# RGCRS: A Retrieval-Guided Agent for Conversational Recommendation

Anonymous ACL submission

## Abstract

Large Language Models (LLMs) have rapidly redefined Conversational Recommender Systems (CRSs), but relying on their parametric memory invites hallucinated items and attributes, and simply exposing a dense retriever as a tool does not fix this. We identify two failure modes that limit retrieval-as-a-tool in CRS: *internal-prior dominance*, where the agent overrides or skips retrieval in favor of confident but unreliable parametric knowledge, and a *conversational query–corpus gap*, where off-the-shelf encoders attain only Recall@10≈0.1 on agent-generated queries across three retriever families. We propose RGCRS (Retrieval-Guided CRS), an autonomous agent that decides when to elicit preferences and how to query a retriever, trained sequentially via rollout-driven retriever adaptation followed by reinforcement learning. On ReDial and OpenDialKG, with both simulator and blind human evaluation, RGCRS outperforms strong baselines and exhibits two emergent behaviors not directly rewarded—strategic retrieval timing and improved factuality.

## 1 Introduction

Conversational Recommender Systems (CRSs) elicit user preferences through multi-turn natural language interaction and recommend items grounded in a domain database (Chen et al., 2019; Zhou et al., 2020; Wang et al., 2022b; Wei et al., 2025). Large Language Models (LLMs) have redefined this setting: a single instruction-tuned model can both produce fluent dialogue *and* reason about candidates from its parametric memory (He et al., 2023; Xie et al., 2024; Fang et al., 2024; Zhu et al., 2025; Yang and Fang, 2025). Yet this parametric reasoning is precisely where modern CRSs fail. As illustrated in Table 1, a popular open-weight LLM, asked for a 1990s thriller with Nicolas Cage as a serial killer, confidently asserts that Cage played the lead in *Seven (1995)*, contradicting the database's actual cast metadata.

**What specifically fails?** A natural response is to equip the LLM with a dense retriever over the item database, as in retrieval-augmented generation (RAG) (Lewis et al., 2020) or its CRS-specific variant (Yang and Fang, 2025). However, we identify two failure modes that retrieval-as-a-tool does not eliminate. First, *internal-prior dominance*: as shown in Table 1, the LLM's parametric memory can be confidently wrong on long-tail and niche compositional requests in CRS domains such as movies, books, and music—cases where retrieval, if invoked, would in principle help. Yet our analysis in §5.4 reveals that even with a retriever on-tap, the untrained agent under-invokes it even when invocation would help on average; and when it does retrieve, internal priors can still override the retrieved evidence in its chain-of-thought (CoT) (Wei et al., 2022) and responses. Second, a *conversational query–corpus gap*: simply plugging an off-the-shelf retriever into the CRS pipeline yields only marginal gains over the no-retrieval baseline. In §5.3 we observe that this phenomenon is retriever-agnostic, and hypothesize that the bottleneck lies in the gap between agent-formulated conversational queries and entity-centric item-side text.

**Challenge.** Tool-use agents such as ReAct (Yao et al., 2023), Toolformer (Schick et al., 2023), and Self-RAG (Asai et al., 2024) address *when* to invoke external tools, but within an agent's reasoning loop on a fixed user query. CRS demands the same decision over multi-turn user-facing dialogue, where the agent additionally faces an exploration–exploitation tradeoff between eliciting further preference and retrieving from the database. Mitigating the two failure modes above within this setting is the central challenge of this paper.

Table 1: Hallucination example (movie-domain) using the same backbone LLM (Qwen3-4B). Bold: hallucination. Vanilla LLM: Qwen3-4B with retrieval disabled. A second case (entity hallucination) is given in App. B.

| Scenario | Role | Dialogue Content |
|---|---|---|
| **Attribute Conflation** (Mismatch) | **User Intent** | 1990s dark psychological thriller starring Nicolas Cage as a serial killer |
| | **Vanilla LLM** | Are you thinking of *Seven (1995)*? **Nicolas Cage played John Doe, a serial killer.** Any other preferences? |
| | **RGCRS (ours)** | None of the 1990s films listed have Nicolas Cage as the serial killer. The closest are: *8MM (1999)* (investigation) and *Snake Eyes (1998)*. Would you like more options? |

**Central finding.** Our central empirical finding is that our trained agent exhibits two measurable emergent behaviors not directly incentivized by the recommendation reward: (i) *strategic retrieval timing*—a phase-aware policy that elicits preferences early and retrieves only after sufficient context (Table 4); and (ii) *factuality*—internal-prior dominance is suppressed in favor of retrieved evidence (Table 6). Notably, no behavior-specific signal supervises either: both arise as instrumental byproducts of a simple target-item recommendation reward.

**Method.** To produce and study this joint intervention, we construct **RGCRS** (Retrieval-Guided CRS), an autonomous agent operating under a simple *elicit/retrieve* action interface decided autonomously through CoT—which we call *Retrieval Guidance* (RG; Fig. 1). RG is trained in two stages: (1) *Rollout-driven Retriever Adaptation* (RRA) aligns the dense retriever to the empirical distribution of agent-generated queries via contrastive adaptation to rollout-generated queries, remaining architecture-agnostic (§5.3); (2) *policy refinement* with GRPO (Shao et al., 2024) then trains the agent, with the RRA-adapted retriever in the loop, to decide when to invoke retrieval and how to phrase the query, rewarding both recommendation quality and adherence to retrieved evidence.

We validate RGCRS on ReDial and OpenDialKG using both a user simulator (with leakage and rejection-rate controls; §4) and a blind *out-of-simulator* human evaluation (§5.7). RGCRS outperforms KG-based and LLM-based CRSs (including RAG-based) on standard recommendation metrics; ablations attribute the largest gain to RRA, with RL providing a further increment on top of the RRA-adapted retriever. Beyond the emergent behaviors above, RGCRS maintains appropriate responses on out-of-simulator user behaviors that recommendation metrics cannot capture, such as adversarial unfulfillable queries, sudden preference shifts, and item-explanation requests, as measured

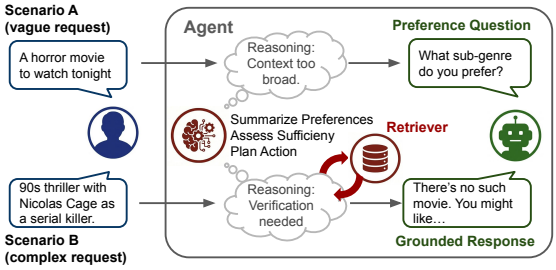


Figure 1: RGCRS uses CoT to autonomously choose between preference elicitation (Scenario A) and grounded retrieval (Scenario B), preventing attribute conflation.

by a blind human study. This robustness comes primarily from the RL stage; the gain on adversarial inputs directly measures the suppression of internal-prior dominance ($p < 0.05$).

**Contributions.**

- **Analytical.** We characterize two failure modes of retrieval-as-a-tool in CRS—*internal-prior dominance*, acute in the partial-knowledge domains CRS typically addresses, and a CRS-specific *conversational query–corpus gap*—and quantify the latter across three diverse retriever families.
- **Empirical finding.** We show that two emergent agentic behaviors (strategic retrieval timing, factuality) arise from RL refinement on top of the RRA-adapted retriever, and characterize them through behavioral statistics, LLM-as-a-judge ratings, and a blind out-of-simulator human evaluation.
- **Method.** We introduce RGCRS, combining Retrieval Guidance with a sequential RRA + RL pipeline; RRA itself, a contrastive retriever adaptation driven by agent rollouts, is architecture-agnostic.

## 2 Related Work

Conventional CRSs involved knowledge graphs to complement external knowledge (e.g., movie titles, genres) (Chen et al., 2019; Zhou et al., 2020;

Wang et al., 2022b; Wei et al., 2025). The emergence of LLMs has brought significant changes to CRS research. He et al. (2023) have demonstrated that LLMs can provide better recommendation in a zero-shot way only using their internal knowledge. Beyond a simple use of LLMs, additional mechanisms have been considered: combining LLMs with past interactions (Xie et al., 2024), planning and refinement for response generation based on multi-agent (Fang et al., 2024), improving zero-shot CRS using past recommendation failure experiences (Zhu et al., 2025). These methods use the external item database only for recommendation; no item knowledge is used for response generation. Yang and Fang (2025) proposed a framework that employs a dense retriever as an external database. Unlike RGCRS, however, the framework invokes a retriever at every turn and lacks the ability to synthesize responses through CoT.

Retrieval-Augmented Generation (RAG) (Lewis et al., 2020) mitigates hallucinations in LLMs, with recent *Tool-use Agents* (ReAct (Yao et al., 2023), Toolformer (Schick et al., 2023), Self-RAG (Asai et al., 2024)) further enabling autonomous decisions about *when* and *what* to retrieve. However, these general-purpose agents target QA or reasoning tasks, overlooking the elicitation-vs-retrieval dilemma unique to CRS, which our work addresses.

User responses are needed to evaluate and train under multi-turn settings. While iEvaLM (Wang et al., 2023) enables multi-turn evaluation, it suffers from unnatural target item leakage. Building on recent investigations into this issue (Zhu et al., 2024; Lucas de Carvalho et al., 2025; Kim et al., 2025), we introduce a robust simulator and a validation procedure in Sec. 4 to ensure reliable evaluation.

## 3 Method: RGCRS

**Setup.** This section formalizes the components from the Introduction: the dense retriever (RRA's target), the elicit/retrieve action set (governed by RG), and the per-turn CoT step that RL refines. We formulate the CRS task as a sequential decision-making problem involving an LLM agent, a user, and an external item database $\mathcal{I}$. Each item $i \in \mathcal{I}$ is identified by a text description (e.g., genre, director for movies). A trainable encoder $\phi_\omega$ maps queries and items into a shared latent space, and top-$k$ search against $\mathcal{I}$ via cosine similarity is denoted $\textsc{Top}_k(q; \mathcal{I}, \phi_\omega)$. The agent maintains a state $S_t = \{m_1, \ldots, m_t\}$ where each message $m_i$ originates from agent (A), user (U), or retriever (R); the latter two are treated as environment feedback. Upon receiving feedback, the agent decides its next action from $\mathcal{A} = \{\text{respond, call retriever}\}$ through CoT reasoning; agent messages always include a CoT field, $m_i = \{\text{role: A, CoT: } c, \ldots\}$. The conversation concludes when the user accepts a recommendation or a maximum turn $T_{\max}$ is reached; the agent's objective is to identify the user's target items $I^* \subset \mathcal{I}$, evaluated by ranking metrics.

We first introduce the inference mechanism, Retrieval Guidance (§3.1), then the rollout procedure (§3.2), and finally the sequential RRA + RL training pipeline (§3.3). An end-to-end trace illustrating the full agent–environment interaction is provided in Fig. 2 (Appendix A).

### 3.1 Retrieval Guidance

Retrieval Guidance (RG) lets the agent autonomously decide between generating a response and invoking the dense retriever via a search query $q$; when retrieval is invoked, the response is synthesized from retrieved results $\textsc{Top}_k(q; \mathcal{I}, \phi_\omega)$ to ground it in the item database.

**Calling Dense Retriever.** The agent is informed of the retriever through a function schema $F$ (Appendix H). Given state $S_t$, a single CoT pass simultaneously evaluates the necessity of external grounding and, if retrieval is deemed necessary, samples a query $q \sim LLM(S_t, F)$. The agent then emits a function call $m_{t+1} = \{\text{role: A, CoT: } c, \text{query: } q\}$ (cf. $m_4$ in Fig. 2). The environment executes $\textsc{Top}_k(q; \mathcal{I}, \phi_\omega) = \{i_1, \ldots, i_k\}$ with $k = 5$, yielding $m_{t+2} = \{\text{role: R, result: } \{i_1, \ldots, i_k\}\}$.

**Response Synthesis.** Given $m_{t+2}$, the agent again decides its next action via CoT. When responding to the user, it generates $m_{t+3}$ (cf. $m_6$ in Fig. 2) by reasoning over candidate-item attributes in $m_{t+2}$ against the conversation context.

### 3.2 Rollout/Inference

The interaction procedure serves a dual purpose: during training, it collects trajectory samples for policy optimization; at inference, it drives engagement with the user. From the initial state $S_0 = \{P_{\text{sys}}, C_0\}$ (system prompt and initial dialogue; Appendix H), the agent at each step reasons over $S$ via CoT and either invokes the retriever ($S \leftarrow S \oplus m_R$ with the top-$k$ result; reason again) or responds to the user (reply appended;

turn counter $t$ advances), terminating on user acceptance or $t = T_{\max}$. Algorithm 1 (Appendix C) gives the formal statement.

### 3.3 Training

We train in two sequential stages: first the dense retriever is adapted to agent-generated queries (RRA), then the agent's policy is refined with RL (GRPO) using the adapted retriever. Both stages use rollout samples generated by the user simulator (Sec. 4).

#### 3.3.1 Rollout-driven Retriever Adaptation

We expect the dense retriever to capture not only domain knowledge but also the linguistic patterns of agent-generated queries. To bridge this gap, we finetune the retriever on queries observed during exploration: we sample a user $u$ with initial state $S_0^u$, perform a rollout $S^u \sim \textsc{Rollout}(S_0^u)$, sample a final query $q_u \sim LLM(S^u, F)$ as in Sec. 3.1, and pair $q_u$ with the corresponding target items $i_u^*$. We finetune $\phi_\omega$ with the decoupled contrastive loss (Yeh et al., 2022) on the resulting $(q_u, i_u^*)$ pairs, with hard negatives sampled from $\textsc{Top}_{50}(q_u, \mathcal{I})$ under the retriever being trained.

#### 3.3.2 Reinforcement Learning

We then refine the agent's policy with GRPO (Shao et al., 2024), simultaneously training when to retrieve, how to phrase the query, and how to synthesize the response (Appendix F for parameter update). Because Recall@$k$ alone provides only a sparse signal, we add a dense semantic reward measuring the cosine similarity between the agent's query and the user's target item. For each user $u$, we generate $G$ rollouts $S^{u,g} \sim Rollout(S_0^u)$ and, post-rollout, sample an evaluation query $q^{u,g} \sim LLM(S^{u,g}, F)$ used solely for reward computation, with $I^{u,g} = \textsc{Top}_k(q^{u,g}, \mathcal{I})$. The reward is $r(S^{u,g}) = \sum_{k\in\{1,5,10\}} \mathrm{R@}k(I^{u,g}, i_u^*) + (\beta/|i_u^*|)\sum_{i\in i_u^*} \cos(q^{u,g}, i)$, where $\beta > 0$ is a constant; if the agent fails to call the retriever due to invalid formatting we assign the minimum reward $r = -\beta/|i_u^*|$. RRA is applied before RL because, without it, Recall@$k$ rewards remain too flat for policy gradients to climb (§5.3); the retriever is frozen during RL for simplicity.

## 4 User Simulator

Both our RRA + RL training (§3.3) and the experiments (§5) rely on agent–user rollouts, which require a simulator with two properties: (1) target-item leakage avoidance, so that the agent cannot bypass preference elicitation; and (2) feedback consistency, so that target items are accepted when recommended. Our simulator separates user profile from target-item knowledge: an intent classifier first routes the agent's utterance, and preference questions are answered conditioned solely on a leakage-controlled persona (LLM-generated, with target-item details stripped) while recommendations are accepted or rejected via an external ground-truth lookup—not via LLM internal knowledge—ensuring a $0\%$ reject rate for target items. We validate the design with two metrics, *Leak score* (LLM-as-judge, range 0–6) and *Reject rate*, against iEvaLM (Wang et al., 2023) and a profile-only baseline similar to Lucas de Carvalho et al. (2025). Our simulator achieves Leak score $0.91/1.43$ on ReDial/OpenDialKG with $0\%$ reject rate, improving over iEvaLM ($3.26/3.94$, $0\%$) and profile-only ($1.04/1.13$, $60\%/45\%$). Full design, prompts, and the per-dataset table are deferred to Appendix E.

## 5 Experiments

We evaluate RGCRS through experiments designed to answer four questions. **RQ1 (Overall Effectiveness)**: does the proposed framework improve recommendation accuracy over state-of-the-art baselines? **RQ2 (Component Contribution)**: to what extent do Retrieval Guidance (RG) and GRPO-based RL contribute? **RQ3 (Analysis of Agent Strategy)**: how does RL reshape the agent's policy? We analyze this from three complementary angles: (a) quantitative shifts in retrieval timing and query semantics (§5.4), (b) pairwise LLM-as-judge ratings of factuality and reasoning quality (§5.5), and (c) a case-level execution trace (§5.6). **RQ4 (Out-of-Simulator Robustness)**: does RGCRS maintain (or improve) its ability to respond appropriately on user behaviors outside the simulator's distribution?

### 5.1 Settings

**Datasets.** We evaluate RGCRS on two benchmark datasets for conversational recommendation: ReDial (Li et al., 2018) and OpenDialKG (Moon et al., 2019). For the ReDial dataset, we set target items as the movies liked but unseen by the seeker. We created user profiles for each example in these datasets to facilitate user simulation (Sec. 4).

**Evaluation Procedure.** Unlike standard evaluation protocols that measure turn-level accuracy

Table 2: Evaluation Results (upper) and Ablation Study (lower).

| Category | Method | ReDial | | | | OpenDialKG | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | R@1 | R@5 | R@10 | N@10 | R@1 | R@5 | R@10 | N@10 |
| Conventional CRSs | KBRD | 0.0083 | 0.0186 | 0.0302 | 0.0191 | 0.0881 | 0.1661 | 0.1932 | 0.1391 |
| | BARCOR | 0.0185 | 0.0920 | 0.1316 | 0.0791 | 0.1898 | 0.2593 | 0.2932 | 0.2409 |
| | UniCRS | 0.0378 | 0.1083 | 0.1606 | 0.1010 | 0.1612 | 0.2271 | 0.2689 | 0.2211 |
| | MSCRS | 0.0399 | 0.1211 | 0.1688 | 0.1102 | 0.1685 | 0.2324 | 0.2835 | 0.2383 |
| Vanilla LLMs | Qwen3-4B | 0.0209 | 0.0461 | 0.0686 | 0.0450 | 0.0610 | 0.0955 | 0.1209 | 0.0901 |
| | Qwen3-14B | 0.0444 | 0.0833 | 0.1036 | 0.0792 | 0.1316 | 0.1791 | 0.2181 | 0.1743 |
| | GPT-4.1 | 0.0480 | 0.1458 | 0.2173 | 0.1271 | 0.2627 | 0.4582 | 0.5514 | 0.4514 |
| LLM-based CRSs | ZSCRS | 0.0752 | 0.1767 | 0.2193 | 0.1566 | 0.3119 | 0.4644 | 0.4864 | 0.4020 |
| | NBCRS | 0.1077 | 0.2662 | 0.3508 | 0.2391 | 0.3910 | 0.5650 | 0.6243 | 0.5134 |
| | MACRS | 0.1355 | 0.2735 | 0.3614 | 0.2544 | 0.3831 | 0.5864 | 0.6746 | 0.5152 |
| | CRAVE | 0.0744 | 0.1810 | 0.2489 | 0.1717 | 0.4113 | 0.5994 | 0.6740 | 0.5367 |
| | ReGeS | 0.1245 | 0.2234 | 0.3016 | 0.1912 | 0.3898 | 0.6271 | 0.6780 | 0.5419 |
| | **RGCRS** (Ours) | **0.1734** | **0.3482** | **0.4084** | **0.3380** | **0.5785** | **0.6802** | **0.7073** | **0.6505** |
| | (Ablation Study) w/o RL | 0.1675 | 0.3264 | 0.3662 | 0.2909 | 0.4667 | 0.6056 | 0.6497 | 0.5655 |
| | w/o RRA, RL | 0.0492 | 0.0789 | 0.0985 | 0.0676 | 0.1638 | 0.2209 | 0.2497 | 0.2072 |
| | w/o RRA, RG, RL | 0.0209 | 0.0461 | 0.0686 | 0.0450 | 0.0610 | 0.0955 | 0.1209 | 0.0901 |

on fixed, static dialogue histories, our goal is to assess the agent's ability to autonomously steer the conversation. Since static evaluation cannot capture the cumulative impact of strategic retrieval and multi-turn planning, we define the initial state $S_0$ by extracting the beginning of the dialogue logs up to and including the first seeker response, and conduct a multi-turn conversation using the user simulator. Specifically, the interaction continues until a recommendation is accepted or the maximum turn limit ($T_{\max} = 3$) is reached.[1] To ensure a fair comparison, the same user simulator is employed across all baselines. We used OpenAI GPT-4.1-mini for the user simulator. Unless otherwise noted, we report the average Recall@$k$ ($k = 1, 5, 10$) and NDCG@10 measured at the conclusion of these initiated dialogues; decoding, run count, and statistical-testing protocol are detailed in Appendix D.

**Baselines.** We compare against three categories: (i) *conventional CRSs* (KBRD (Chen et al., 2019), BARCOR (Wang et al., 2022a), UniCRS (Wang et al., 2022b), MSCRS (Wei et al., 2025)), evaluated through the same user simulator; (ii) *vanilla LLMs* (Qwen3-4B, Qwen3-14B and GPT-4.1), evaluated under the same multi-turn protocol without retrieval to isolate purely parametric behavior; and (iii) *LLM-based CRSs* (ZSCRS (He et al., 2023), NBCRS (Xie et al., 2024), MACRS (Fang et al., 2024), CRAVE (Zhu et al., 2025), ReGeS (Yang and Fang, 2025)). For RGCRS, we use Qwen3-4B as the backbone and `all-mpnet-base-v2` as the dense retriever. Baseline standardization (LLM backbone and retriever), hyperparameters, and per-baseline implementation details are deferred to Appendix D.

**Compute.** Training RGCRS takes ∼9 h on two NVIDIA H200 GPUs (RRA ∼1 h, GRPO ∼8 h; peak 89 GB VRAM; Appendix D). Inference takes ∼0.45 s/turn on a single 24 GB GPU via vLLM, and the strategic policy skips the retriever in ∼58% of first turns (Table 4), yielding lower per-turn cost than always-retrieve RAG.

### 5.2 Main Results (RQ1)

Table 2 shows that RGCRS significantly outperforms all baselines on both datasets. Conventional KG-based CRSs perform poorly, constrained by supervised learning on narrow datasets and lacking the generalized semantic reasoning of LLMs. Vanilla LLMs (e.g., GPT-4.1) reason fluently but, without external grounding, hallucinate non-existent or irrelevant items from internal priors; the low Recall (Qwen3-4B R@1=0.02, GPT-4.1 R@1=0.05) quantifies the *partial but unreliable* item knowledge anticipated in §1—confident on popular items, confidently wrong on the long-tail and compositional requests typical of CRS. Among the LLM-based CRSs, RGCRS outperforms multi-agent consensus (MACRS), past-experience methods (NBCRS, CRAVE), and—crucially—the closest comparable baseline ReGeS, despite ReGeS using the same RAG paradigm.

[1] We selected this limit because empirical analyses of LLM-based simulators indicate that recommendation success rates saturate by the third turn, with subsequent turns offering diminishing returns (Zhu et al., 2024).

The gain over ReGeS reflects the methodological difference: ReGeS's non-autonomous, supervised retrieve-then-generate pipeline limits preference elicitation, whereas our RL-optimized agent dynamically balances elicitation and retrieval. Turn-wise ranking metrics (Appendix I) further show that recall improves with each turn and continues to improve beyond the training horizon, indicating a generalizable strategy rather than overfitting to a fixed episode length.

### 5.3 Ablation Study (RQ2)

We isolate the contributions of Retrieval Guidance (RG; the agent's capability to invoke the dense retriever and synthesize responses from external data), Rollout-driven Retriever Adaptation (RRA), and Reinforcement Learning (RL) by adding them one at a time (Table 2, lower block). Starting from the Vanilla Qwen3-4B backbone (w/o RRA, RG, RL; R@1=0.0209), turning on RG alone lets the agent invoke an off-the-shelf retriever (w/o RRA, RL), but lifts R@1 only to 0.0492, consistent with the conversational query–corpus gap (§5.3): a pretrained retriever returns too little relevant evidence for RG to exploit. The dominant step is RRA: replacing the pretrained retriever with the RRA-adapted one (w/o RL) lifts R@1 to 0.1675, a 3.4× gain over the RG-only variant, with R@10 rising from 0.0985 to 0.3662. GRPO-based RL on top of the RG+RRA stack yields the final increment to the full model. Beyond aggregate Recall@$k$ gains, RL induces favorable qualitative shifts in agent behavior, which we analyze in §5.4–§5.7.

**Robustness of RRA across retrievers.** Applying the identical RRA procedure to two retrievers that differ from `all-mpnet-base-v2` in both pretraining corpus and objective—Contriever (Izacard et al., 2022) and GTR-T5 (Ni et al., 2022)—yields Table 3. All three off-the-shelf encoders attain $R@10 \in [0.08, 0.11]$ on agent-generated queries, a consistent failure pattern that locates the bottleneck in the conversational query–corpus gap rather than encoder choice; RRA delivers a 2.2–3.5× recall gain on each, confirming architecture-agnosticism. A sensitivity analysis of the GRPO hyperparameters ($\beta$ and $G$) is shown in Appendix I.

### 5.4 Analysis of Agent Strategy (RQ3-a)

**Evolution of Retrieval Strategy.** Table 4 reports the turn-wise retriever calling ratio before and after RL. At Turn 1 both models retrieve infrequently, with only a modest increase after RL (35.0% → 41.6%); the Post-RL agent prioritizes preference elicitation when the initial context is insufficient. From Turn 2 onward a dramatic divergence appears: the Post-RL agent retrieves near-universally ($> 99\%$), while the Pre-RL agent plateaus around 75–79%. This phase-aware pattern—elicit early, retrieve after gathering context—is key to aligning recommendations with user intent. The Pre-RL plateau, on the other hand, is the behavioral signature of *internal-prior dominance*: even after the dialogue context warrants retrieval, the agent stays on its parametric memory in $\sim 25\%$ of turns.

Table 3: RRA across retrievers (R@10, ReDial).

| **Encoder** | w/o RRA | w/ RRA |
|---|---|---|
| `all-mpnet-base-v2` | 0.0985 | 0.3474 |
| Contriever | 0.0885 | 0.2631 |
| GTR-T5 | 0.1073 | 0.2399 |

Table 4: Per-turn retriever calling ratio (%).

| | Turn 1 | Turn 2 | Turn 3 |
|---|---|---|---|
| Pre-RL | 35.0 | 75.3 | 78.8 |
| Post-RL (Ours) | 41.6 | 99.6 | 99.4 |

**Evaluation of Semantic Query Quality.** To isolate dialogue strategy from query generation, Table 5 reports a counterfactual on ReDial: starting from RGCRS (RL/RL), we revert only the query model to its pretrained state (RL/Pretrained). The results reveal that even with a strategically optimized dialogue history, reverting the query model causes NDCG@10 to drop from 0.3380 to 0.3143. This degradation confirms that RL enhances the agent's proficiency in "tool-use", allowing it to formulate search terms with higher semantic precision (0.4811 vs 0.4703) on identical contexts.

### 5.5 Qualitative Evaluation (RQ3-b)

The quantitative results in Sec. 5.2 and Sec. 5.3 confirm performance gains, and the behavioral statistics in Sec. 5.4 reveal the mechanism: the Post-RL agent generates search queries that exhibit higher cosine similarity with the target item descriptions. This suggests that the agent has learned to "speak the language" of the retriever, aligning its query representations with the text embedding space.

Semantic alignment is not a measure of reasoning quality. To validate that this similarity stems from **strategic intent understanding** rather than pattern matching, we conduct a multi-dimensional pairwise evaluation using GPT-5.2 as a judge. We

Table 5: Cross-Evaluation Analysis: RL improves both rollout strategy and query generation capabilities.

| Rollout strategy | Query model | Cos | N@10 |
|---|---|---|---|
| Pre-RL | Pre-RL | 0.4661 | 0.2909 |
| Post-RL (ours) | Pre-RL | 0.4703 | 0.3143 |
| Post-RL (ours) | Post-RL (ours) | **0.4811** | **0.3380** |

Table 6: LLM-based Qualitative Evaluation (Win rate). $^{***}p<0.001$, $^{**}p<0.01$ (binomial test).

| | **Phase 1 (Query)** | | **Phase 2 (Response)** | |
|---|---|---|---|---|
| **Metric** | **Post-RL** | **Pre-RL** | **Post-RL** | **Pre-RL** |
| Factuality | **96%**$^{***}$ | 4% | **82%**$^{***}$ | 18% |
| Constraints | **87%**$^{***}$ | 13% | **65%**$^{**}$ | 35% |
| Logic | **97%**$^{***}$ | 3% | **70%**$^{***}$ | 30% |
| **Overall** | — | — | **82%**$^{***}$ | 18% |

compare 100 dialogue sessions where the Pre-RL (with RRA and RG) and Post-RL (RGCRS) agents start from the same dialogue history.

The judge is presented with the full execution trace—CoT, generated query, retrieved metadata, and final response—for both models in a blinded, randomized order, and rates each pair on four dimensions (see Fig. 5 for the actual prompt): *factuality & grounding* (faithfulness to retrieved metadata; absence of hallucinated titles or plots), *constraint* (adherence to user requirements), *logic* (logical flow of CoT and response), and *overall* (holistic quality). To isolate the impact of RL on the agent's reasoning process, we evaluate at two distinct stages: the Query Generation phase (**Phase 1**: CoT and generated query) and the Final Response phase (**Phase 2**: CoT and synthesized response against the retrieved evidence). Table 6 summarizes these results. The results indicate a decisive shift in how the agent invokes the retriever and generates a response.

**Tool-use Failure and Downstream Bottleneck.** The Pre-RL baseline shares the same RG mechanism as Post-RL but fails to wield it: its low Phase 1 win rates (Factuality 4%, Logic 3%) reflect *internal-prior dominance*—reasoning capacity spent retrieving imperfect parametric knowledge rather than formulating a query, leaving RG ineffective. The gap narrows in Phase 2 (Logic 30%, Constraint 35%): once retrieved metadata is available, the baseline can perform ad-hoc constraint checking with its latent reasoning capability. This recovery, however, is bounded by upstream query quality—the Phase 1 query routinely retrieves irrelevant items, so downstream logic is applied to the wrong evidence. By securing high-quality evidence first, Post-RL maximizes downstream reasoning utility, yielding the **82% Overall win rate**.

### 5.6 Case Study (RQ3-c)

**Strategic Evolution via RL.** We here investigate the qualitative shift in agent behavior. Using the execution trace in Table 7 (the user is searching for a "poignant zombie movie"), we analyze how RL refines decision-making.

The Pre-RL agent reveals a lack of discipline, easily distracted by irrelevant internal knowledge. For instance, it wanders into a hallucination, misidentifying the crime drama *No Country for Old Men* as a zombie film. Preoccupied with this internal noise, it fails to focus on query precision, omitting the user's primary constraint ("zombie") and retrieving unrelated titles.

In contrast, the Post-RL agent demonstrates a disciplined focus on query optimization. Its internal reasoning identifies the request's core pillars—"poignant drama" and "avoid jump scares"—and concludes it must "emphasize emotional depth over horror." By suppressing irrelevant associations, it produces a query that incorporates user requirements. This transition confirms that the RL process trains the agent to prioritize query construction over its own linguistic priors.

**Effect of RG.** Beyond RL's strategy shift, RG itself is what suppresses internal-prior hallucination in niche or mismatched scenarios: in Table 1 (Nicolas Cage in *Seven*) the no-retrieval baseline conflates actor–role tokens, and in the entity-hallucination case (App. B) the same baseline fabricates a non-existent title, whereas RGCRS verifies retrieved metadata and either abstains or corrects.

### 5.7 Out-of-Simulator Behavior (RQ4)

To verify that the simulator-measured gains are not artifacts of the LLM-simulator, we conducted a blind human evaluation on five interaction categories deliberately under-represented in the simulator's distribution: *Preference Shift* (PS; a mid-dialogue change in user intent), *Reason Explanation* (RE; justifying why an item fits), *Item Explanation* (IE; clarifying a retrieved item's content), *Emotional Request* (Emo; affect-laden queries), and *Adversarial Queries* (AQ; intentionally unfulfillable constraints). For each category we manually authored dialogue prompts and obtained responses from Pre-RL (with RRA and RG, no RL)

Table 7: End-to-end Execution Trace (Pre-RL (Baseline) vs. RGCRS (Ours)). RL optimization improves both the strategic formulation of search queries and the faithfulness of the final recommendation to the retrieved evidence.

| **Stage** | **Pre-RL (with RRA and RG)** | **RGCRS (Post-RL)** |
|---|---|---|
| **Context** | User: "I want a zombie movie that isn't a horror film." → Assistant: "So something more like a comedy or a poignant drama?" → User: "A poignant drama. I want to avoid the 'jump scares' and focus on the emotions." | |
| **Phase 1** | **CoT:** [*Severe Hallucination*] "...Oh, 'No Country for Old Men' has zombies but is a crime drama. ... 'The Survival' (2014) is a drama about..."<br>**Query:** "poignant drama about humanity, avoiding jump scares, focusing on emotions"<br>→ Lack of the keyword "zombie" | **CoT:** [*Strategic Grounding*] "...Key points are 'poignant drama' and 'avoid jump scares'. I'll structure the query to emphasize emotional depth over horror."<br>**Query:** "poignant drama zombie movie focusing on emotions, avoiding jump scares" |
| **Retrieved** | The Awakening, The Village, Home, It Comes at Night, They Look Like People | Warm Bodies, American Zombie, Train to Busan, The Girl with All the Gifts, The Dead |
| **Phase 2** | **CoT:** [*Ad-hoc Correction*] "...The user wanted a zombie movie... none of these are zombie movies. ... 'The Awakening' is about ghosts... maybe that is the best fit here."<br>**Final Resp.:** "Here's a poignant drama: **The Awakening (2011)**. It explores grief... through a supernatural lens." | **CoT:** [*Grounded Analysis*] "...Warm Bodies has drama elements... The Girl with All the Gifts is a drama... I should present the ones that best fit."<br>**Final Resp.:** "Here are some poignant drama-style zombie movies: 1. Warm Bodies, 2. The Girl with All the Gifts, 3. American Zombie." |
| **Outcome** | **Failure**: Hallucinates genre (P1); substitutes a ghost movie for the zombie request (P2). | **Optimal**: Maintains strict consistency; provides high-quality zombie dramas as requested. |

Table 8: Out-of-simulator human evaluation (appropriateness %). $^*p<0.05$ (binomial test).

| **Method** | PS $n$=22 | RE $n$=21 | IE $n$=21 | Emo $n$=28 | AQ $n$=50 | Tot. $n$=142 |
|---|---|---|---|---|---|---|
| Vanilla LLM | — | — | — | — | 22% | — |
| Pre-RL | 95% | 71% | 100% | 54% | 60% | 72% |
| RGCRS | 91% | **90%** | 100% | 54% | **80%**$^*$ | **81%** |

and Post-RL (RGCRS) under the same dialogue context; on AQ we additionally evaluated Vanilla LLM (Qwen3-4B, no retrieval) as the no-grounding floor. Appropriateness rating requires fact verification beyond non-expert scope, so one of the authors, blinded to model identity and presentation order, rated each response as *appropriate* or *not appropriate* against a category-specific criterion (protocol, criteria, and representative dialogues in Appendices J, L). We report the appropriateness rate over 142 dialogue prompts (Table 8); for AQ, we draw five independent responses per (prompt, model) pair to account for sampling variance and report per-prompt counts in Appendix K.

The headline result is on adversarial queries: appropriateness rises from 60% (Pre-RL) to 80% (RGCRS), a 20-point gain significant at $p < 0.05$. This directly measures the suppression of *internal-prior dominance*: when no item satisfies the constraints, the prior-dominated agent confabulates a fit, whereas RGCRS abstains and offers grounded alternatives. The Vanilla LLM ceiling of 22% isolates retrieval grounding as the floor that makes this gain reachable. The AQ gain comes with no degradation on the four breadth-check categories (PS/RE/IE/Emo): PS differs by a single sample (21/22 vs. 20/22, not significant), IE and Emo are identical, and RE shows an upward trend (71% → 90%). Together with no degradation on the breadth-check categories, the AQ gain indicates that RL's policy generalizes beyond the simulator distribution rather than overfitting to it.

## 6 Conclusion

We presented RGCRS, an autonomous agent that recasts conversational recommendation as a strategic tool-use problem: Retrieval Guidance lets the LLM decide when and how to consult an external item database, and a two-stage pipeline—Rollout-driven Retriever Adaptation followed by GRPO with dense semantic rewards—bridges the conversational-query/item-embedding gap and supplies a learning signal in an otherwise sparse setting. On ReDial and OpenDialKG, RGCRS outperforms strong baselines. Beyond these aggregate gains, analysis identifies two emergent behaviors not directly rewarded—strategic retrieval timing and improved factuality—validated through behavioral statistics, LLM-as-judge ratings, case studies, and a blind out-of-simulator human evaluation, indicating that the RL policy refines along axes orthogonal to the simulator's coverage.

## Limitations

**Reliance on an LLM-based user simulator.** LLM-based user simulators are standard practice for multi-turn CRS evaluation (Wang et al., 2023; Zhu et al., 2024; Lucas de Carvalho et al., 2025), and we follow this convention for both RL reward computation and aggregate quantitative evaluation. The concern that gains may overfit to the simulator is addressed directly: target-leakage and rejection-rate calibration (Sec. 4) reduce on-policy bias, and a blind out-of-simulator human evaluation (Sec. 5.7) tests the trained policy on five interaction categories deliberately chosen to probe behavior beyond the simulator's training distribution—preference shift, reason/item explanation, emotional request, and adversarial unfulfillable queries—on which RL delivers a $60\% \rightarrow 80\%$ appropriateness gain on the adversarial category ($p < 0.05$). Residual limitations remain: the human study covers 142 dialogue turns in English only and was scored by a single author rater (justified by the need for per-prompt fact verification; blinding and release protocol in App. J); we do not measure long-horizon satisfaction, repeated engagement, or robustness to real distribution shift.

**Scope of attribution for emergent behaviors.** We report component ablations (RRA / RG / RL on/off) on aggregate ranking metrics in Table 2, and analyze the two emergent behaviors at Pre-RL vs. Post-RL granularity (Tables 4, 6). We do not decompose the RRA contribution *within* each emergent behavior, nor evaluate query precision (Table 5) outside the RRA-adapted embedding space. This is a deliberate consequence of the design rather than an oversight: without RRA, off-the-shelf encoders attain only R@10≈0.1 (§5.3), leaving the retrieval-mediated evidence path effectively non-functional, so behavioral metrics defined over retriever outputs—when to invoke, whether to follow retrieved evidence—lose discriminative signal. We therefore treat RRA as a prerequisite for the rollout distribution to be analyzable, and report emergent-behavior shifts conditional on it. A finer 2×2×2 grid (rollout strategy × query model × retriever) over a corpus where off-the-shelf encoders are competitive is left to future work.

**Computational requirements.** The two-stage training (RRA + GRPO; details in App. D) requires multiple high-VRAM GPUs upfront—a cost shared with most RL-finetuned LLM systems—which may limit applicability when frequent policy retraining is required. At inference, however, the trained policy partially amortizes this overhead: by skipping retrieval in a large fraction of first turns, it yields lower per-turn latency than always-retrieve RAG (§5.1).

**Domain coverage.** Our empirical study uses ReDial and OpenDialKG. Much of the CRS literature has historically developed around knowledge-graph-based methods, and these two benchmarks are natural choices for like-for-like comparison since their knowledge-graph infrastructure is already established in the literature. Extending and evaluating RGCRS on non-movie domains (e.g., books, music, dining) are interesting topics for future work. The framework itself is more loosely coupled to the domain—neither Retrieval Guidance nor the sequential RRA+RL pipeline encodes movie-specific structure, and the user simulator's intent classifier (Sec. 4) operates on generic recommendation/preference labels—so deploying it to a new domain requires only re-running RRA and RL on that domain's simulator and item corpus (∼9 H200-hours in our setting). In the absence of cross-domain CRS benchmarks, however, the zero-shot transferability of a trained policy—and whether the emergent retrieval-timing behavior carries over to domains with sparser metadata or different conversational norms—remains an open empirical question.

**Backbone and language scope.** We use Qwen3-4B as the agent backbone. The internal-prior dominance phenomenon we characterize is plausibly backbone-dependent: stronger or differently-aligned models may exhibit different failure modes, and could conceivably require less RL correction. All datasets, simulator prompts, and human evaluation interactions are in English. Whether the same elicit-versus-retrieve dilemma manifests in other languages, and whether RRA aligns retrievers equally well for non-English item corpora, is left to future work.

**Reward design and hyperparameter sensitivity.** Our reward combines Recall@$k$ at $k \in \{1, 5, 10\}$ with a cosine-similarity term scaled by $\beta$. A sensitivity analysis (App. I) shows that recall remains stable across $\beta \in \{0, 0.5, 1\}$ and across rollout group sizes $G \in \{4, 16\}$, but we have not searched the reward space systematically nor explored alternative formulations (e.g., diversity penalties; see

Ethical Considerations below). The interaction between reward shape and the emergent retrieval-timing policy is not yet understood.

## Ethical Considerations

**Filter bubbles and recommendation diversity.** Like prior CRS architectures, RGCRS is optimized for preference matching, which can reinforce a user's existing tastes at the expense of discovery. Our LLM-driven retrieval may amplify this risk: if the agent's chain-of-thought distills user preferences into a small set of high-precision query patterns, the retrieved candidate sets become tightly clustered around the user's prior preferences. The current reward function does not penalize concentration; introducing a diversity or exploration term in the reward—e.g., a $-\lambda \cdot \text{intra-list similarity}$ component—is a natural and inexpensive follow-up that practitioners deploying RGCRS should consider.

**Privacy in autonomously formulated queries.** Because RGCRS lets the agent autonomously phrase retrieval queries from accumulated conversational context, sensitive user-disclosed information (emotional state, demographics, identifying details) can flow into the query stream and reach the retrieval backend. In deployment scenarios where retrieval is performed by an external service, this constitutes a privacy surface that does not exist for static retrieve-then-generate pipelines, in which queries can be filtered at fixed points. A pragmatic mitigation is to instruct the agent's CoT to strip personally identifying content before emitting a function call. We did not implement or evaluate such a filter; we flag this as a deployment-time concern.

## References


Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2024. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations*.

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)*, pages 1803–1813, Hong Kong, China. Association for Computational Linguistics.

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

Zhankui He, Zhouhang Xie, Rahul Jha, Harald Steck, Dawen Liang, Yesu Feng, Bodhisattwa Prasad Majumder, Nathan Kallus, and Julian J. McAuley. 2023. Large language models as zero-shot conversational recommenders. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, CIKM 2023, Birmingham, United Kingdom, October 21-25, 2023*, pages 720–730. ACM.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022. Unsupervised dense information retrieval with contrastive learning. *Transactions on Machine Learning Research*.

SungHwan Kim, Kwangwook Seo, Tongyoung Kim, Jinyoung Yeo, and Dongha Lee. 2025. Stop playing the guessing game! evaluating conversational recommender systems via target-free user simulation. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 19588–19605. Association for Computational Linguistics.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Raymond Li, Samira Ebrahimi Kahou, Hannes Schulz, Vincent Michalski, Laurent Charlin, and Chris Pal. 2018. Towards deep conversational recommendations. In *Advances in Neural Information Processing Systems 31 (NIPS 2018)*.

Gustavo Adolpho Lucas de Carvalho, Simon Benigeri, Jennifer Healey, Victor Bursztyn, David Demeter, and Lawrence Birnbaum. 2025. A flash in the pan: Better prompting strategies to deploy out-of-the-box LLMs as conversational recommendation systems. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 8385–8398. Association for Computational Linguistics.

Seungwhan Moon, Pararth Shah, Anuj Kumar, and Rajen Subba. 2019. Opendialkg: Explainable conversational reasoning with attention-based walks over knowledge graphs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernandez Abrego, Ji Ma, Vincent Y. Zhao, Yi Luan, Keith B. Hall, Ming-Wei Chang, and Yinfei Yang. 2022. Large dual encoders are generalizable retrievers. In *Proceedings of the 2022 Conference on*

*Empirical Methods in Natural Language Processing (EMNLP)*, pages 9844–9855.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessí, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: language models can teach themselves to use tools. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y.K. Li, Y. Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2403.03952*.

Ting-Chun Wang, Shang-Yu Su, and Yun-Nung Chen. 2022a. Barcor: Towards a unified framework for conversational recommendation systems. *Preprint*, arXiv:2203.14257.

Xiaolei Wang, Xinyu Tang, Xin Zhao, Jingyuan Wang, and Ji-Rong Wen. 2023. Rethinking the evaluation for conversational recommendation in the era of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10052–10065. Association for Computational Linguistics.

Xiaolei Wang, Kun Zhou, Ji-Rong Wen, and Wayne Xin Zhao. 2022b. Towards unified conversational recommender systems via knowledge-enhanced prompt learning. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1929–1937.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Yibiao Wei, Jie Zou, Weikang Guo, Guoqing Wang, Xing Xu, and Yang Yang. 2025. Mscrs: Multi-modal semantic graph prompt learning framework for conversational recommender systems. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '25, page 42–52, New York, NY, USA. Association for Computing Machinery.

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 *Proceedings of the 18th ACM Conference on Recommender Systems*, RecSys '24, page 1045–1050, New York, NY, USA. Association for Computing Machinery.

Dayu Yang and Hui Fang. 2025. Reges: Reciprocal retrieval-generation synergy for conversational recommender systems. *Preprint*, arXiv:2509.21371.

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 *International Conference on Learning Representations (ICLR)*.

Chun-Hsiao Yeh, Cheng-Yao Hong, Yen-Chi Hsu, Tyng-Luh Liu, Yubei Chen, and Yann LeCun. 2022. Decoupled contrastive learning. In *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVI*, page 668–684, Berlin, Heidelberg. Springer-Verlag.

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*, KDD '20, page 1006–1014, New York, NY, USA. Association for Computing Machinery.

Lixi Zhu, Xiaowen Huang, and Jitao Sang. 2024. How reliable is your simulator? analysis on the limitations of current llm-based user simulators for conversational recommendation. WWW '24, page 1726–1732. Association for Computing Machinery.

Yaochen Zhu, Harald Steck, Dawen Liang, He Yinhan, Nathan Kallus, and Jundong Li. 2025. Llm-based conversational recommendation agents with collaborative verbalized experience. In *EMNLP Findings*.

## A Detailed System Overview

Figure 2 traces a two-turn dialogue end-to-end to make the agent–environment interaction concrete. In Turn 1 the user issues a broad request and the agent's CoT decides that the context is too vague for retrieval; it responds with a clarifying question and does *not* call the retriever ($m_2$). In Turn 2 the user supplies an anchor ("21 Jump Street"), the CoT judges that the context is now sufficient, and the agent emits a function call with a synthesized query ($m_4$); the retriever returns top-$k$ items ($m_5$), which the agent grounds into the final recommendation ($m_6$). The trace illustrates the two trainable components in their training roles: the agent (LLM) is updated by reinforcement learning over full rollouts, while the retriever is updated by Rollout-driven Retriever Adaptation (RRA) using the agent's own queries as the query distribution.

## B Second Hallucination Example

In addition to the attribute-conflation case shown in Table 1 of Sec. 5.4 (intro), Table 9 shows a second

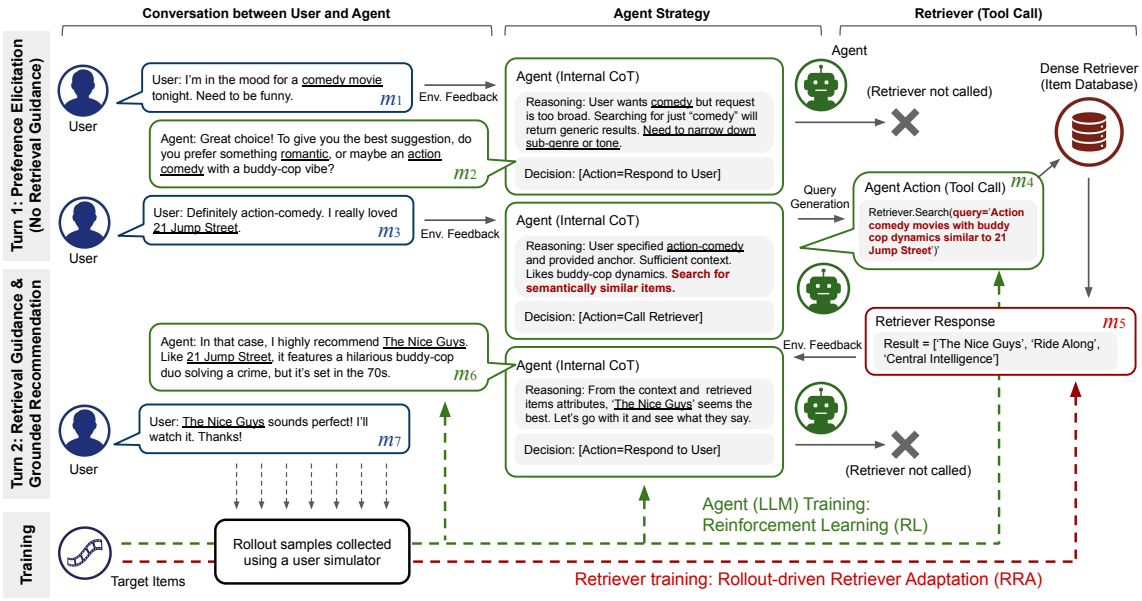


Figure 2: Detailed overview of RGCRS. The agent autonomously determines whether to call the dense retriever through CoT. Agent actions at step $t+1$ are made based on the state $S_t = \{m_1, \cdots, m_t\}$, where $m_i$ $(i = 1, \cdots, 7)$ is a message originating from either user, agent, or retriever. Messages from user and retriever are treated as environment feedback. Both the agent and retriever are trained using rollout samples via RL and RRA, respectively.

representative failure mode—entity hallucination, in which the vanilla LLM fabricates a non-existent item to satisfy a niche genre request.

## C Dialogue Rollout Algorithm

Algorithm 1 formalizes the interaction procedure between the agent and the environment (user and retriever) referenced in Sec. 3.2.

**Algorithm 1** Dialogue Rollout Procedure

**procedure** ROLLOUT($S_0$) *# $S_0$: Initial state with system prompt and user utterance*
  $S \leftarrow S_0$, $t \leftarrow 0$
  **while** $t < T_{\max}$ **do** *# $T_{max}$: Maximum allowed turns*
    $S, t \leftarrow$ STEPROLLOUT($S, t$)
  **return** $S$
**procedure** STEPROLLOUT($S, t$)
  $m_A \leftarrow$ agent.respond($S$) *# Agent decides action via CoT*
  $S \leftarrow S \oplus m_A$ *# Add agent message*
  **if** $m_A$ invokes a query $q$ **then** *# Agent Decision: Call retriever*
    $R(q) \leftarrow$ retriever.search($q$) *# Dense retrieval*
    $m_R \leftarrow$ {role: R, result: $R(q)$}
    $S \leftarrow S \oplus m_R$ *# Add retriever feedback*
  **else** *# Agent Decision: Respond to user*
    $m_U \leftarrow$ user.respond($S$) *# Obtain user response*
    $S \leftarrow S \oplus m_U$ *# Add user response*
    $t \leftarrow t + 1$ *# Increment turn after user feedback*
    **if** $m_U$ accepts $m_A$ **then** *# If user is satisfied,*
      **return** $S, T_{\max}$ *# then dialogue ends*
  **return** $S, t$

## D Implementation Details

**RGCRS (ours).** We use Qwen3-4B as the agent backbone and `all-mpnet-base-v2` as the dense retriever. Retrieval Guidance is implemented via the function-calling mechanism of the Qwen3 series. For Rollout-driven Retriever Adaptation (RRA), we sample 5,000 rollouts from the training set and train for 5 epochs with the Adam optimizer at learning rate $10^{-5}$; the temperature in the decoupled contrastive loss is set to $\tau = 0.04$. For reinforcement learning, we use GRPO with learning rate $5\times10^{-5}$, batch size 512 (with gradient accumulation), group size $G = 16$, LoRA $\alpha = 16$ and rank $r = 8$, and reward weight $\beta = 0.5$; we run 100 steps of Adam. Multi-turn interactions are managed with the `verifiers` library (v0.1.7.post0). Training uses two NVIDIA H200 (140 GB) GPUs.

**Evaluation protocol.** At evaluation time, both the agent and the user simulator decode with temperature $T = 0.7$; main-table metrics correspond to a single rollout per test dialogue, aggregated over all test dialogues. Statistical comparisons for the LLM-judge and human evaluations use a binomial test, with $p$-values reported in Tables 6 and 8.

**Baseline standardization.** For conventional CRS baselines (KBRD, BARCOR, UniCRS, MSCRS), we re-evaluate the original models through interactions with our user simu-

Table 9: Second hallucination example: entity hallucination on a niche genre (Qwen3-4B backbone). Bold: hallucination. Baseline: Qwen3-4B with retrieval disabled (no-RG ablation).

| Scenario | Role | Dialogue Content |
|---|---|---|
| **Entity Hallucination** (Niche Genre) | **User Intent** | 1920s sci-fi comedy movies |
| | **Vanilla LLM** | **"The Man Who Dreamed of the Moon" (1927) is a silent sci-fi comedy** about a man's whimsical quest to reach the moon. Would you like more examples or clarification on its style? |
| | **RGCRS (ours)** | I found some films, but most aren't from the 1920s. For example, "A Trip to the Moon" (1902) and "Abbott and Costello Go to Mars" (1953) are outside the 1920s. Would you like me to refine the search for strictly 1920s-era sci-fi comedies or explore films with a 1920s aesthetic? |

lator, departing from their original turn-level fixed-history evaluation. For vanilla LLMs, the retriever is `all-mpnet-base-v2` for open-weight backbones (Qwen3-4B/14B) and OpenAI `text-embedding-3-large` for the closed-weight backbone (GPT-4.1). For LLM-based CRSs (ZSCRS, NBCRS, MACRS, CRAVE, ReGeS), we follow the original implementations but standardize the LLM backbone to GPT-4.1 and the retriever to `text-embedding-3-large` (or `all-mpnet-base-v2` for methods that finetune the retriever); for methods that do not include a dedicated response generator, GPT-4.1 is used for response generation.

**Artifact licenses.** All datasets and models used in this work permit academic research use. ReDial is released under CC BY 4.0; OpenDialKG under CC BY-NC 4.0. Qwen3 (Apache 2.0), `all-mpnet-base-v2` (Apache 2.0), GTR-T5 (Apache 2.0), and Contriever (CC BY-NC 4.0) are used under their respective licenses. GPT-4.1 and `text-embedding-3-large` are accessed via the OpenAI API under its terms of service. Our use is consistent with the intended academic-research use of all artifacts. Derivatives created in this work (the RRA-adapted retrievers and the GRPO-trained agent) are intended for academic research only; any future release would inherit the most restrictive upstream license of its constituent artifacts.

**Dataset documentation.** We use the standard splits of two English movie-recommendation dialogue datasets. **ReDial** (Li et al., 2018) was collected on Amazon Mechanical Turk between paired crowdworkers (a seeker and a recommender) and contains 11,348 dialogues (10,006 train / 1,342 test) and 182,150 utterances over a catalog of 6,486 unique movies. **OpenDialKG** (Moon et al., 2019) contains 13,802 human–human dialogues (91,209 turns) grounded in a knowledge graph of 100,813 entities and 1,358 relations, spanning four domains (movie, book, sports, music); we use the movie-domain subset. Each item is represented by a textual description (title, genre, director, principal cast, short plot). Neither dataset records seeker demographic attributes; demographic coverage is therefore not reportable. All dialogues are in English.

**PII and offensive content.** ReDial and OpenDialKG are public movie-domain dialogue datasets widely used in CRS research; their public release implies prior vetting by the dataset creators, and the movie domain does not solicit personally identifying information from users. Our human evaluation does not collect new data from external participants—all 142 dialogue prompts were manually authored by an author of this paper and were rated by an author (§5.7). We inspected our authored prompts and observed no offensive content or identifying references to real individuals.

**Use of AI assistants.** The authors used AI assistants (large language models) for writing assistance (paraphrasing and copy-editing). All research ideas, experimental design, analysis, and final wording decisions are the authors'. AI assistants were not used to generate research content, analyze experimental data, or write the methods/results sections. The role of LLMs as artifacts of study (backbone agent, user simulator, LLM-as-judge) is described in the main text and elsewhere in this appendix.

## E User Simulator: Design and Validation Details

Here we elaborate on the user simulator briefly described in Sec. 4.

**Simulator Design.** The simulator adopts a modular architecture that separates user profile from target-item knowledge:

- **User Profile.** We generate a persona using an LLM that reflects general preferences and current interests subtly, without mentioning uniquely-identifiable item details. Concrete

Table 10: Comparison of User Simulators (ReDial vs. OpenDialKG). "Leak" (score 0–6, lower is better) and "Reject" ratio.

| | ReDial | | OpenDialKG | |
|---|---|---|---|---|
| Method | Leak | Reject | Leak | Reject |
| iEvaLM | 3.26 | 0% | 3.94 | 0% |
| Profile-only | 1.04 | 60% | 1.13 | 45% |
| Our simulator | 0.91 | 0% | 1.43 | 0% |

prompts and an example profile are shown in Appendix G.

- **Conditional Generation.** The simulator first performs intent recognition on the agent's utterance. For preference questions, it generates responses based solely on the profile to minimize leakage. For recommendations, it determines acceptance using external ground-truth data rather than the LLM's internal knowledge, ensuring a $0\%$ reject rate for target items.

**Validation Procedure.** We validate the simulator using "Leak score" (LLM-as-a-judge; range 0–6, lower is better) and "Reject rate" (proportion of simulator responses that fail to accept an agent message recommending a target item). We compare our approach with two simulators: (1) iEvaLM (Wang et al., 2023), which includes target item information directly in the prompt, and (2) a "profile-only" baseline, which is a single-prompt simulator similar to Lucas de Carvalho et al. (2025) without conditional generation.

As shown in Table 10, iEvaLM suffers from high leakage, while the profile-only model frequently rejects target items (45–60%) due to the lack of conditional generation. In contrast, our simulator achieves a $0\%$ reject rate while maintaining low leakage across both datasets. This ensures a reliable environment for RL where agents are rewarded for genuine discovery rather than unintended hints.

The leak score is computed via GPT-5.2 using the prompt shown in Figure 4; both metrics are averaged over 100 randomly sampled rollouts.

# F Parameter Update via Reinforcement Learning

The model parameters are updated to increase the probability of high-reward reasoning/action paths. The GRPO loss for the rollouts $S^u := \{S^{u,g}\}_{g=1}^{G}$ is defined as follows:

$$\mathcal{L}(\theta) = -\frac{1}{G}\sum_{g=1}^{G}\sum_{t=1}^{|o_g|}\left[m(o_{g,t})\frac{\pi_\theta(o_{g,t})}{[\pi_\theta(o_{g,t})]_{\text{no grad}}}\hat{A}_g\right], \tag{1}$$

where $o_{g,t}$ is the $t$-th token in $S^{u,g}$, and $m(\cdot)$ is a mask function defined as 1 for tokens generated by the agent and 0 otherwise. $\pi_\theta(o_{i,t}) = \pi_\theta(o_{i,t}|o_{i,<t})$ is the token generating probability where $o_{i,<t}$ is the tokens preceding $t$. $\pi_{\text{ref}}$ is the reference probability, "$[\cdot]_{\text{no grad}}$" means function computation without backprop. Then, we compute the *advantage*, denoted as $\hat{A}_i$, by normalizing the rewards as follows: $\hat{A}_i = (r(o_i) - \mu)/\sigma$ where $\mu$ and $\sigma$ are the mean and standard deviation of $r(o_i)$ over $G$ samples.

# G User Simulator

## G.1 User Profile Generation

The prompt below was used to generate user profiles. Here, `{movie_info}` is a placeholder for movie information including movie title and meta information, such as genre, director, actors, and short plot. For ReDial dataset, we used the movies in the following three categories to create user profiles based on the annotation: (1) Watched and liked, (2) watched and disliked, (3) haven't watched and liked (current interest, target movies). Movies in the OpenDialKG dataset do not have such annotations. We simply used all movies in the dialogue. GPT-4.1 was used to create user profiles.

**Prompt for user profile generation**

```
# Task
Create a user persona based on the movie
information and dialogue below.

# Movie Information
{movie_info}

# Dialogue
{dialogue}

# Instructions
- Do not contain the movie titles and their
  specific details in the user persona.
- The persona is used to simulate conversations
  about movie recommendations to find the
  target items.
- The persona should reflect the user's general
  movie preferences as well as the current
  interests indicated by the target items.
  The current interests should be subtly
  implied without explicitly mentioning the
  target item information that is uniquely
  identifiable.
```

The following is a user profile example for a sample in the ReDial dataset (Target item: Avengers: Infinity War):

> Epic Adventure Seeker: This user is passionate about high-stakes adventures and immersive action across diverse settings, from futuristic worlds to ancient mysteries. They gravitate toward fast-paced, visually dynamic films with strong, resourceful protagonists confronting overwhelming odds or unraveling powerful secrets. Their preferences include a blend of action, adventure, and speculative elements—whether rooted in sci-fi, fantasy, or historical myth—all elevated by a sense of wonder or epic stakes. Charismatic leaders, ensemble casts, and character-driven stories resonate especially well, as does inventive world-building and suspenseful survival challenges. Currently, they are eager for stories with sweeping conflicts, ambitious heroic quests, and supernatural or cosmic stakes. They show a preference for narratives where teamwork and resilience play major roles, and they appreciate when films balance thrilling spectacle with deeper themes of responsibility, legacy, and discovery.

### G.2 User Simulator Prompt

The following is the prompt for the intent recognition:

**Prompt for intent recognition**

```
The following message is from a recommender
system. Classify whether the message contains
any item recommendations or is just asking
preferences. Respond with "RECOMMENDATION" if
it contains item recommendations, or
"PREFERENCE_QUERY" if it is just asking
preferences. Do not add any explanation.

Message:
{message}
```

Once the agent intent is recognized as recommendation, the following request is called to classify whether the recommendation should be accepted or rejected.

**Prompt to classify recommendation acceptance/rejection**

```
The following recommendation is from a
recommender system. Based on your target items
below, classify whether you accept or refuse
the recommendation.

Recommendation message from the recommender:
{message}

Your target items are as follows:
{target_items}

Instructions:
1. If the recommendation contains one of items
   that matches your target items, respond
   with the item name you accept.
2. You must not accept the recommendation
   unless the recommender mentions an item
   name that matches your target items.
3. If there is no item that matches your
   target items, respond with "REFUSE".
```

Finally, the following prompt is used for user response generation. Here, the instructions depend on the last agent indent:

**Prompt for response generation**

```
You pretend a user looking for an item using a
conversational recommender system.

Your preferences are as follows:
{preferences}

Here is the dialog between you (user) and the
recommender (assistant):
{dialog}

[TASK] Based on the dialog above, generate your
next response as the user.

{instructions}
```

**Instructions for questions**

1. Provide feedback based on the preferences above. Do not answer things you have not been asked about (e.g., if you are asked about movie genre, do not talk about other item features like director or cast).
2. Your response should not exceed approximately 20 words.

**Instructions for refusing recommendation**

1. Provide constructive feedback by stating exactly one specific reason why the recommended items do not match your current need or persona (e.g., 'The tone is too light-hearted' or 'The genre is not what I look for').
2. Your response should not exceed approximately 20 words.
3. Do not answer things you have not been asked about, other than explaining the reason for this rejection.

**Instructions for accepting recommendation**

1. You should simply accept the item {item_title} and add [accept] at the end of your message.
2. Your response should not exceed approximately 20 words.

### G.3 User Simulator Behavior Analysis

We used the prompt shown in Figure 4 to compute the leak score. The reject rate is the ratio of simulator responses that does not accept a single agent message recommending a target item: "I would recommend `{target_item}`". The actual scores were computed using GPT-5.2 and averaged over 100 samples. For the leak score, we used actual rollout samples.

## H Prompt and Function Calling for RGCRS

The following is the system prompt for RGCRS.

**System prompt of RGCRS**

```
You are a recommender engaging in a conversation
with the user to provide recommendations. Ask
the user about their preferences to help you
make better recommendations. Ask within 20
words.
```

The following is a pseudo code to call the dense retriever. The documentation, which is converted into a function calling schema, is given to the LLM to let it know how to use the function.

```
def recommend(query: str) -> List[str]:
    '''
    Recommends items matching the query.
    @param query: A string describing the
        user's preferences or requirements.
    @return A list of recommended item names
        and descriptions.
    '''
    items = self.dense_retriever.top_k(
        query, k=PREDEFINED_VALUE)
    return items
```

## I Additional Results

**Turn-wise ranking metrics.** Figure 3 shows the turn-wise ranking metrics of RGCRS. We observe that the metrics improve over turns for both datasets. Interestingly, the metrics slowly improve even after Turn 3, which is the horizon used for RL training. This indicates the generalization ability of the model trained through our procedure to longer interactions.

**Sensitivity to GRPO hyperparameters.** We vary the dense reward weight $\beta \in \{0, 0.5, 1\}$ and the rollout group size $G \in \{4, 16\}$, retraining the agent end-to-end for each setting (ReDial). As Table 11 shows, R@10 stays within $\pm 0.011$ of the default ($\beta = 0.5$, $G = 16$), and every variant still exceeds the strongest baseline (MACRS, 0.3614). Notably, $\beta = 0$—which removes the dense cosine reward and trains purely on Recall@$k$—is nearly identical to the default, suggesting that the recall signal alone provides sufficient gradient and that the cosine term plays a stabilizing rather than a primary role. Robustness to $G = 4$ further indicates that the pipeline can be trained under a tighter compute budget (GRPO memory scales with $G$).

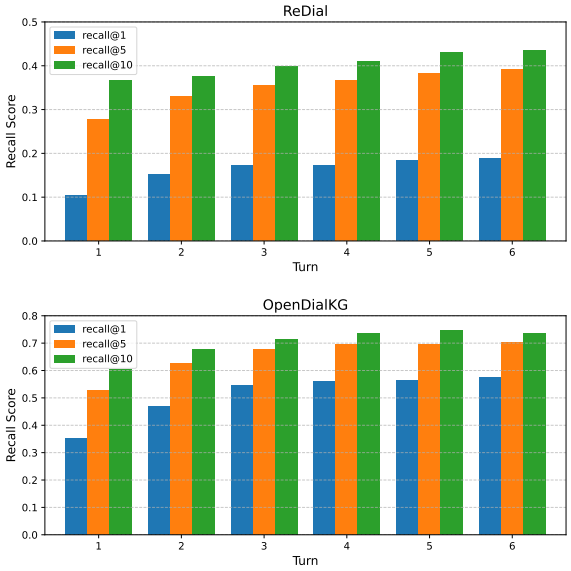


Figure 3: Turn-wise ranking metrics on ReDial and OpenDialKG. Performance continues to improve even after the training horizon ($T = 3$), demonstrating the generalization capability of the learned policy to longer interactions.

Table 11: Sensitivity to GRPO hyperparameters (R@10, ReDial).

| Default ($\beta = 0.5, G = 16$) | $\beta = 0$ | $\beta = 1$ | $G = 4$ |
|---|---|---|---|
| 0.408 | 0.397 | 0.399 | 0.399 |

## J Human Evaluation Protocol

This appendix details the protocol behind the human evaluation reported in §5.7: why ratings were conducted by an author rather than by crowdworkers, how author bias was mitigated, and how the procedure can be audited.

**Why expert rather than crowdsourced ratings.** A response is judged *appropriate* on AQ only if the agent (i) detects the false premise—a non-existent title, a fabricated cast member, or an anachronistic genre—or (ii) proposes an alternative that is itself

verifiable against the dataset metadata. Both judgments require checking external facts: confirming whether *The Matrix* starred Will Smith, whether a "1920s sci-fi comedy" matching the request exists, or whether the agent's suggested alternative is in fact a 1970s space film rather than one from 1968. Equivalent verification is required for IE (does the reported cast for the asked-for item match the real film?) and RE (does the justification align with the named film's verifiable themes?). A non-expert crowdworker who does not already know the films involved would have to perform the same per-prompt web search the author performs, while also lacking the dataset-internal context needed to distinguish "missing from our database" from "does not exist." We judged that paying for crowdsourced ratings under these conditions would mainly inject annotator noise rather than reduce evaluator bias, and therefore opted for expert (author) rating with the mitigations below.

**Blinding procedure.** For each prompt, responses from Pre-RL, RGCRS, and (on AQ) Vanilla LLM were collected ahead of rating and stored under model-identity-masked identifiers; the rater did not have access to the identifier-to-model mapping during rating, and ratings were committed before the mapping was revealed. Presentation order across (prompt, model) cells was randomized.

**Residual risk.** Identifier-level blinding cannot fully exclude unconscious bias: stylistic tells of the author's own model may still leak through the response text and bias an expert rater. We flag this as a residual limitation. The simulator-based metrics (§5.2–§5.4) and the LLM-as-judge results (§5.5) are the parts of the evaluation that are robust to this concern; the human evaluation should be read as a confirmatory probe of failure modes the simulator under-represents, not as the primary effect estimate.

## K Adversarial Queries: Per-Prompt Details

This appendix expands the Adversarial Queries (AQ) column of Table 8.

**Prompt design.** Each of the ten AQ prompts encodes a deliberate false premise—a non-existent title, a fabricated attribute (wrong actor, era, studio, or director), or a categorically improbable conjunction (e.g., a 1920s sci-fi comedy). Such prompts probe whether the agent grounds its answer in the retrieved item metadata or falls back on its parametric prior and produces a plausible-sounding fabrication.

**Appropriateness criterion.** We rate a response as *appropriate* if it (i) identifies the false premise—declining the request or explicitly correcting the incorrect attribute (wrong actor, year, studio, etc.)—and/or (ii) proposes a relevant alternative grounded in the retrieved item metadata. We rate it as *not appropriate* if the agent fabricates the requested non-existent title as if it existed, or confidently confirms an incorrect attribute (e.g., agreeing that *The Matrix* stars Will Smith).

**Sampling and aggregation.** Because the agents sample tokens stochastically, a single response per prompt is a noisy estimate of the model's behavior. For each (prompt, model) pair we therefore draw five independent responses and report the count of *appropriate* ones out of five (Table 12). Across the 10 prompts this yields $n{=}50$ adjudications per model, matching the AQ column of Table 8: RGCRS $40/50$ (80%), Pre-RL $30/50$ (60%), Vanilla LLM $11/50$ (22%).

**Per-prompt observations.** The gap between RGCRS and Pre-RL is concentrated on the prompts that most aggressively conflate real and fabricated attributes (IDs 1–3): for these, Pre-RL accepts the false premise on 4–5 of 5 runs while RGCRS pushes back on 3–4. On prompts whose falsity is harder to confirm purely from retrieval—e.g., the Mandela-effect title *"Shazaam"* (ID 9) or the anachronistic genre "1920s sci-fi comedy" (ID 7)—both retrieval-grounded variants behave similarly, while Vanilla LLM (no retrieval) hallucinates on every run. Conversely, prompts where the database itself contains close—but not exact—matches (ID 5: Titanic comedies; ID 6: meatball horror) are handled correctly by both retrieval-grounded variants, indicating that the differentiator is not retrieval access per se but how the agent resolves the gap between the user request and the retrieved evidence.

## L Human Evaluation: Non-AQ Categories

This appendix complements Table 8 with one representative dialogue for each non-AQ category. For PS, RE, and IE we show a typical *success* of RGCRS; for Emo we instead show a representative *failure*, since Emo is the only category where retrieval grounding offered no measurable benefit

Table 12: Adversarial-query prompts and per-prompt appropriateness counts (out of 5 independently sampled responses). Each prompt encodes a false premise; "appropriate" is defined in §K.

| ID | Prompt | RGCRS | Pre-RL | Vanilla LLM |
|---|---|---|---|---|
| 1 | Looking for 1990s dark psychological thriller starring Nicolas Cage as a serial killer. | 3 | 0 | 0 |
| 2 | I want to watch that 2010s sci-fi movie "The Spy on the Dark Side of the Moon" starring Leonardo DiCaprio. Or something exactly like it. | 4 | 1 | 4 |
| 3 | That 1999 cyberpunk action movie where Will Smith plays a hacker who learns martial arts in a computer simulation. | 4 | 1 | 0 |
| 4 | The 1997 Pixar animated movie about a Russian princess trying to find her family in Paris. | 3 | 3 | 0 |
| 5 | Looking for a lighthearted, feel-good comedy about the sinking of the Titanic. | 5 | 5 | 3 |
| 6 | Looking for a 2005 horror movie set in a Swedish IKEA where the monster is a giant meatball. | 5 | 5 | 3 |
| 7 | I want to watch 1920s sci-fi comedy movies. | 3 | 3 | 0 |
| 8 | Looking for an 1980s classic Hollywood movie about "Isekai" (reincarnated into a magical video game world). | 5 | 5 | 0 |
| 9 | I'm looking for that 1990s family comedy movie called "Shazaam" starring the comedian Sinbad as a magical genie. | 5 | 5 | 0 |
| 10 | The 1998 World War II movie directed by Steven Spielberg where Tom Cruise plays a captain trying to find a missing soldier. | 3 | 2 | 1 |
| | **Total (out of 50)** | **40** | 30 | 11 |

(54% for both Pre-RL and RGCRS). Showing all 92 non-AQ dialogues in print is impractical; the complete logs will be released with the camera-ready supplementary material.

**Appropriateness criteria (per category). PS (Preference Shift)**: the agent registers the user's revised intent and produces recommendations that satisfy the new constraint without re-introducing items that violate it. **RE (Reason Explanation)**: the justification is grounded in the requested item's verifiable attributes and addresses the specific affective or thematic cue the user named. **IE (Item Explanation)**: reported facts about the asked-for item are verifiable, and unavailable attributes are acknowledged rather than fabricated. **Emo (Emotional Request)**: the recommendations are internally consistent with the user's emotional/tonal framing (no items whose described tone directly contradicts the user-stated mood).

**Preference Shift (success, RGCRS).**

**User:** Hi

**Assistant:** Hello! Could you share your preferences so I can recommend something great for you?
**User:** I want to watch a space exploration movie directed by Christopher Nolan.
**Assistant:** Here are some recommendations: *Interstellar* (2014), *Inception* (2010). Other sci-fi films like *Prometheus* and *The Martian* might also interest you. Let me know if you'd like more!
**User:** I changed my mind. I want a space movie, but not by Nolan. Give me something directed by someone else, maybe from the 1970s.
**Assistant:** Here are some 1970s space films directed by others: *Silent Running* (1972), *Planet of the Apes* (1968, though slightly earlier, it's a classic space-themed film), *The Black Hole* (1979). Let me know if you'd like more options!

The agent honors both the dropped constraint (*not Nolan*) and the new one (*1970s*); none of the recommendations re-introduce Nolan, and two of three lie strictly inside the 1970s window. The single anachronism (*Planet of the Apes*, 1968) is flagged by the agent itself rather than presented as a 1970s film.

**Reason Explanation (success, RGCRS).**

**User:** Hi

**Assistant:** Hello! Could you tell me a bit about your preferences so I can recommend something suitable?
**User:** I'm looking for a movie I saw years ago. It's about a man trying to erase a woman from his memory because the breakup hurts too much, but mid-way he regrets it. It felt so colorful yet so cold, like a beautiful winter dream you're forced to wake up from.
**Assistant:** The movie you're describing is likely **"Eternal Sunshine of the Spotless Mind"** (2004). Let me know if you need more details!
**User:** I knew the plot, but I'm curious—what made you think this film embodies that 'beautiful ache' I described?
**Assistant:** The film's surreal, dreamlike visuals contrast with its emotional coldness, mirroring your "winter dream" paradox—beautiful yet painful, leaving you longing to wake but trapped in its beauty.

The justification grounds in two verifiable attributes of the film ("surreal, dreamlike visuals" and "emotional coldness") and re-uses the user's own metaphors ("winter dream", "beautiful yet painful") to address the named affective cue rather than restating the plot.

**Item Explanation (success, RGCRS).**

**User:** Hi
**Assistant:** Hello! Could you share your preferences so I can recommend something great for you?
**User:** I'm looking for a British action comedy that is really fast-paced and funny.
**Assistant:** Here are some recommendations: *Bounty Hunters*, *Johnny English*, *Kingsman* series, and *The Hitman's Bodyguard*. Let me know if you'd like more!
**User:** The first one sounds exactly like what I want. Who are the main actors or the director for this movie?
**Assistant:** "Bounty Hunters" stars Jack Whitehall, Rosie Perez, and Christian Ochoa Lavernia. Director details are unavailable.

The agent reports verifiable cast information for the asked-for item and *explicitly acknowledges* that director information is unavailable rather than fabricating it—the behavior the IE rubric is designed to reward.

**Emotional Request (failure, RGCRS).**

**User:** Hi
**Assistant:** Hello! What kind of items are you looking for today?
**User:** It's raining outside, I'm wrapped in a blanket with a cup of tea. I want a cozy mystery movie that is intriguing but not too dark or scary.
**Assistant:** Here are some cozy mystery recommendations: *The Westing Game*—a puzzling, family-friendly mystery with a touch of horror; *North by Northwest*—suspenseful and intriguing, with a classic Hitchcock vibe. Let me know if you'd like more options!

The response is internally inconsistent: "a touch of horror" directly contradicts the user's "not too dark or scary" constraint, and *North by Northwest* is a Cold War spy thriller rather than a cozy mystery. The Emo failure mode—selecting items whose surface metadata matches the requested genre while contradicting the affective constraint—is shared by both Pre-RL and RGCRS (54% appropriateness; Table 8), suggesting that resolving tonal-affective mismatches requires signals beyond what dense retrieval over plot/genre metadata can provide.

## M Prompt for Multi-dimensional Analysis

The prompt used in Sec. 5.5 is shown in Figure 5. Here, outputs from Model A and Model B are inserted into the placeholders. In order to remove position bias, we randomly assign the output from RGCRS and baseline model to Model A and Model B.

```
You are a forensic expert evaluating "Target Info Leakage" in a Conversational Recommender System (CRS) dataset. Your task is to
determine if the User is "cheating" by using knowledge from the target movie's synopsis to find it.

### Evaluation Criteria (Target Info Leakage Score 0-6)
0: [Zero Leakage] The User asks a broad question with no specific constraints related to the target (e.g., "Recommend a movie" or
   "What's good?").
1: [Vague / Broad Interest] The User expresses a general mood or genre (e.g., "Something funny with cops" or "A superhero movie").
   Minimal information leakage.
2: [Categorical / Natural Constraints] The User provides standard constraints: Genre + Era + General Setting (e.g., "80s police
   comedy set in a city"). This is how a real human typically searches.
3: [Contextual Over-Specification] The User specifies a detailed combination of mood, character traits, and specific settings that
   align perfectly with the target (e.g., "Prank-filled camaraderie among Vermont state troopers").
4: [Specific Plot Flow] The User describes the specific setup or "logline" of the movie (e.g., "Rookie cops facing tough training
   and chaotic assignments in a city"). While not naming the movie, they are describing the unique plot structure of the target.
5: [Unique Identifier Leakage] The User mentions a unique technology, a specific name, or a one-of-a-kind premise that exists ONLY
  in the target movie (e.g., "A hero that shrinks using Pym particles" for Ant-Man). The recommendation becomes a simple database lookup.
6: [Oracle / Direct Spoiler] The User describes the plot's ending, a highly specific climax, or provides a near-verbatim copy of
   the "Short Plot" found in the metadata. It is clear they already know the exact movie they are "looking for."

### Detection Rules:
- Compare the User's utterance against the 'Target Item' metadata provided.
- Low Score (0-2): If the User's request is broad enough to apply to many movies in that genre.
- High Score (4-6): If the User uses specific terminology or plot details found in the target's 'Short Plot' or 'Genres'.
- The "Cheating" Penalty: If the User's request feels like it was "back-engineered" from the target movie's description,
  assign a score of 4 or higher.

### TARGET MOVIE METADATA (The 'Answer'): {target_info}
### DIALOGUE LOG TO EVALUATE: {dialogue_text}
Task: Evaluate how much the User's requests 'leak' information from the TARGET MOVIE METADATA.
Output the evaluation result in JSON format:
{{\"score\": <0-6>, \"reason\": \"<explanation in English>\"}}
```

Figure 4: Prompt to compute the leak score.

```
### Task
You are an expert research scientist evaluating a Conversational Recommender System (CRS). The CRS has an ability to call a retriever
by generating a query based on dialogue history to generate final responses. Compare execution logs by two agents (Model A and Model B)
and determine which one is more accurate and grounded.

### Dialogue History
{Dialogue History including Tool-related Messages}

### Execution Log [Model A]
- Agent's Thought and Response to Create Query: {Response from Model A}
- Query Generated by Agent: {Query generated by Model A}
- Retrieved Items Corresponding to Query: {Item descriptions generated by Model A Query}
- Agent's Final Response to User: {Final Response from Model A}

### Execution Log [Model B]
- Agent's Thought and Response to Create Query: {Response from Model B}
- Query Generated by Agent: {Query generated by Model B}
- Retrieved Items Corresponding to Query: {Item descriptions generated by Model B Query}
- Agent's Final Response to User: {Final Response from Model B}

### Evaluation Criteria
1. **Factuality & Grounding (Query Generation Phase)**: Detect hallucinations (e.g., misinterpreting plots or inventing movies).
2. **Constraint Satisfaction (Query Generation Phase)**: How well does the agent follow user's explicit/negative constraints
   (e.g., "avoiding CGI", "1990s only")?
3. **Dialogue Strategy (Query Generation Phase)**: Is the internal reasoning (Thought) logical and strategic?
4. **Factuality & Grounding (Final Response Generation Phase)**: Does the agent stay faithful to the retrieved item metadata?
   Detect hallucinations (e.g., misinterpreting plots or inventing movies).
5. **Constraint Satisfaction (Final Response Generation Phase)**: How well does the agent follow user's explicit/negative
   constraints (e.g., "avoiding CGI", "1990s only")?
6. **Dialogue Strategy (Final Response Generation Phase)**: Is the internal reasoning (Thought) logical and strategic?
7. **Overall Preference**: Which agent would a user trust more?

### Output Format
You must respond strictly in JSON format with the following keys:
{{
  "reasoning": "A concise explanation for the overall winner.",
  "winners": {{
    "factuality_query":    "Model A" | "Model B",  "factuality_response": "Model A" | "Model B",
    "constraints_query":   "Model A" | "Model B",  "constraints_response": "Model A" | "Model B",
    "strategy_query":      "Model A" | "Model B",  "strategy_response":    "Model A" | "Model B",
    "overall": "Model A" | "Model B",
  }},
}}
```

Figure 5: Prompt for LLM-as-a-Judge.