the item content-parameterized multinomial likelihood on all the $(c_j, \mathcal{I}_j^{gt})$ pairs in $\mathcal{C}_{train}$ as:

$$\mathcal{I}_j \sim \text{multi}\left(\text{softmax}\left(\mathbf{W}^T \mathbf{\Psi}(c_j)\right), |\mathcal{I}_j^{gt}|\right), \quad (7)$$

where $\mathbf{W} = \mathbf{\Psi}(T) \in \mathbb{R}^{d \times |\mathcal{I}|}$, $T = \left[t_1 | \ldots | t_{|\mathcal{I}|}\right]$ is the stacked content of all the catalog items described in natural language. For movies, we directly use the movie title as t_k, as the BERT model $\mathbf{\Psi}$ already gained sufficient knowledge of the movies through pretraining. d is the dimension of the content embeddings. During finetuning, we monitor the overlap of groundtruth items (as Eq. (6)) of validation conversations with top-K retrieved training conversations. The final collaborative retrieval network e is formulated as follows:

$$e(c, \mathcal{E}_i) = \left\{e_{k,i} | k \in \text{top}_K(\text{sim}(\hat{\mathbf{\Psi}}(c), \hat{\mathbf{\Psi}}(c_k)))\right\}, \quad (8)$$

where the top_K function selects the indices of training conversations with the top K collaborative similarity judged by the finetuned Sentence-BERT $\hat{\mathbf{\Psi}}$.

4.3 Experience Augmented Generation

Finally, combining the established experience bank $\mathcal{E}_i$ and the collaborative retrieval network e finetuned from $\mathbf{\Psi}$, for the COT agent A_{cot}, the recom-

Table 1: Statistics of `Redial` and `Reddit-v2` datasets, where **#train** and **#val/test** denote the number of rounds.

dataset	#items	#train	#val/test
Redial	8,010	186,546	11,564
Reddit-v2	20,193	108,989	7,565

mendations for a test conversation c^{te} with collaborative verbalized experience can be formed as:

$$r_{cot}^{te} \sim \pi_{cot}\left(a|c^{te}, e_{cot}^{te}\right) = \mathbf{\Phi}\left(T_{cot}^{fe}, F_{cot}^{fe}, c^{te}, e_{cot}^{te}\right), \tag{9}$$

where the task-specific prompt T_{cot}^{fe} instructs the LLM $\mathbf{\Phi}$ to *utilize the retrieved verbalized collaborative experience* $e_{cot}^{te} = e(c^{te}, \mathcal{E}_{cot})$ to reason with user preference based on the conversation c^{te} and make recommendations accordingly. In addition, with the format instruction F_{cot}^{fe}, the recommendation list $\hat{\mathcal{I}}_c^{te}$ can be directly extracted from the response r_{cot}^{te}. For DCA, collaborative experience augmented responses can be formulated as follows:

$$\begin{aligned} r_i^{te} \sim & \pi_i\left(a|c^{te}, r_{<i}^{te}, e_i^{te}\right) \\ & = \mathbf{\Phi}\left(T_i^{fe}, F_i^{fe}, r_{<i}^{te}, c^{te}, e_i^{te}\right), \end{aligned} \tag{10}$$

where instructions T_i^{fe} and F_i^{fe} are similar to those defined in Eqs. (3) and (9). After the experience augmented debate among the agents $A_i \in \mathcal{D}$ and the evaluation from the critic Q, item-score pairs are extracted from the response of Q, i.e., r_{-1}^{te}, and the items are re-ranked by the quality score judged by Q to form the final recommendation list $\hat{\mathcal{I}}_{dca}^{te}$.

5 Empirical Study

5.1 Datasets

We consider two widely used real-world CRS datasets, i.e., the `Redial` dataset (Li et al., 2018) and a movie-name corrected Reddit dataset (Zhu et al., 2025) (which we name `Reddit-v2`). For both datasets, two people play the role of the `user` who seeks movies to watch and the `system` that responds with movie recommendations through conversations. For pre-processing, we keep the training set and randomly split the original test set into a validation part to select the collaborative retrieval model (see Section 4.2.3) and a test part for the final CRA evaluation. The statistics of the two datasets are summarized in Table 1 for reference.

5.2 Implementation Details

Previous works found that the state-of-the-art GPT-4o (OpenAI, 2024) achieves the best zero-shot performance on both `Redial` and `Reddit-v2` datasets (He et al., 2023). In the main paper, we use GPT-4o as the backbone and show that `CRAVE` can further improve the zero-shot performance of GPT-4o by forming a CRA system with augmented collaborative experience. For the collaborative retrieval network, we leverage the 400M Stella model as the backbone Sentence-BERT and finetune it as with Eq. (7) for 7 epochs with the learning rate $1e^{-5}$. The normalized overlap metric defined in Eq. (6) on the validation set was monitored during training (see Fig. 3 for the training dynamics with ten independent training runs of the collaborative retrieval network), where the best $\hat{\mathbf{\Psi}}$ is saved and used for retrieving the verbalized collaborative experience.

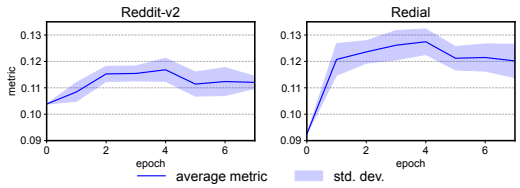
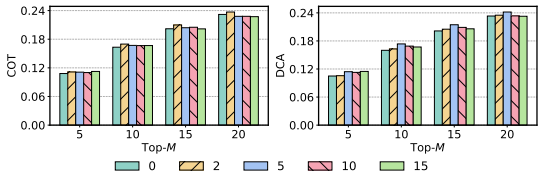
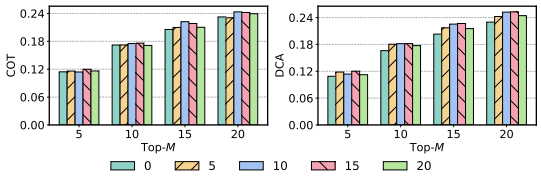


Figure 3: Dynamic of finetuning collaborative retrieval network with content-parameterized mult. likelihood.



(a) Reddit-v2 Dataset



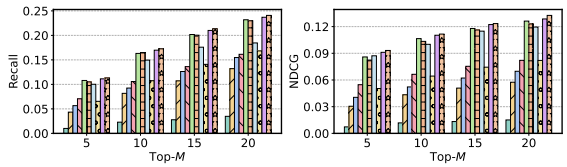
(b) Redial Dataset

Figure 4: Performance of CRAVE with the COT agent and DCA system w.r.t. different number of retrieval K.

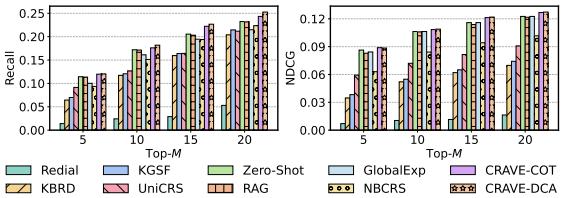
Computational Overhead. After extracting the experiences, retrieval and augmentation increase the input context of CRA during inference for a new conversation. Therefore, `CRAVE` could increase the latency compared with the backbone CRA.

5.3 Performance w.r.t. Number of Retrieval

For `CRAVE`, the number of training samples from which the collaborative experiences are retrieved, i.e., K in Eq. (8), is an important hyperparameter. Therefore, we first explore the performance of `CRAVE` for both the COT agent and DCA system when K increases. The results are illustrated in Fig. 4. Fig. 4 shows that the performance of



(a) Reddit-v2 Dataset



(b) Redial Dataset

Figure 5: Comparison between CRAVE and various baselines on the Reddit-v2 and Redial datasets.

CRAVE first peaks and then drops as K gets larger. This is because when K is too small, insufficient experiences are retrieved, which fail to guide the CRA to take reasonable actions that lead to good recommendations. In constrast, when K is too large, less relevant experiences may be retrieved, which could risk biasing the recommendations. In addition, we note that a smaller value of K leads to optimal performance on the Reddit-v2 dataset. This is probably because of its more diverse topics compared with the Redial dataset, where a small neighborhood keeps the retrieved experiences most relevant to the test query. Finally, we also note that DCA cannot outperform COT on *Redial* dataset in a zero-shot manner (see Fig. 4). However, it benefits significantly from CRAVE and outperforms COT when collaborative experience is retrieved to facilitate the debate and the critique. Qualitative analysis of the top-K experiences and recommendations is provided in Section B of the Appendix.

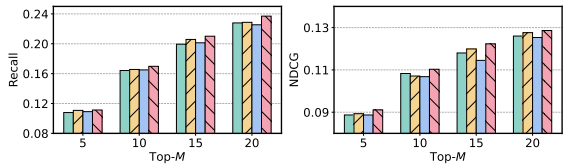
5.4 Comparison with Baselines

We now use the best K selected on the validation set shown in Fig. 4 and compare CRAVE with various state-of-the-art CRS baselines as follows:

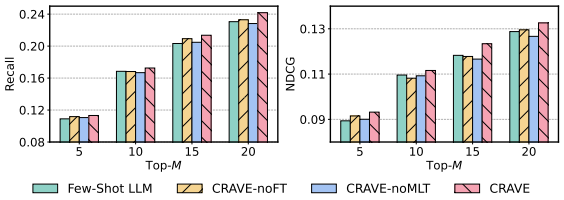
- **Redial** (Li et al., 2018) uses an RNN to model conversations and a denoising autoencoder to model items and generate recommendations.
- **KBRD** (Chen et al., 2019) introduces a relational graph neural network (GNN) on the DBpedia to model entities, and optimize similarity between tokens and entities to fuse semantics.
- **KGSF** (Zhou et al., 2020) incorporates ConceptNet to model the conversations, with mutual information maximization w.r.t. entity KG embeddings to fuse the entity information.
- **UniCRS** (Wang et al., 2022) introduces a pre-trained transformer, i.e., DialoGPT, to capture the context information w.r.t. the entity KG embeddings used for semantic fusion.
- **Zero-shot LLM** (He et al., 2023) directly inputs the historical dialogue with task-specific prompt and formats instructions for CRS without any retrieval from the external knowledge database.
- **RAG** (Lewis et al., 2020) denotes the model that retrieves item-related sentences from a database of movie plots and metadata based on semantic similarity between the query and sentences.
- **NBCRS** (Xie et al., 2024) uses Sentence-BERT to retrieve training queries and take vote of groundtruths as recommendations. Neighbor size is selected based on the validation set.
- **GlobalExp** proposed in (Xi et al., 2024)[3] goes through all training samples and summarized rules that are shared among all test queries.

The results are illustrated in Fig. 5. From the figure we can find that, the zero-shot LLM is the strongest baseline that outperforms Redial, KBRD, KGSF, and UniCRS, where laborious training is required to achieve good performance. We also observe that NBCRS, i.e., a simple neighborhood-based method, outperforms most traditional methods. Furthermore, we note that naively retrieving the movie content with RAG does not help, which is probably due to the large semantic gap between the content and user preference. In addition, we also observe limited improvement when including global rules summarized from the training data to the performance of the zero-shot LLM, as applying the same rules for all the queries may not cater to different users' personalized preferences. Augmented with verbalized collaborative experience gained on historical recommendations via reflection on user feedback, we can see in Fig. 5 that CRAVE outperforms all the baselines. In addition, we note that the DCA system has better coverage of groundtruth items compared with the COT agent.

[3]Please note that MemoCRS (Xi et al., 2024) is composed of multiple components, and only the global experience part is relevant to the VRL-based method studied in this paper.



(a) CRAVE with COT Agent



(b) CRAVE with DCA System

Figure 6: Comparison between CRAVE and various ablation models on the Reddit-v2 dataset.

5.5 Ablation Study

We conduct the ablation study to demonstrate the effectiveness of the ***(i)*** verbalized experience collection module (see Section 4.2.2) and the ***(ii)*** collaborative experience retrieval module (see Section 4.2.3) in CRAVE. To answer the research questions, we design the following baselines:

- **Few-shot LLM** directly retrieves the query-groundtruth pairs from $\mathcal{C}_{train}$ to augment the test query instead of summarizing the experiences by reflecting on the actions on $\mathcal{C}_{train}$.
- **CRAVE-noFT** directly uses pretrained Stella-400M model for experience retrieval instead of finetuning it with item content parameterized multinomial likelihood defined in Eq. (7).
- **CRAVE-noMLT** uses the normalized overlap metric in Eq. (6) as the pairwise loss for conversations to finetune the retrieval network, without considering item content information.

For all the ablation models, the best K for retrieving top-K training samples is determined by the validation set. The results are illustrated in Fig. 6. From Fig. 6, we can find that CRAVE compares favorably with the few-shot LLM baseline. This further demonstrates the effectiveness of the reflection-based collaborative experiences module introduced in CRAVE, as LLMs may fail to directly generalize from the in-context demonstrations due to the large semantic gap. In addition, the substantial improvement of CRAVE over the CRAVE-noFT and CRAVE-noMLT variants further shows the effectiveness of the collaborative retrieval network in retrieving user-preference-oriented experiences for the conversational recommendation agents. In addition, we note that finetuning the retrieval network with non-content based metric (i.e., CRAVE-noMLT) decreases the performance compared to the variant with no finetuning at all (i.e., CRAVE-noMLT), which highlights the importance of both collaborative and content information to retrieve the verbalized experiences for the LLM-based CRS agents.

Table 2: Comparison of in-list recommendation similarity (***inv. diversity***) between CRAVE for COT agent and DCA system on Reddit-v2 and Redial datasets.

Methods	Reddit-v2		Redial	
	cont. ↓	collab. ↓	cont. ↓	collab. ↓
CRAVE-COT	0.505	0.431	0.465	0.457
CRAVE-DCA	0.458	0.425	0.449	0.454

6 Diversity Analysis

Finally, we compare the recommendation diversity of CRAVE with COT and DCA backbones in Table 2. Specifically, we consider two aspects of in-list recommendation diversity, i.e., content diversity and collaborative diversity. We use the average in-list pairwise similarity of Stella-400M embeddings of the titles of recommended movies (averaged over all test conversations) to measure the content diversity. Additionally, we assess collaborative similarity with movie embeddings derived by training an EASE (Steck, 2019) model on item co-mentions in the training conversations. However, due to the sparsity of item co-mention data, this is not as sensitive as the content metric. The results indicate that DCA augmented with verbal experience not only improves recommendation accuracy but also enhances diversity compared with the COT agent.

7 Conclusions

In this paper, we introduced CRAVE, i.e., conversational recommendation agents with collaborative verbalized experience. We show that valuable experiences can be gained from the training set via trajectory sampling and self-reflection, which substantially augments the current state-of-the-art LLM-based CRS agent systems. We also find that the improvement is especially evident for the debater-critic system, which resembles an open-end debate without definite groundtruth. In addition, we show that the collaborative retriever network, which is a fine-tuned BERT model that encodes preference similarity of conversations, plays a key role in CRAVE to retrieve personalized experiences.

Acknowledgment

The authors would like to acknowledge the gift funding support from Netflix.

8 Limitations

One limitation of CRAVE is that, gaining the retrieved collaborative experience generally requires a large LLM with emergent ability. Therefore, the main aim of CRAVE is to improve the state-of-the-art large LLMs, e.g., GPT-4o, Llama-3.1 405B, which is the current state-of-the-art in CRS research. CRAVE generally cannot be applied to smaller PLMs with substantially weaker reasoning ability to gain the experience themselves.

9 Ethics Statement

LLMs may inherit social bias from the real-world data in their training corpora. Since we leverage the LLMs to recommend items to users through conversations, there is the risk of triggering the inherent biases of LLMs similar to other approaches where LLMs are used in downstream tasks. In addition, the trained collaborative retrieval network may inherit the bias in the training set, which could be prevented by filtering the training conversations.

References

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. Dbpedia: A nucleus for a web of open data. In *International Semantic Web Conference*, pages 722–735. Springer.

Chi-Min Chan, Weize Chen, Yusheng Su, Jianxuan Yu, Wei Xue, Shanghang Zhang, Jie Fu, and Zhiyuan Liu. 2024. Chateval: Towards better llm-based evaluators through multi-agent debate. In *ICLR*.

Qibin Chen, Junyang Lin, Yichang Zhang, Ming Ding, Yukuo Cen, Hongxia Yang, and Jie Tang. 2019. Towards knowledge-based recommender dialog system. In *EMNLP*.

Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NeurIPS Workshop*.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Baobao Chang, et al. 2024. A survey on in-context learning. In *EMNLP*, pages 1107–1128.

Jiabao Fang, Shen Gao, Pengjie Ren, Xiuying Chen, Suzan Verberne, and Zhaochun Ren. 2024. A multi-agent conversational recommender system. *arXiv preprint arXiv:2402.01135*.

Yue Feng, Shuchang Liu, Zhenghai Xue, Qingpeng Cai, Lantao Hu, Peng Jiang, Kun Gai, and Fei Sun. 2023. A large language model enhanced conversational recommender system. *arXiv preprint arXiv:2308.06212*.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*.

Zhankui He, Zhouhang Xie, Rahul Jha, Harald Steck, Dawen Liang, Yesu Feng, Bodhisattwa Prasad Majumder, Nathan Kallus, and Julian McAuley. 2023. Large language models as zero-shot conversational recommenders. In *CIKM*.

Dietmar Jannach, Ahtsham Manzoor, Wanling Cai, and Li Chen. 2021. A survey on conversational recommender systems. *CSUR*, 54(5):1–36.

Geunwoo Kim, Pierre Baldi, and Stephen McAleer. 2024. Language models can solve computer tasks. In *NeurIPS*.

Zhenyu Lei, Zhen Tan, Song Wang, Yaochen Zhu, Zihan Chen, Yushun Dong, and Jundong Li. 2025. Learning from diverse reasoning paths with routing and collaboration. *EMNLP*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *NeurIPS*, pages 9459–9474.

Raymond Li, Samira Ebrahimi Kahou, Hannes Schulz, Vincent Michalski, Laurent Charlin, and Chris Pal. 2018. Towards deep conversational recommendations. In *NeurIPS*.

Xuan Lu, Sifan Liu, Bochao Yin, Yongqi Li, Xinghao Chen, Hui Su, Yaohui Jin, Wenjun Zeng, and Xiaoyu Shen. 2025. Multiconir: Towards multi-condition information retrieval. *Preprint*, arXiv:2503.08046.

OpenAI. 2024. Hello gpt-4o. https://openai.com/index/hello-gpt-4o/.

N Reimers. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Steffen Rendle. 2010. Factorization machines. In *ICDM*, pages 995–1000.

Yunxiao Shi, Wujiang Xu, Zhang Zeqi, Xing Zi, Qiang Wu, and Min Xu. 2025. PersonaX: A recommendation agent-oriented user modeling framework for long behavior sequence. In *Findings of ACL*, pages 5764–5787. Association for Computational Linguistics.

Yunxiao Shi, Xing Zi, Zijing Shi, Haimin Zhang, Qiang Wu, and Min Xu. 2024. Enhancing retrieval and managing retrieval: A four-module synergy for improved quality and efficiency in rag systems. In *ECAI*, pages 2258–2265.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2024. Reflexion: Language agents with verbal reinforcement learning. In *NeurIPS*.

David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. 2014. Deterministic policy gradient algorithms. In *ICML*, pages 387–395. Pmlr.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *AAAI*.

Harald Steck. 2019. Embarrassingly shallow autoencoders for sparse data. In *WWW*, pages 3251–3257.

Rohan Surana, Junda Wu, Zhouhang Xie, Yu Xia, Harald Steck, Dawen Liang, Nathan Kallus, and Julian McAuley. 2025. From reviews to dialogues: Active synthesis for zero-shot llm-based conversational recommender system. *arXiv preprint arXiv:2504.15476*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, and Aidan N Gomez. 2017. Attention is all you need. In *NeurIPS*.

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. 2008. Extracting and composing robust features with denoising autoencoders. In *ICML*, pages 1096–1103.

Xiaolei Wang, Kun Zhou, Ji-Rong Wen, and Wayne Xin Zhao. 2022. Towards unified conversational recommender systems via knowledge-enhanced prompt learning. In *KDD*, pages 1929–1937.

Yu Wang, Zhiwei Liu, Jianguo Zhang, Weiran Yao, Shelby Heinecke, and Philip S Yu. 2023. Drdt: Dynamic reflection with divergent thinking for llm-based sequential recommendation. *arXiv preprint arXiv:2312.11336*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS*.

Yunjia Xi, Weiwen Liu, Jianghao Lin, Bo Chen, Ruiming Tang, Weinan Zhang, and Yong Yu. 2024. Memocrs: Memory-enhanced sequential conversational recommender systems with large language models. In *CIKM*, pages 2585–2595.

Zhouhang Xie, Junda Wu, Hyunsik Jeon, Zhankui He, Harald Steck, Rahul Jha, Dawen Liang, Nathan Kallus, and Julian McAuley. 2024. Neighborhood-based collaborative filtering for conversational recommendation. In *RecSys*, pages 1045–1050.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. In *ICLR*.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. 2024. Expel: Llm agents are experiential learners. In *AAAI*, pages 19632–19642.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

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 *KDD*, pages 1006–1014.

Yaochen Zhu and Zhenzhong Chen. 2022. Mutually-regularized dual collaborative variational auto-encoder for recommendation systems. In *WWW*, pages 2379–2387.

Yaochen Zhu, Chao Wan, Harald Steck, Dawen Liang, Yesu Feng, Nathan Kallus, and Jundong Li. 2025. Collaborative retrieval for large language model-based conversational recommender systems. In *WWW*, pages 3323–3334.

Yaochen Zhu, Liang Wu, Qi Guo, Liangjie Hong, and Jundong Li. 2024. Collaborative large language model for recommender systems. In *WWW*, pages 3162–3172.

Appendix

A Prompts Used in the Main Paper

In this section, we provide the task-specific prompt and format instructions that we defined in the main paper for trajectory sampling, experience reflection, and recommendation stages of CRAVE.

Eq. (2): COT Agent, Trajectory Sampling

T^f_{cot}: Pretend that you are a movie recommender system. Here is the user's query: $\{c_j\}$

F^f_{cot}: Specifically, after writing down your reasoning, write #### to mark the beginning of your recommendation list. Then, list EXACTLY 20 movie recommendations, each on a new line with no extra sentences.

Eq. (3): Debater Agent, Trajectory Sampling

T^f_2: Pretend you are a movie recommender system. Here is a user's query: $\{c_j\}$. Below is the reasoning and recommendation list from another movie recommender system: $\{r_{j,1}\}$. Evaluate the reasoning for any potential issues. Even if the reasoning is sound, the provided recommendations may not align well with it. Analyze these aspects and provide your corrected reasoning and recommendations.

F^f_2: After completing your reasoning, write #### to indicate the start of your recommendation list. Then, list EXACTLY 20 movie recommendations, each on a new line with no extra sentences.

Eq. (3): Critic Agent, Trajectory Sampling

T^f_3: You are a judge for a debate on movie recommendations for the user query: $\{c_j\}$. The debate between two movie recommender systems is as follows: Movie Recommender System 0: $\{r_{j,1}\}$ \n\n Movie Recommender System 1: $\{r_{j,2}\}$ \n\n Your task is to reflect on both movie recommender systems and comprehensively critique the reasoning and recommendations from each side. After providing your analysis, generate scores for each movie from both recommender systems to indicate the quality of the recommendation. Use the following scale: -2 for very bad, -1 for bad, 0 for neutral, 1 for good, and 2 for very good.

F^f_3:Write #### to mark the beginning of your judgment on the recommendation list. Then, list the movies from both sides with their scores in the format: movie_name####score, each on a new line with no extra sentences.

Eq. (4): COT Agent, Experience Reflection

T^b_{cot}: You are evaluating a movie recommender system. Assess the reasoning and recommended movies based on the user query: $\{c_j\}$. Here is the reasoning and recommended movies from the system: $\{r_{j,cot}\}$. Here are the ground truth movies the user wants to watch: $\{\mathcal{I}_j\}$. Determine if the reasoning and recommendations are successful by checking the consistency with the ground truth movies and the overlap with recommended movies.

F^b_{cot}: First, provide your judgment: success/failure, followed by ####. Next, analyze why the reasoning/recommendations succeed or fail based on the user query and ground truth movies, followed by ####. Finally, summarize general guidelines for making movie recommendations for similar user queries, based on your analysis.

Eq. (5): Debater Agent, Experience Reflection

T^b_2: You are evaluating a debate on movie recommendations. Assess the reasoning and recommended movies of Debater 1 based on the user query: $\{c_j\}$ and the initial argument by Debater 0: $\{r_{j,1}\}$. Here is the reasoning and recommended movies of Debater 1: $\{r_{j,2}\}$. Here are the ground truth movies the user wants to watch: $\{\mathcal{I}_j\}$ \n\n Determine if Debater 1's reasoning and recommendations are successful by checking the consistency with the ground truth movies and the overlap with recommended movies.

F^b_2: First, provide your judgment: success/failure, followed by ####. Next, analyze why Debater 1's reasoning/recommendations

succeed or fail based on the user query and ground truth movies, followed by ####. Finally, summarize general guidelines for Debater 1 when debating for similar user queries, based on your analysis.

Eq. (5): Critic Agent, Experience Reflection

T_3^b: You are evaluating a critique on a debate about movie recommendations. Assess the critique provided on the recommendations from two movie recommender systems based on the user query: $\{c_j\}$. Here is the reasoning and recommendations from Debater 0: $\{r_{j,1}\}$ \n\n Here is the reasoning and recommendations from Debater 1: $\{r_{j,2}\}$ \n\n Here is the critique you need to evaluate: $\{r_{j,3}\}$ \n\n Here are the ground truth movies the user wants to watch: $\{\mathcal{I}_j\}$. Determine if the critique's reasoning and recommendations are successful by checking consistency with the ground truth movies and overlap with highly scored movies.

F_3^b: First, provide your judgment: success/failure, followed by ####. Next, analyze why the critique's reasoning/scoring succeeds or fails based on the user query and ground truth movies, followed by ####. Finally, summarize general guidelines for critiquing movie recommendations for similar user queries, based on your analysis.

Eq. (9): COT Agent, Exp. Augmented Gen.

T_{cot}^{fe}: Pretend you are a movie recommender system. Here is a user's query $\{c^{te}\}$. When making recommendations, consider the following guidelines: $\{e_{cot}^{te}\}$ \n\n

F_{cot}^{fe}: First, provide your judgment: success/failure, followed by ####. Next, analyze why the critique's reasoning/scoring succeeds or fails based on the user query and ground truth movies, followed by ####. Finally, summarize general guidelines for critiquing movie recommendations for similar user queries, based on your analysis.

Eq. (10): Debater Agent, Exp. Augmented Gen.

T_2^{fe}: Pretend you are a movie recommender system. Here is a user's query: $\{c^{te}\}$. Below is the reasoning and recommendation list from another movie recommender system: $\{r_1^{te}\}$. Evaluate the reasoning for any potential issues. Even if the reasoning is sound, the recommendations may not align well with it. Analyze these aspects and provide your corrected reasoning and recommendations. When doing evaluation and making recommendations, consider the following guidelines: $\{e_2^{te}\}$ \n\n

F_2^{fe}: After completing your reasoning, write #### to indicate the start of your recommendation list. Then, list EXACTLY 20 movie recommendations, each on a new line with no extra sentences.

Eq. (10): Critic Agent, Exp. Augmented Gen.

T_3^{fe}: You are a judge for a debate on movie recommendations for the user query: $\{c^{te}\}$. The debate between two movie recommender systems is as follows: Movie Recommender System 0: $\{r_1^{te}\}$ \n\n Movie Recommender System 1: $\{r_2^{te}\}$ \n\n Your task is to reflect on both movie recommender systems and comprehensively critique the reasoning and recommendations from each side. After providing your analysis, generate scores for each movie from both recommender systems to indicate the quality of the recommendation. Use the following scale: -2 for very bad, -1 for bad, 0 for neutral, 1 for good, and 2 for very good. Consider the following rules when you make the judgment: $\{e_3^{te}\}$ \n\n

F_3^{fe}: Write #### to mark the beginning of your judgment on the recommendation list. Then, list the movies from both sides with their scores in the format: movie_name####score, each on a new line with no extra sentences.

B Qualitative Analysis

In this section, we present a qualitative analysis of the retrieved experiences for CRAVE with the COT and DCA backbones, as shown in Tables 3 and 4. From these tables, we observe that the retrieved experiences maintain a good balance between generalization and catering to the specific preferences expressed in the query, whihch results in overall better recommendations compared to zero-shot LLMs.

Table 3: Qualitative analysis of the recommendations of zero-shot LLM and CRAVE for the COT Agent.

User Query	**Recommendations and Groundtruth**
User: Films with news headline montages. Anyone have suggestions of films that have a montage of news headlines and newscasters? Looking for some montage style references.	**COT (Zero-shot)**: Citizen Kane, Good Night, and Good Luck, The Social Network, All the President's Men **COT (CRAVE)**: **Network**, Citizen Kane, The Truman Show, Wag the Dog, Good Night, and Good Luck **Groundtruth:** Network
Exemplar Experiences **Broaden the thematic scope:** While focusing on media and advertising, consider films that offer a critical or satirical perspective on society, even if they don't directly address media themes. **Consider Tone and Style:** Pay attention to the tone and style of the films the user mentions, such as satire or drama, and ensure recommendations align with these preferences.	
User: Movies where a big-timer befriends a small-timer... then the small-timer gets into a major conflict with the big-timer. Some examples: Last King of Scotland, A Bronx Tale, and Gangs of New York.	**COT (Zero-shot)** The Devil's Advocate, The Departed, Training Day, Scarface, The Godfather **COT (CRAVE)** The Departed, **American Gangster**, Donnie Brasco, The Devil's Advocate, Training Day **Groundtruth:** American Gangster
Exemplar Experiences **Align with User's Examples:** Pay close attention to the examples provided by the user and ensure that recommendations closely match the tone, style, and themes of those examples. **Understand the Core Criteria:** Identify the core elements the user is interested in, such as unexpected temper, hero-type characters, and memorable scenes, and ensure recommendations embody these traits.	
User: Movies with/about unsuspected murderers. Like a character, main or not, who seems so innocent but ends up ruthlessly murdering someone or multiple people.	**COT (Zero-shot)** Psycho, Gone Girl, The Usual Suspects, **Primal Fear**, Scream **COT (CRAVE)** **Primal Fear**, Gone Girl, The Talented Mr. Ripley, Se7en, The Usual Suspects **Groundtruth:** Primal Fear
Exemplar Experiences **Clarify User Intent:** Ensure a clear understanding of whether the user is more interested in the procedural aspect of the investigation or the psychological exploration of the killer. **Balance Explicit and Implicit Interests:** While addressing the explicit request, also consider the user's implicit interests, which might be inferred from the examples they provide or their viewing history.	
User: The main character is accused of something and thrown in jail/imprisoned somewhere for a long time. Something like The Count of Monte Cristo.	**COT (Zero-shot)** The Count of Monte Cristo, V for Vendetta, The Green Mile, The Shawshank Redemption, Kill Bill: Vol. 1 **COT (CRAVE)** V for Vendetta, **Oldboy**, The Shawshank Redemption, Law Abiding Citizen, The Prestige **Groundtruth:** Oldboy
Exemplar Experiences **Consider User-Provided Examples:** Use examples as strong indicators of their preferences and ensure that similar movies are prioritized in the recommendations. **Include a Broader Range:** Include a broader range of movies that fit the theme, including lesser-known films that might align with the user's interests.	

Table 4: Qualitative analysis of the recommendations of zero-shot LLM and CRAVE for the DCA system.

<table>
<tr><th>User Query</th><th>Recommendations and Groundtruth</th></tr>
<tr><td>User: Any films that will leave me feeling more intelligent?. Looking for some well-made films that will get me thinking, have me learning, leave me questioning. Any suggestions?</td><td>DCA (Zero-shot): Inception, The Matrix, Interstellar, Eternal Sunshine of the Spotless Mind, Arrival
DCA (CRAVE): Synecdoche, New York, The Tree of Life, The Seventh Seal, The Double Life of Véronique, Solaris
Groundtruth: Synecdoche, New York</td></tr>
<tr><td colspan="2">Exemplar Experiences from Debater 1
Understand User Preferences: Pay close attention to the specific themes and types of movies the user is interested in, such as existential or philosophical themes, rather than focusing solely on science and technology.
Diverse Themes: Include a broader range of themes, such as personal growth, emotional depth, and interconnectedness, to better match the user's interests.

Exemplar Experiences from Critic
Alignment with User Preferences: Ensure that the recommendations align with the user's specific interests and preferences, as indicated by any ground truth or explicit requests.
Diversity in Themes and Genres: While maintaining a focus on the user's request (e.g., cerebral films), include a variety of themes and genres to capture different aspects of the user's interests.</td></tr>
<tr><td>User: What movie feels like a warm blanket?. The most notable one for me is Harry Potter, but these days i like a little bit more grit. So lately Prometheus has topped it. Some others that come to mind for me are Blade Runner, the Planet of the Apes trilogy, LOTR, the Holiday, Hunger Games, True Detective. I'm realizing most of these are fantasy, but they don't have to be. I'm looking for that rainy day feeling that wraps you up. I get a cup of tea and bundle up and pretend i live in a city that ever gets cold, which i don't. I'd love to have some recommendations.</td><td>Critic (Zero-shot): Pan's Labyrinth, Children of Men, Annihilation, Ex Machina, The Shape of Water
Critic (CRAVE): Pan's Labyrinth, Arrival, The Lord of the Rings: The Fellowship of the Ring, The Shape of Water, The Grand Budapest Hotel
Groundtruth: The Lord of the Rings: The Fellowship of the Ring</td></tr>
<tr><td colspan="2">Exemplar Experiences from Debater 1
Understand User Preferences: Pay close attention to the examples and preferences provided by the user to tailor recommendations that align with their desired vibe or theme.
Emphasize Comfort and Warmth: Prioritize films that are uplifting, heartwarming, and comforting, especially when the user is seeking a cozy experience.

Exemplar Experiences from Critic
Understand User Preferences: Carefully analyze the user's query and any examples they provide to understand their preferences. Look for themes, genres, or specific qualities that the user is seeking in their movie recommendations.
Evaluate Alignment: Assess how well the recommended movies align with the user's preferences. Consider whether the films match the themes, emotional tone, and atmosphere that the user is looking for.</td></tr>
</table>