HARPO: Hierarchical Agentic Reasoning for User-Aligned Conversational Recommendation

Subham Raj[1] **Aman Vaibhav Jha**[1] **Mayank Anand**[2] **Sriparna Saha**[1]
[1]Department of Computer Science and Engineering, Indian Institute of Technology Patna, India
[2]Indian Institute of Information Technology Allahabad, India
{subham_2221cs25, 2201ai54_aman, sriparna}@iitp.ac.in
anandmayank698@gmail.com

Abstract

Conversational recommender systems (CRSs) operate under incremental preference revelation, requiring recommendation decisions under uncertainty. While recent LLM-based approaches achieve strong performance on proxy metrics such as Recall@K and BLEU, they often fail to deliver high-quality, user-aligned recommendations in practice, as they optimize intermediate objectives like retrieval accuracy or fluent generation rather than recommendation quality itself. We propose HARPO (Hierarchical Agentic Reasoning with Preference Optimization), an agentic framework that reframes conversational recommendation as a structured decision-making process optimized for multi-dimensional recommendation quality. HARPO integrates (i) hierarchical preference learning that decomposes recommendation quality into interpretable dimensions (relevance, diversity, satisfaction, and engagement) with context-dependent weighting; (ii) deliberative tree-search reasoning guided by a learned value network evaluating candidate paths on predicted quality; and (iii) domain-agnostic reasoning abstractions through Virtual Tool Operations and multi-agent refinement. We evaluate HARPO on ReDial, INSPIRED, and MUSE, demonstrating consistent improvements over strong baselines on recommendation-centric metrics while maintaining competitive response quality. Our code is available at https://harpo-bench.github.io.

1 Introduction

Conversational recommender systems (CRSs) aim to assist users in discovering items that match their preferences through natural language interaction. Unlike traditional recommendation systems that rely on static user profiles or historical behavior, CRSs operate in a sequential setting where user preferences are revealed incrementally through dialogue. As a result, these systems must interpret nuanced and often underspecified user intent and make recommendation decisions under uncertainty.

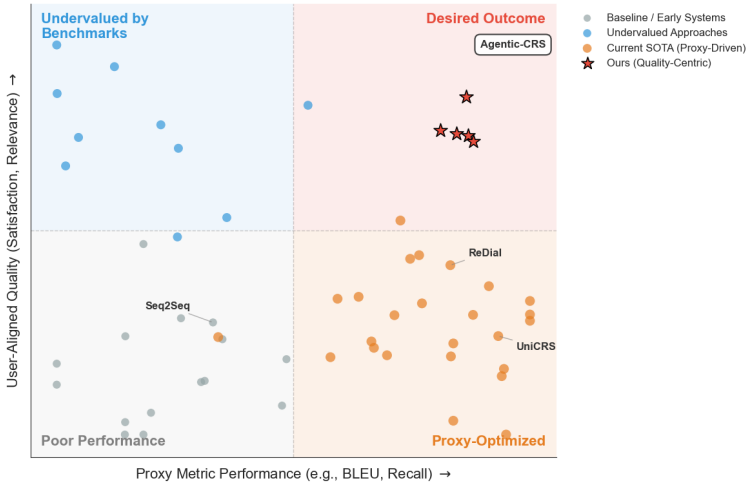

Figure 1: Conceptual illustration of the evaluation landscape for conversational recommender systems. The horizontal axis denotes performance on proxy metrics (e.g., BLEU, Recall), while the vertical axis represents user-aligned recommendation quality (e.g., satisfaction and relevance). Existing systems often achieve high proxy scores with limited alignment to user preferences. Positions are illustrative rather than measured values.

Recent advances in large language models have significantly improved conversational recommendation performance (Wang et al., 2022b; Dao et al., 2024). Modern CRS approaches, including retrieval-augmented and generation-based systems (Li et al., 2018; Hayati et al., 2020; Wang et al., 2022b), achieve strong results on widely used benchmarks, with high scores on metrics such as Recall@K, BLEU, and tool invocation accuracy. As illustrated in Figure 1, these results create an impression that conversational recommendation is largely solved, despite limited alignment with user-centric recommendation quality.

However, strong proxy-metric performance does not necessarily translate into high-quality recom-

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 35580–35599

mendations from a user's perspective. A system may retrieve appropriate items and generate fluent responses, yet still fail to capture nuanced user intent. For instance, a request for "something casual for a summer wedding" may be interpreted as everyday casual wear rather than context-appropriate wedding attire. While such responses score well on automatic metrics, they often lead to low user satisfaction in practice.

This gap reveals a fundamental misalignment between how conversational recommender systems are trained and evaluated and the actual objective of conversational recommendation. Existing approaches (Li et al., 2018; Hayati et al., 2020; Wang et al., 2022b) primarily optimize proxy objectives such as lexical overlap or retrieval of ground-truth items, which only weakly correlate with user-aligned recommendation quality (Jannach et al., 2021; Gao et al., 2021). Table 1 illustrates this reliance on proxy objectives, with explicit modeling of recommendation quality largely absent.

To address this limitation, conversational recommendation is framed as a structured decision-making problem (Wei et al., 2022; Yao et al., 2023) rather than a byproduct of response generation or tool execution. From this perspective, a system should explicitly reason over multiple candidate recommendation strategies, evaluate their expected quality, and select recommendations based on user-aligned criteria rather than proxy signals alone.

HARPO is proposed as an agentic framework that operationalizes this perspective by combining deliberative reasoning with explicit preference optimization across multiple dimensions of recommendation quality, including relevance, diversity, predicted user satisfaction, and engagement. Through structured reasoning and learned quality evaluation, HARPO enables systems to explore, compare, and refine recommendation decisions before generating final responses. It is evaluated on multiple conversational recommendation benchmarks, including ReDial (Li et al., 2018), INSPIRED (Hayati et al., 2020), and MUSE (Wang et al., 2025).

In summary, HARPO makes the following contributions:

- *A fundamental misalignment in conversational recommender systems is identified, showing that proxy metrics fail to reflect user-aligned recommendation quality.*
- *HARPO is introduced as an agentic framework that formulates conversational recommendation as a structured decision-making problem.*
- *Multi-dimensional modeling of recommendation quality is incorporated to directly optimize user-aligned objectives such as relevance, diversity, and satisfaction.*
- *A quality-centric evaluation perspective is proposed and empirically validated across multiple benchmarks.*

The remainder of the paper is organized as follows. Section 2 reviews related work. Section 3 details the HARPO framework. Section 4 reports experimental results. Section 5 concludes the paper.

2 Related Work

We review prior work in conversational recommendation, reasoning in language models, preference optimization, and tool-augmented generation, emphasizing the reliance on proxy objectives rather than explicit recommendation quality.

2.1 Conversational Recommender Systems

Conversational recommender systems (CRSs) have evolved from early constraint-based dialogue systems to neural architectures supporting open-ended interaction. Early approaches framed recommendation as a structured decision process, relying on attribute-based queries and explicit constraint elicitation to narrow candidate sets (Christakopoulou et al., 2016; Sun and Zhang, 2018). ReDial (Li et al., 2018) established the modern generation-based CRS paradigm, enabling natural language interaction while reinforcing reliance on automatic proxy metrics from recommendation and dialogue modeling.

Subsequent work incorporated external knowledge to improve recommendation accuracy and mitigate cold-start issues (Chen et al., 2019; Zhou et al., 2020). Pre-trained language models further unified conversational recommendation architectures, enabling joint dialogue understanding, reasoning, and recommendation (Wang et al., 2022a,b), with more recent methods integrating retrieval augmentation and multimodal grounding (Dao et al., 2024; Wei et al., 2025). Beyond conversational settings, multimodal recommendation has been explored through multitask learning frameworks that jointly optimize genre classification and rating prediction (Raj et al., 2023),

Method	Recall@K	BLEU	Tool Acc.	Explicit Quality Modeling	User-Aligned Objective
ReDial-style CRS	✓	✓	–	×	×
UniCRS	✓	✓	✓	×	×
DCRS	✓	✓	✓	×	×
HARPO (ours)	✓	✓	✓	✓	✓

Table 1: Comparison of evaluation objectives across representative conversational recommender systems. Existing CRS methods are primarily evaluated using proxy metrics, whereas HARPO explicitly optimizes user-aligned recommendation quality.

learn unified user-movie representations from visual and textual modalities (Raj et al., 2025), and incorporate genre-aware scoring for domain-specific settings (Mondal et al., 2023); however, these approaches operate on static user profiles rather than the incremental preference revelation central to CRS. Despite broader advances in fluency, retrieval accuracy, and tool usage, existing CRSs are still primarily trained and evaluated using proxy metrics such as Recall@K and BLEU, which reward technical correctness but do not explicitly capture alignment with nuanced user intent or subjective satisfaction, motivating the need for quality-centric evaluation and optimization frameworks.

2.2 Reasoning and Preference Optimization

Explicit reasoning mechanisms such as chain-of-thought prompting and tree-based search improve performance on multi-step decision problems (Wei et al., 2022; Yao et al., 2023). In parallel, preference optimization methods align model outputs with human judgments through learned reward signals (Ouyang et al., 2022; Rafailov et al., 2023). However, existing approaches typically optimize correctness-based or single-scalar objectives, leaving open how reasoning can be guided toward multi-dimensional, user-aligned outcomes. HARPO bridges this gap by combining deliberative reasoning with hierarchical preference learning, enabling value-guided exploration over candidate recommendation strategies based explicitly on predicted recommendation quality.

2.3 Tool-Augmented Language Models

Augmenting language models with external tools has proven effective for tasks requiring access to up-to-date information, specialized computation, or structured data sources (Schick et al., 2023; Qin et al., 2023). In conversational recommendation, tools facilitate interaction with product catalogs, user histories, and knowledge graphs that cannot be fully captured within model parameters.

However, most existing tool-augmented approaches rely on domain-specific tools and training, limiting transferability and complicating evaluation. This tight coupling between tools and tasks further reinforces proxy-metric-driven benchmarks, as evaluation becomes dependent on specific tool implementations rather than underlying recommendation behavior.

To mitigate this issue, Virtual Tool Operations (VTOs) are introduced to decouple high-level reasoning from domain-specific tools. By defining domain-agnostic operations that are mapped to concrete tools at runtime, VTOs support transferable reasoning and more consistent evaluation across domains. This abstraction mirrors interface-based design in software engineering, enabling models to reason about recommendation decisions independently of underlying tool details.

3 The HARPO Framework

HARPO integrates four components within a unified architecture built upon a pre-trained language model backbone. We first formalize the optimization objective, then describe each component in detail, explaining the intuition and mathematical formulation of key mechanisms. An overview of the framework and the interaction between its components is shown in Figure 2. Additionally, the complete Virtual Tool Operations (VTOs) taxonomy, operation definitions, illustrative examples, and annotation procedure are provided in Appendix A.

3.1 Problem Formulation

Let $\mathcal{C} = \{(u_1, r_1), \ldots, (u_{t-1}, r_{t-1}), u_t\}$ denote a conversation context consisting of user-system turn pairs and the current user utterance u_t. The system must generate a response r_t containing recommendations. Let $\mathcal{V}$ denote the VTOs space and $\mathcal{D}$ the set of recommendation domains.

We seek a policy $\pi_\theta(r_t, \mathbf{v}_t \mid \mathcal{C}, d)$ that jointly generates responses and VTOs sequences, maxi-

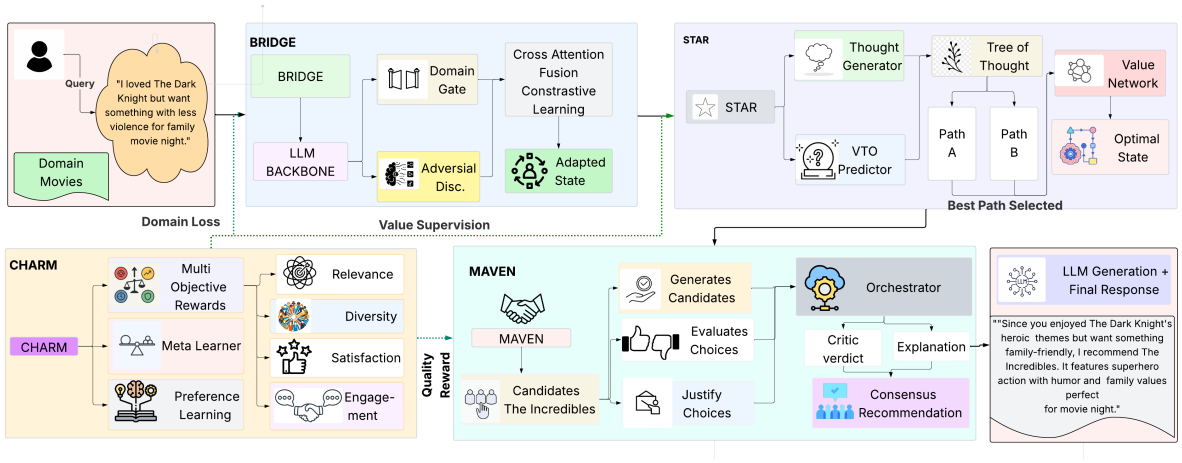

Figure 2: Overall architecture of the HARPO framework. The model integrates four components: STAR for structured agentic reasoning, CHARM for hierarchical preference optimization, BRIDGE for cross-domain transfer, and MAVEN for multi-agent refinement, all built on a shared language model backbone.

mizing expected recommendation quality:

$$\theta^* = \arg\max_\theta \mathbb{E}_{\mathcal{C},d}\left[\mathcal{Q}(r_t, \mathcal{C}) \mid \pi_\theta\right] \tag{1}$$

where $\mathbf{v}_t \in \mathcal{V}^*$ is the predicted VTOs sequence, $d \in \mathcal{D}$ is the domain, and $\mathcal{Q}(\cdot)$ measures recommendation quality.

The key departure from prior work lies in explicitly optimizing for $\mathcal{Q}$ rather than proxy objectives. Previous approaches optimize for item retrieval accuracy (maximizing probability of ground-truth items) or response quality (maximizing likelihood under reference responses), treating recommendation quality as an emergent property. Our formulation directly targets $\mathcal{Q}$, which we decompose into interpretable dimensions through CHARM.

3.2 STAR: Structured Tree-of-Thought Agentic Reasoning

STAR is a structured tree-of-thought(Yao et al., 2023) reasoning module that enables explicit exploration of multiple candidate recommendation strategies, guided by a learned value function estimating recommendation quality. Reasoning proceeds by expanding and evaluating candidate paths using beam search over structured reasoning states.

Reasoning State. Each node in the reasoning tree is represented as

$$s = (\mathbf{h}, \tau, \mathbf{v}, d), \tag{2}$$

where $\mathbf{h}$ encodes accumulated dialogue and reasoning context, τ is the current natural language thought, $\mathbf{v}$ denotes predicted VTOs, and d is the search depth. The hidden state $\mathbf{h}$ is computed from the full conversation context and the reasoning path, ensuring that value estimates reflect the complete trajectory.

Value Network. STAR employs a value network that predicts recommendation quality rather than task completion. Quality is decomposed into four dimensions—relevance, diversity, satisfaction, and engagement—each estimated by a dedicated head:

$$V_k(\mathbf{h}) = \sigma\left(\mathbf{W}_k^{(2)} \cdot \text{GELU}\left(\mathbf{W}_k^{(1)} \cdot \mathbf{h}\right)\right), \tag{3}$$

where $\mathbf{W}_k^{(1)} \in \mathbb{R}^{d_h \times d_h}$ and $\mathbf{W}_k^{(2)} \in \mathbb{R}^{1 \times d_h}$ are learnable projection matrices, with the overall value computed as:

$$V(\mathbf{h}) = \sum_k \alpha_k V_k(\mathbf{h}), \tag{4}$$

where $\alpha_k = \text{softmax}(\mathbf{w})_k$ are learnable weights over quality dimensions.

Thought Generation and Search. At each node, the language model generates b candidate next steps,

$$\{(\mathbf{h}_i, \mathbf{v}_i, q_i)\}_{i=1}^b = \text{ThoughtGen}(\mathbf{h}), \tag{5}$$

where q_i is the predicted quality score for candidate i, which are filtered and explored using beam search with width w and depth D:

$$s^* = \arg\max_{s \in \text{Beam}(s_0, w, D)} V(\mathbf{h}_s), \tag{6}$$

where $s_0 = (\mathbf{h}_0, \tau_0, \mathbf{v}_0, 0)$ is the initial reasoning state corresponding to the conversation context

prior to any reasoning step. The final response is generated from the highest-valued reasoning path.

By training the value function with quality signals from CHARM, STAR prioritizes reasoning paths that yield contextually appropriate, diverse, and satisfying recommendations, rather than optimizing task completion alone.

3.3 CHARM: Contrastive Hierarchical Alignment with Reward Marginalization

CHARM optimizes recommendation quality by decomposing it into four interpretable dimensions—relevance, diversity, satisfaction, and engagement—and learning context-dependent weighting over these dimensions.

Hierarchical Reward Decomposition. Each quality dimension is modeled by a dedicated reward head:

$$R_k(\mathbf{h}) = \tanh\Big(\mathbf{W}_k^{(2)} \cdot \text{GELU}\Big(\mathbf{W}_k^{(1)} \cdot \mathbf{h}\Big)\Big)\,, \tag{7}$$

where $\mathbf{W}_k^{(1)} \in \mathbb{R}^{d_h \times d_h}$ and $\mathbf{W}_k^{(2)} \in \mathbb{R}^{1 \times d_h}$ are learnable projection matrices, and outputs are bounded to $[-1, 1]$ for stable optimization. The total reward is computed as:

$$R(\mathbf{h}) = \sum_k w_k \cdot R_k(\mathbf{h}). \tag{8}$$

Adaptive Weighting. Context-dependent weights are learned via meta-learning:

$$\mathbf{w} = \text{softmax}(\mathbf{W}_{\text{meta}} \cdot [\text{Enc}(\mathbf{h}); \mathbf{e}_d] + \mathbf{b})\,, \tag{9}$$

where $\mathbf{W}_{\text{meta}}$ is a learnable matrix, $\mathbf{e}_d$ is a domain embedding, $\mathbf{b}$ is a bias vector, and $\text{Enc}(\cdot)$ is a context encoder, allowing different quality dimensions to be emphasized based on conversational context and domain.

Preference Optimization. Given preference pairs (r^+, r^-), optimization is performed using a margin-based objective:

$$\mathcal{L}_{\text{pref}} = -\log \sigma\big(\beta \cdot (R(\mathbf{h}^+) - R(\mathbf{h}^-) - m)\big)\,, \tag{10}$$

with an adaptive margin $m = m_0 + \frac{1}{2}\sigma(\mathbf{W}_m \cdot [\mathbf{h}^+; \mathbf{h}^-])$, where m_0 is a base margin and $\mathbf{W}_m$ is a learnable matrix, which prevents trivial preference separation and improves generalization.

3.4 BRIDGE: Cross-Domain Transfer

BRIDGE enables cross-domain generalization by learning representations that capture domain-invariant reasoning patterns while selectively preserving domain-specific information. It combines adversarial domain adaptation with gated representation learning to balance invariance and specificity.

Adversarial Domain Adaptation. To promote domain-invariant representations, we employ gradient reversal (Ganin and Lempitsky, 2015):

$$\mathcal{L}_{\text{domain}} = \text{CE}(\text{Disc}(\text{GRL}_\alpha(\mathbf{z})), d)\,, \tag{11}$$

where $\mathbf{z}$ is the encoded representation, GRL_α is the gradient reversal layer with reversal strength α, $\text{Disc}(\cdot)$ is a domain discriminator, and reversed gradients encourage indistinguishable representations across domains.

Task Preservation. To prevent the loss of task-relevant information during adaptation, we introduce an auxiliary VTOs prediction objective:

$$\mathcal{L}_{\text{task}} = \text{BCE}(\text{VTOHead}(\mathbf{h}), \mathbf{v})\,, \tag{12}$$

where $\text{VTOHead}(\cdot)$ is a linear projection head predicting the VTO sequence $\mathbf{v}$ from $\mathbf{h}$. This regularizes the adaptation process by preserving reasoning-related features essential for recommendation.

Domain Gates. Pure invariance may suppress useful domain-specific signals. We therefore introduce learnable domain gates:

$$\mathbf{z}' = \sigma(\mathbf{g}_d) \odot \mathbf{z} + (1 - \sigma(\mathbf{g}_d)) \odot \mathbf{h}, \tag{13}$$

where $\mathbf{g}_d \in \mathbb{R}^{d_h}$ is a learnable domain-specific gate vector and $\mathbf{z}'$ is the gated representation, which allows the model to interpolate between the domain-adapted representation $\mathbf{z}$ and the original hidden state $\mathbf{h}$ based on domain context.

3.5 MAVEN: Multi-Agent Refinement

MAVEN refines recommendations through collaboration among specialized agents with complementary roles for recommendation, critique, and explanation. Agents operate on shared representations while producing role-specific outputs.

Each agent a has a dedicated encoder and output head:

$$\mathbf{o}_a = \text{Head}_a\left(\text{Enc}_a(\mathbf{h})\right), \tag{14}$$

where $\text{Enc}_a(\cdot)$ and $\text{Head}_a(\cdot)$ are agent-specific encoder and output projection, and $a \in$

$\{\text{rec}, \text{crit}, \text{exp}\}$ denotes the recommender, critic, and explainer agents respectively. Separate encoders allow agents to specialize according to their roles.

Orchestration. An orchestrator aggregates agent outputs into a final response:

$$\mathbf{o}_{\text{final}} = \text{FFN}\left([\mathbf{o}_{\text{rec}}; \mathbf{o}_{\text{crit}}; \mathbf{o}_{\text{exp}}]\right), \quad (15)$$

where $\mathbf{o}_{\text{rec}}$, $\mathbf{o}_{\text{crit}}$, and $\mathbf{o}_{\text{exp}}$ are the outputs of the recommender, critic, and explainer agents respectively, and $\text{FFN}(\cdot)$ is a feed-forward network that aggregates agent contributions based on conversational context.

Agreement Loss. To encourage coherent collaboration, we introduce a soft consensus objective:

$$\mathcal{L}_{\text{agree}} = 1 - \cos(\mathbf{o}_{\text{rec}}, \mathbf{o}_{\text{crit}}). \quad (16)$$

This promotes alignment between recommendation and critique while allowing disagreement when necessary. Detailed training procedure is discussed in Appendix B.

4 Experiments and Results

We conduct extensive experiments to answer the following research questions:

RQ1: Does HARPO improve user-aligned recommendation quality compared to state-of-the-art baselines?

RQ2: How do individual components (CHARM, STAR, BRIDGE, MAVEN) contribute to performance?

RQ3: Can HARPO transfer recommendation reasoning across domains?

RQ4: How does hierarchical reward decomposition compare to flat reward signals?

RQ5: How much is the model sensitive to its hyperparameters?

4.1 Datasets

We evaluate on three conversational recommendation benchmarks with diverse characteristics (Table 2). Details about the dataset description is shown in Appendix Section D.1.

Statistic	ReDial	INSPIRED	MUSE
Domain	Movies	Movies	Fashion
# Conversations	10,006	1,001	7,000
# Utterances	182,150	35,811	83,204
# Unique Items	51,699	8,952	13,754
Avg. Turns/Conv.	18.2	35.8	11.9
Avg. Items/Conv.	4.3	3.8	5.2
Avg. Words/Turn	7.6	7.9	46.6
Modality	Text	Text	Text+Image
Interaction Style	Transactional	Sociable	Scenario-based
Train / Val / Test	8K/1K/1K	800/100/101	5.6K/0.7K/0.7K

Table 2: Dataset statistics of Redial, INSPIRED, and MUSE.

4.2 Baselines

We compare against methods spanning four representative paradigms. **All baselines are reproduced using official code releases**, with hyperparameters tuned on validation sets; reproduced results are within 2% of reported numbers where available.

Knowledge-Enhanced Methods: **KBRD** (Chen et al., 2019): R-GCN over DBpedia entities. **KGSF** (Zhou et al., 2020): DBpedia and ConceptNet with mutual information maximization. **UniCRS** (Wang et al., 2022b): Knowledge-enhanced prompt learning for unified recommendation and generation.

Retrieval-Augmented Methods: **BARCOR** (Wang et al., 2022a): BART-based joint recommendation and generation. **DCRS** (Dao et al., 2024): Contrastive retrieval for demonstration selection.

LLM-Based Methods: **ChatGPT**: GPT-3.5-turbo with 5-shot task-specific prompting. **GPT-4**: GPT-4 with chain-of-thought prompting. **LLaMA-2-Chat** (Touvron et al., 2023): 7B and 13B instruction-tuned variants.

Reasoning-Enhanced Methods: **RecMind** (Wang et al., 2024): Self-inspiring reasoning with planning. **InteRecAgent** (Huang et al., 2025): Interactive agent with learned tool usage.

Baseline Fairness: All methods use identical training splits. LLM baselines employ carefully engineered prompts (Appendix C) with effort comparable to our method, and knowledge-enhanced methods rely on the same DBpedia and ConceptNet versions. Implementation details and evaluation protocols are provided in Appendix D.

4.3 Main Results (RQ1)

Table 3 reports recommendation performance. HARPO achieves state-of-the-art results across all metrics and datasets, with particularly strong gains on user-aligned measures.

Key Findings: **(1) Consistent improvements across paradigms:** HARPO outperforms all baseline categories, achieving 17–21% average improvement over the strongest baseline (GPT-4/GPT-4V) across datasets. Improvements are larger on user-aligned metrics (Satisfaction, Engagement) than on proxy metrics (Recall, NDCG), indicating effective mitigation of proxy-metric misalignment.

(2) Largest gains on INSPIRED: The sociable dialogue setting with implicit preference signals benefits most from STAR's deliberative reasoning (+45.7% R@10 vs. GPT-4), demonstrating the advantage of tree-of-thought reasoning when preferences are inferred rather than explicitly stated.

(3) Effective cross-modal transfer: Despite primarily text-based training, HARPO performs strongly on the multimodal MUSE dataset, suggesting that VTO abstractions enable transferable, domain-agnostic reasoning.

Generation Quality and VTO Accuracy: HARPO maintains strong generation quality alongside high VTO prediction accuracy (F1 0.74–0.79; Table 7 provided in Appendix), indicating that explicit reasoning supervision benefits both response quality and reasoning alignment.

Human Evaluation: To provide non-circular quality evidence, three expert annotators rated 200 test samples per dataset (Table 8, Appendix). HARPO improves over GPT-4 by +0.55 (Rec. Quality), +0.50 (Exp. Quality), and +0.55 (Overall) on ReDial (1–5 scale, κ>0.74), confirming that gains extend beyond model-based metrics.

Reward Model Validity: User Satisfaction and Engagement in Table 3 are computed via CHARM's reward model. To address potential circularity, we validate through: (1) Pearson correlations with independent human judgments on held-out data—relevance (r=0.71), diversity (r=0.68), satisfaction (r=0.73), engagement (r=0.64)—confirming CHARM captures meaningful quality dimensions; (2) human evaluation (Table 8) showing consistent gains across all systems, providing non-circular evidence. Critically, CHARM is trained only on Stage 2 preference pairs and frozen thereafter, preventing test-time reward hacking.

Table 4 isolates component contributions on ReDial. **Component Analysis: (1) CHARM is critical:** Removing hierarchical preference learning causes the largest performance drop (-17.4% R@10, -19.1% Satisfaction), highlighting the importance of decomposed rewards for quality-centric optimization. **(2) STAR improves ranking:** Tree search primarily benefits ranking metrics (-9.4% R@10) through deliberative exploration of reasoning paths. **(3) BRIDGE enables transfer:** Domain adaptation yields modest within-domain gains (-4.7% R@10) but is crucial for cross-domain transfer (Section 4.4). **(4) VTOs provide inductive bias:** Removing VTO abstractions causes substantial degradation (-21.5% R@10), confirming that domain-agnostic reasoning primitives capture transferable logic.

Mechanism Analysis: **Flat Reward:** Collapsing hierarchical rewards into a scalar reduces satisfaction by 14.7%, indicating the benefit of optimizing quality dimensions separately. **Fixed Weights:** Replacing meta-learned weights with uniform weights lowers satisfaction by 8.8%, showing the value of context-adaptive weighting. **Greedy Search:** Substituting beam search with greedy search reduces R@10 by 10.1%, underscoring the importance of exploring multiple reasoning paths.

Training Stage Analysis: Each training stage yields incremental improvements, with CHARM (Stage 2) providing the largest single-stage gain ($+21.3\%$ R@10 over SFT-only), validating the four-stage curriculum design.

4.4 Cross-Domain Transfer (RQ3)

Table 9 provided in Appendix evaluates zero-shot transfer between domains.

BRIDGE's adversarial domain adaptation encourages domain-invariant representations for shared reasoning patterns while preserving domain-specific information through learned gates. These gains build upon VTO abstractions that provide domain-agnostic structure; the improvements reflect synergistic effects rather than BRIDGE alone. **Gate Analysis:** Gates for product-specific features (visual attributes, genre conventions) remain low (0.2-0.3), while gates for reasoning operations (preference modeling, constraint satisfaction) remain high (0.7-0.8), showing interpretable domain specialization.

4.5 Hierarchical Reward Analysis (RQ4)

Adaptive Weight Distribution: The meta-learner adjusts reward weights based on conversational context: **Explicit preference queries** ("I love action movies"): Relevance weight increases to 0.42 (vs. 0.30 baseline). **Exploratory dialogues**

Method	Ranking Metrics (%)					Satisfaction		Avg.
	R@1	R@10	R@50	MRR@10	NDCG@10	User Sat.	Engage.	
ReDial Dataset								
KBRD	2.9±0.2	16.7±0.4	36.2±0.7	7.4±0.2	10.2±0.3	0.42±0.02	0.38±0.02	0.291
KGSF	3.8±0.2	18.1±0.5	37.4±0.7	8.4±0.3	11.6±0.4	0.45±0.02	0.41±0.02	0.318
BARCOR	3.0±0.2	16.8±0.4	36.8±0.6	7.8±0.2	10.8±0.3	0.44±0.02	0.40±0.02	0.306
UniCRS	4.8±0.3	21.2±0.5	40.8±0.8	10.1±0.3	13.8±0.4	0.51±0.02	0.47±0.02	0.364
DCRS	7.5±0.3	25.1±0.6	43.6±0.9	12.2±0.4	15.2±0.5	0.56±0.02	0.52±0.02	0.408
ChatGPT	3.3±0.4	17.0±0.7	37.8±1.1	8.0±0.4	11.0±0.5	0.49±0.03	0.45±0.03	0.324
GPT-4	4.5±0.4	19.4±0.8	40.2±1.2	9.6±0.5	13.2±0.6	0.55±0.03	0.51±0.03	0.368
LLaMA-2-7B	2.2±0.3	13.6±0.6	33.4±0.9	6.2±0.3	8.6±0.4	0.38±0.02	0.34±0.02	0.260
LLaMA-2-13B	2.8±0.3	15.4±0.6	35.6±1.0	7.2±0.4	9.9±0.5	0.43±0.02	0.39±0.02	0.293
RecMind	5.8±0.3	22.6±0.6	42.2±0.9	11.2±0.4	15.3±0.5	0.54±0.02	0.50±0.02	0.385
InteRecAgent	5.2±0.3	21.4±0.6	41.0±0.8	10.4±0.4	14.3±0.5	0.52±0.02	0.48±0.02	0.369
HARPO	**9.1±0.3**	**29.8±0.7**	**50.2±1.0**	**15.6±0.5**	**21.2±0.6**	**0.68±0.02**	**0.64±0.02**	**0.481**
Δ vs. DCRS	+21.3%	+18.7%	+15.1%	+27.9%	+39.5%	+21.4%	+23.1%	+17.9%
INSPIRED Dataset								
KGSF	2.4±0.3	13.8±0.6	31.6±1.0	6.4±0.3	8.8±0.4	0.40±0.03	0.36±0.02	0.270
UniCRS	3.8±0.3	17.6±0.7	37.2±1.2	8.6±0.4	11.8±0.5	0.48±0.03	0.44±0.03	0.331
GPT-4	4.2±0.5	18.8±0.9	39.4±1.5	9.4±0.5	12.9±0.6	0.53±0.03	0.49±0.03	0.358
RecMind	4.8±0.4	20.4±0.8	41.2±1.3	10.2±0.5	14.0±0.6	0.52±0.03	0.48±0.03	0.370
HARPO	**7.2±0.4**	**27.4±0.9**	**48.8±1.4**	**14.2±0.6**	**19.4±0.7**	**0.66±0.03**	**0.62±0.03**	**0.454**
Δ vs. RecMind	+50.0%	+34.3%	+18.4%	+39.2%	+38.6%	+26.9%	+29.2%	+22.7%
MUSE Dataset (Multimodal Fashion)								
UniCRS†	1.6±0.3	11.8±0.6	27.4±1.1	5.1±0.3	7.2±0.4	0.36±0.03	0.32±0.02	0.229
GPT-4V	4.4±0.5	23.2±0.9	42.6±1.4	10.8±0.5	14.8±0.6	0.54±0.03	0.50±0.03	0.374
Qwen2-VL-7B‡	8.4±0.4	34.2±0.8	52.8±1.3	17.2±0.4	23.1±0.5	0.61±0.03	0.57±0.03	0.468
LLaVA-Next-8B‡	5.2±0.4	25.4±0.7	44.2±1.2	12.0±0.4	16.2±0.5	0.52±0.03	0.48±0.03	0.380
HARPO	**10.2±0.4**	**38.6±0.9**	**58.4±1.3**	**19.8±0.5**	**26.4±0.6**	**0.72±0.03**	**0.68±0.03**	**0.524**
Δ vs. Qwen2-VL	+21.4%	+12.9%	+10.6%	+15.1%	+14.3%	+18.0%	+19.3%	+12.0%

Table 3: Main recommendation results on three datasets. R@K: Recall@K (%). Values: mean±std over 3 runs with different random seeds. User Sat./Engage.: normalized [0,1]. Avg.: macro-average of normalized metrics. †Text-only adaptation using item descriptions. ‡Fine-tuned following the protocol in Wang et al. (2025). Bold: best; Δ: relative improvement (%) over strongest baseline. All HARPO improvements are statistically significant at $p < 0.01$ (paired t-test with Bonferroni correction).

Variant	R@10	MRR@10	NDCG@10	Sat.	Eng.
HARPO (Full)	**29.8**	**15.6**	**21.2**	**0.68**	**0.64**
Component Ablation					
w/o CHARM	24.6	12.6	17.2	0.55	0.51
w/o STAR	27.0	14.0	19.0	0.63	0.59
w/o BRIDGE	28.4	14.9	20.2	0.66	0.62
w/o MAVEN	28.0	14.7	19.9	0.65	0.61
w/o VTOs	23.4	12.0	16.4	0.53	0.49
Mechanism Variants					
Flat Reward	26.0	13.4	18.2	0.58	0.54
Fixed Weights	27.2	14.1	19.1	0.62	0.58
Greedy Search	26.8	13.8	18.7	0.61	0.57
No Backtracking	28.2	14.6	19.8	0.65	0.61
Training Stage Ablation					
SFT Only (Stage 1)	21.6	10.6	14.6	0.50	0.46
+ CHARM (Stages 1–2)	26.2	13.4	18.2	0.61	0.57
+ STAR (Stages 1–3)	28.4	14.8	20.1	0.65	0.61

Table 4: Ablation study on ReDial. Component ablations retrain without the specified module; "w/o VTOs" sets λ_v=0 in Eq. 17. Mechanism variants modify inference only. Component contributions are non-additive due to interactions. Sat./Eng. $\in [0, 1]$; all metrics in %.

("I want to try something new"): Diversity weight increases to 0.31 (vs. 0.20 baseline). **Extended conversations** (>10 turns): Engagement weight increases to 0.28 (vs. 0.20 baseline).

Correlation with Human Judgments: Each reward dimension correlates most strongly with its corresponding human rating: Relevance reward ↔ Human relevance: $r = 0.71$ ($p < 0.001$) Diversity reward ↔ Human diversity: $r = 0.68$ ($p < 0.001$) Satisfaction reward ↔ Human satisfaction: $r = 0.73$ ($p < 0.001$) Engagement reward ↔ Follow-up rate: $r = 0.64$ ($p < 0.001$)

This validates semantic alignment between learned reward components and intended quality dimensions.

4.6 Hyperparameter Sensitivity (RQ5)

HARPO is robust to hyperparameter choices as shown in Table 10 provided in Appendix. Key findings: **(1)** Beam width $w = 3$ achieves 99% of $w = 5$ performance with 40% lower latency. **(2)** Depth $D = 3$ provides near-optimal results; $D = 5$ adds marginal gains (+1.0% R@10) with significant latency increase. **(3)** CHARM $\beta = 0.5$ balances preference learning strength; lower values underfit, higher values overfit to training preferences. **(4)** LoRA rank $r = 16$ is sufficient; $r = 32$ provides no improvement.

A detailed computational complexity analysis is provided in Appendix D.5. An error analysis covering 100 representative failure cases is presented in Appendix D.6. In addition, a qualitative analysis of reasoning quality is reported in Appendix D.4. A FAQ section is also added in Appendix E.

5 Conclusion

Conversational recommender systems are typically trained and evaluated using proxy objectives that are easy to measure but weakly reflect user-experienced recommendation quality. This work reframes conversational recommendation as a quality-centric decision-making problem, arguing that effective systems should explicitly optimize user-aligned outcomes such as relevance, satisfaction, diversity, and engagement.

HARPO is introduced as an agentic framework that combines structured reasoning with hierarchical preference optimization to treat recommendation quality as a first-class objective. Results across multiple benchmarks show consistent gains on user-aligned measures while remaining competitive on standard metrics, highlighting the limitations of proxy-driven evaluation and motivating more principled, quality-aware approaches.

6 Limitations

This work has some limitations. The four-stage training pipeline is computationally intensive, requiring multiple training phases with distinct objectives. The VTOs taxonomy, while effective in practice, is manually designed and may not capture all relevant reasoning patterns. In addition, evaluation relies primarily on automatic metrics with limited human assessment; larger-scale user studies would provide stronger evidence of user satisfaction gains. Finally, cross-domain evaluation is restricted to three datasets, and broader domain coverage would better characterize transferability.

7 Ethics Statement

Conversational recommender systems raise several ethical considerations. Optimizing for engagement could encourage filter bubbles or addictive usage patterns; we partially address this through the diversity reward component, but more explicit safeguards may be warranted. Our training data comes from existing public datasets, but deployed systems would require careful consideration of privacy implications. The multi-agent framework could be extended to include fairness-focused agents ensuring recommendations do not perpetuate demographic biases present in training data.

References

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.

Konstantina Christakopoulou, Filip Radlinski, and Katja Hofmann. 2016. Towards conversational recommender systems. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 815–824.

Huy Dao, Yang Deng, Dung D Le, and Lizi Liao. 2024. Broadening the view: Demonstration-augmented prompt learning for conversational recommendation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 785–795.

Tri Dao. 2023. Flashattention-2: Faster attention with better parallelism and work partitioning. *arXiv preprint arXiv:2307.08691*.

Yaroslav Ganin and Victor Lempitsky. 2015. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pages 1180–1189. PMLR.

Chongming Gao, Wenqiang Lei, Xiangnan He, Maarten de Rijke, and Tat-Seng Chua. 2021. Advances and challenges in conversational recommender systems: A survey. *AI Open*, 2:100–126.

Shirley Anugrah Hayati, Dongyeop Kang, Qingxiaoyang Zhu, Weiyan Shi, and Zhou Yu. 2020. Inspired: Toward sociable recommendation dialog systems. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, pages 8142–8152.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, et al. 2022. Lora: Low-rank adaptation of large language models. *Iclr*, 1(2):3.

Xu Huang, Jianxun Lian, Yuxuan Lei, Jing Yao, Defu Lian, and Xing Xie. 2025. Recommender ai agent: Integrating large language models for interactive recommendations. *ACM Transactions on Information Systems*, 43(4):1–33.

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

Walid Krichene and Steffen Rendle. 2020. On sampled metrics for item recommendation. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1748–1757.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR.

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

Prabir Mondal, Pulkit Kapoor, Siddharth Singh, Sriparna Saha, Jyoti Prakash Singh, and Amit Kumar Singh. 2023. Genre effect toward developing a multimodal movie recommendation system in indian setting. *IEEE Transactions on Consumer Electronics*, 70(1):2517–2526.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, et al. 2023. Toolllm: Facilitating large language models to master 16000+ real-world apis. *arXiv preprint arXiv:2307.16789*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741.

Subham Raj, Prabir Mondal, Daipayan Chakder, Sriparna Saha, and Naoyuki Onoe. 2023. A multi-modal multi-task based approach for movie recommendation. In *2023 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

Subham Raj, Sriparna Saha, Brijraj Singh, and Niranjan Pedanekar. 2025. Multimodal movie recommendation with multitasking architecture and learning user–movie representation: An empirical study. *IEEE Transactions on Computational Social Systems*.

Diederik M Roijers, Peter Vamplew, Shimon Whiteson, and Richard Dazeley. 2013. A survey of multi-objective sequential decision-making. *Journal of Artificial Intelligence Research*, 48:67–113.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. *Advances in neural information processing systems*, 36:68539–68551.

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

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

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

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.

Yancheng Wang, Ziyan Jiang, Zheng Chen, Fan Yang, Yingxue Zhou, Eunah Cho, Xing Fan, Yanbin Lu, Xiaojiang Huang, and Yingzhen Yang. 2024. Recmind: Large language model powered agent for recommendation. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 4351–4364.

Zihan Wang, Xiaocui Yang, Yongkang Liu, Shi Feng, Daling Wang, and Yifei Zhang. 2025. Muse: A multimodal conversational recommendation dataset with scenario-grounded user profiles. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 1027–1053.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837.

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*, pages 42–52.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822.

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*, pages 1006–1014.

A Virtual Tool Operations

A key insight motivating HARPO is that recommendation reasoning shares common cognitive structure across domains. A user seeking movie recommendations and one browsing fashion items both require understanding context, retrieving preferences from history, searching candidate items, ranking options by relevance, and explaining final choices. The specific tools differ—movie databases versus fashion APIs—but the abstract reasoning operations remain constant.

We formalize this intuition through Virtual Tool Operations (VTOs), a taxonomy of domain-agnostic reasoning primitives that abstract away implementation details while preserving the semantic structure of recommendation reasoning.

A.1 VTO Taxonomy

We define 21 VTOs organized into seven functional categories, as shown in Table 5. This taxonomy emerged from systematic analysis of 500 annotated recommendation dialogues spanning three domains: movies (ReDial), conversational search (INSPIRED), and fashion (MUSE). Through iterative coding, we identified recurring reasoning patterns that transcend domain-specific implementations.

A.2 Complete VTO Descriptions

Each VTOs represents a domain-agnostic reasoning primitive. Below we provide concise definitions of all operations, following a consistent schema.

A.2.1 Extraction Operations

`extract_entities` Purpose: Identify salient entities referenced in the user utterance.

Category	Operations

EXTRACTION	`analyze_sentiment`
	`extract_context`
	`extract_entities`

USER MODELING	`retrieve_preferences`
	`identify_constraints`
	`model_user_state`

RETRIEVAL	`search_candidates`
	`filter_results`
	`match_attributes`

RANKING	`rank_options`
	`compare_options`
	`select_best`

REASONING	`query_knowledge`
	`reason_over_graph`
	`infer_implicit`

INTERACTION	`explain_choice`
	`refine_query`
	`handle_rejection`

MEMORY	`track_history`
	`update_beliefs`
	`recall_context`

Table 5: Virtual Tool Operations taxonomy. These 21 domain-agnostic primitives abstract implementation details, enabling cross-domain transfer of learned reasoning patterns.

Inputs: User utterance; dialogue context.
Outputs: Normalized entity mentions.

`extract_context` Purpose: Extract situational or task-level context from the conversation.
Inputs: Dialogue history.
Outputs: Contextual constraints or descriptors.

`analyze_sentiment` Purpose: Infer user sentiment or affective cues relevant to recommendation decisions.
Inputs: User utterance.
Outputs: Sentiment polarity or intensity.

A.2.2 User Modeling Operations

`retrieve_preferences` Purpose: Retrieve explicit or implicit user preferences from history.
Inputs: Dialogue history; user profile.
Outputs: Structured preference representation.

`identify_constraints` Purpose: Identify hard or soft constraints expressed by the user.
Inputs: User utterance.
Outputs: Constraint set.

`model_user_state` Purpose: Maintain a latent representation of the user's evolving intent.

Inputs: Dialogue history.
Outputs: User state embedding.

A.2.3 Retrieval Operations

`rank_options` **Purpose:** Rank candidate items by predicted relevance.
Inputs: Candidate set; scoring signals.
Outputs: Ranked list.

`compare_options` **Purpose:** Compare multiple candidate items along specific dimensions.
Inputs: Candidate pairs; criteria.
Outputs: Comparative judgments.

`select_best` **Purpose:** Select final recommendation(s).
Inputs: Ranked candidates.
Outputs: Chosen items.

A.2.4 Ranking Operations

`query_knowledge` **Purpose:** Query external knowledge relevant to recommendation.
Inputs: Entities; relations.
Outputs: Retrieved facts.

`reason_over_graph` **Purpose:** Perform relational reasoning over structured data.
Inputs: Knowledge graph.
Outputs: Inferred relations.

`infer_implicit` **Purpose:** Infer unstated preferences or intent.
Inputs: Dialogue context.
Outputs: Implicit signals.

A.2.5 Interaction Operations

`explain_choice` **Purpose:** Generate explanations for recommendations.
Inputs: Selected items; reasoning trace.
Outputs: Natural language explanation.

`refine_query` **Purpose:** Update the query based on feedback.
Inputs: User response.
Outputs: Refined constraints.

`handle_rejection` **Purpose:** Adapt recommendations after rejection.
Inputs: Rejection signal.
Outputs: Updated candidate set.

A.2.6 Memory Operations

`track_history` **Purpose:** Maintain dialogue state across turns.
Inputs: Dialogue turns.
Outputs: Updated history.

`update_beliefs` **Purpose:** Update beliefs about user intent.
Inputs: New evidence.
Outputs: Updated belief state.

`recall_context` **Purpose:** Recall relevant past context.
Inputs: Dialogue history.
Outputs: Retrieved context.

A.3 Illustrative Examples of VTO Composition

Example: Compositional Recommendation Reasoning Consider a user request such as "I liked *Inception* suggest something similar but lighter." This response requires composing multiple VTOs. `extract_entities` identifies *Inception* as a reference item; `retrieve_preferences` recalls the user's preference for complex narratives; `search_candidates` retrieves similar films; `filter_results` excludes overly serious options based on the "lighter" constraint; and `explain_choice` articulates why the final recommendation satisfies both similarity and tone requirements.

A.4 Annotation Procedure

Obtaining VTOs annotations for training data requires balancing scalability with quality. We employ a three-stage hybrid approach:

Stage 1: Heuristic Bootstrapping. We develop keyword patterns and syntactic rules to identify obvious operation invocations. Questions containing phrases like "what about" or "can you suggest" trigger `refine_query`; comparative statements trigger `compare_options`; rejection phrases trigger `handle_rejection`. This rule-based approach provides initial labels for approximately 60% of dialogue turns with high precision but limited coverage of subtle cases.

Stage 2: LLM Refinement. For ambiguous cases where rules fail to produce confident labels, we prompt GPT-4 with few-shot examples demonstrating VTOs annotation. The prompt includes operation definitions, annotated examples showing compositional usage, and instructions to output VTOs sequences for each turn. This semi-automatic approach extends coverage while maintaining reasonable quality.

Stage 3: Human Validation. Expert annotators validate a random 20% sample of the combined

annotations, achieving 85% agreement with semi-automatic labels after adjudication. Systematic disagreements inform rule refinement in Stage 1, creating an iterative improvement loop. This procedure enables processing of 50K dialogue turns with annotation quality sufficient for training.

B Training Details

Training proceeds in four stages: supervised fine-tuning, preference optimization with CHARM, reasoning optimization with STAR, and multi-agent refinement with MAVEN. We detail the loss functions, optimization schedules, and hyperparameters used in each stage below.

Stage 1: Supervised Fine-Tuning. We fine-tune the base language model on conversational recommendation data with joint response generation and VTOs prediction:

$$\mathcal{L}_{\text{SFT}} = \mathcal{L}_{\text{LM}} + \lambda_v \mathcal{L}_{\text{VTO}} \tag{17}$$

where $\mathcal{L}_{\text{LM}}$ is standard language modeling loss over reference responses, and $\mathcal{L}_{\text{VTO}}$ is binary cross-entropy for VTOs sequence prediction. We employ LoRA (Hu et al., 2022) for parameter-efficient fine-tuning, enabling training on academic compute budgets.

This stage establishes baseline conversational and recommendation capabilities, teaching the model to generate fluent responses that invoke appropriate reasoning operations. The joint VTOs prediction ensures that learned representations encode reasoning structure useful for downstream components.

Stage 2: CHARM Preference Training. Using preference pairs $(\mathcal{C}, r^+, r^-)$, we train the hierarchical reward structure:

$$\mathcal{L}_{\text{CHARM}} = \mathcal{L}_{\text{pref}} + \lambda_d \mathcal{L}_{\text{domain}} \tag{18}$$

Preference pairs are constructed through three complementary approaches: (1) gold responses versus heuristically degraded variants (removing recommended items, introducing irrelevant suggestions); (2) high-quality versus low-quality responses generated by prompting the Stage 1 model at different temperatures with different prompts; and (3) human annotations for validation on a held-out subset.

This stage teaches the model to distinguish good recommendations from bad ones across multiple quality dimensions, establishing the reward structure that guides subsequent stages.

Stage 3: STAR Reasoning Training. We train STAR components using reward signals from CHARM:

$$\mathcal{L}_{\text{STAR}} = \mathcal{L}_{\text{value}} + \lambda_g \mathcal{L}_{\text{gen}} \tag{19}$$

where $\mathcal{L}_{\text{value}}$ supervises the value network to predict CHARM rewards for completed reasoning paths, and $\mathcal{L}_{\text{gen}}$ trains the thought generator to produce promising candidates.

The key insight is that value network training uses CHARM rewards as supervision, creating a distillation from slow reward computation (requiring full response generation and evaluation) to fast value estimation (requiring only hidden state processing). This enables efficient tree search at inference time.

Stage 4: MAVEN Refinement. Finally, we train the multi-agent system:

$$\mathcal{L}_{\text{MAVEN}} = \mathcal{L}_{\text{task}} + \lambda_a \mathcal{L}_{\text{agree}} \tag{20}$$

where $\mathcal{L}_{\text{task}}$ is overall recommendation quality from CHARM and $\mathcal{L}_{\text{agree}}$ encourages agent consensus. This stage teaches agents to collaborate effectively, producing coherent recommendations that benefit from multiple specialized perspectives.

C Prompts for Data Preprocessing

This appendix documents all LLM prompts used for data preprocessing. We use GPT-4o-mini (gpt-4o-mini-2024-07-18) with temperature 0.3 for classification and 0.9 for generation tasks.

C.1 VTO Annotation

The following prompt annotates conversation turns with Virtual Tool Operations. The `{vto_descriptions}` placeholder expands to all 21 VTO definitions from Table 5.

P1: VTO Annotation

```
Analyze this conversation turn
and identify which Virtual Tool
Operations (VTOs) the system
performed.

VTO Types:
{vto_descriptions}

Context: {context}
User: {user_input}
System: {system_response}

Output JSON: {"vtos": ["vto_name1",
"vto_name2"], "reasoning": "brief
explanation"}
```

C.2 CHARM Preference Generation

For preference learning, we generate contrastive pairs. Original dataset responses serve as chosen; LLM generates rejected responses with `quality="low"`.

P2: CHARM Preference

```
Generate a {quality} quality response
for preference learning.

Context: {context}
User: {user_input}
Domain: {domain}

If "high": Natural, helpful,
relevant recommendations
If "low": Misses context, wrong
approach, unhelpful

Response:
```

C.3 STAR Reasoning

For tree-of-thought reasoning in STAR, we generate intermediate reasoning steps.

P3: STAR Thought

```
Given the context, generate a
reasoning step for recommendation.

Context: {context}
User Query: {user_input}
Previous Thoughts:
{previous_thoughts}

Consider: What information do we
need? What VTOs should we apply?

Thought:
```

C.4 ReDial Dataset Prompts

ReDial processing uses three additional prompts for utterance classification, VTO assignment, and preference extraction.

C.4.1 Utterance Classification

P4: Utterance Type

```
Classify this utterance in a movie
recommendation conversation.

Context: {context}
Current utterance: {utterance}
Speaker: {speaker}
Contains movie mention: {has_movie}

Classify into ONE of these
categories:
- greeting: Initial greeting (hi,
hello)
- ask_preference: Recommender asking
what user wants
- provide_preference: User stating
their preferences
- recommend: Recommender suggesting
specific movies
- explain: Providing information
about movies
- accept: User accepting/liking a
recommendation
- reject: User rejecting/disliking a
recommendation
- ask_info: User asking for more
details
- provide_info: Giving details about
a movie
- thank: Thanking
- goodbye: Ending conversation

Return ONLY the category name,
nothing else.
```

C.4.2 VTO Assignment

P5: VTO Assignment

```
Assign Virtual Tool Operations (VTOs)
for this recommendation system
response.

Context: {context}
User input: {user_input}
System response: {response}
Utterance type: {utterance_type}

Available VTOs:
- extract_context: Extract
situation/occasion from conversation
- extract_entities: Identify movies,
genres, actors mentioned
- retrieve_preferences: Get user's
stated preferences
- identify_constraints: Find
requirements (genre, year, mood)
- search_candidates: Search for
matching movies
- filter_results: Apply filters to
narrow down
- rank_options: Order movies by
relevance
- compare_options: Compare multiple
movies
- explain_choice: Explain why
recommending
- refine_query: Ask clarifying
questions

Return a comma-separated list of 1-4
most relevant VTOs.
Example: search_candidates,
rank_options, explain_choice
```

C.4.3 Preference Extraction

P6: Preference Extraction

```
Extract user preferences from this
movie recommendation conversation.

Conversation:
{conversation}

Extract these preferences if
mentioned (return JSON):
{
"genres": ["list of genres
mentioned"],
"mood": "funny/scary/romantic/
```

```
exciting/thoughtful/null",
"actors": ["actors mentioned"],
"directors": ["directors
mentioned"],
"similar_to": ["movies they liked"],
"avoid": ["things they don't want"],
"era": "recent/classic/80s/90s/null"
}

Return ONLY valid JSON.
```

C.5 INSPIRED Dataset

INSPIRED emphasizes sociable dialogue with chitchat and persuasion. We extend utterance classification accordingly.

P7: INSPIRED Classification

```
Classify this utterance in a sociable
movie recommendation conversation.
INSPIRED conversations include social
chitchat and persuasion strategies.

Context: {context}
Current utterance: {utterance}
Speaker: {speaker}

Classify into ONE of these
categories:
- social_chat: Casual conversation,
personal anecdotes
- ask_preference: Asking about movie
preferences
- provide_preference: Sharing
preferences or opinions
- recommend: Suggesting specific
movies
- persuade: Using persuasion
strategies to convince
- explain: Providing movie
information or reasoning
- accept: Accepting a recommendation
- reject: Rejecting or expressing
disinterest
- ask_info: Requesting more details
- provide_info: Giving additional
information
- acknowledge: Brief acknowledgments
(okay, I see)
- greeting: Opening greetings
- closing: Ending the conversation

Return ONLY the category name,
nothing else.
```

C.6 MUSE Dataset

For multimodal MUSE, we process product images using BLIP-2 (Li et al., 2023) to generate textual captions, which are incorporated into the conversation context.

P8: MUSE Image Context

```
The following fashion item is shown
in the conversation:

[Image Caption: {blip2_caption}]
Product Attributes:
{product_attributes}


Context: {context}
User: {user_input}
System: {system_response}

Considering both the visual
product information and textual
conversation, identify which Virtual
Tool Operations (VTOs) the system
performed.

Output JSON: {"vtos": ["vto_name1",
"vto_name2"], "reasoning": "brief
explanation"}
```

Since DeepSeek-R1-Distill-Qwen-7B is text-only, MUSE images are processed through BLIP-2 caption surrogates.

C.7 Annotation Statistics

Table 6 summarizes annotation statistics across all three datasets.

Statistic	ReDial	INSPIRED	MUSE
Total turns	182,150	35,811	83,204
Heuristic (%)	61.2	54.8	58.3
LLM-annotated (%)	38.8	45.2	41.7
Human-validated (%)	20	20	20
Human-LLM agree (%)	93.2	91.7	89.4
Avg. VTOs/turn	2.4	2.7	2.9

Table 6: VTO annotation statistics across datasets.

C.8 Reproducibility

- **Model:** GPT-4o-mini (gpt-4o-mini-2024-07-18)
- **Temperature:** 0.3 (classification), 0.9 (generation)
- **Max tokens:** 50–100 (classification), 256 (generation)
- **Rate limit:** 500 requests/minute
- **Parallelization:** 10 workers via ThreadPoolExecutor

D Extended experimental details

D.1 Dataset Description

ReDial (Li et al., 2018) contains 10,006 movie recommendation dialogues collected via Amazon Mechanical Turk with workers alternating seeker/recommender roles. The dataset includes entity annotations linked to DBpedia and represents the most widely-used CRS benchmark. **Preprocessing:** We use the official train/test split and apply

LLM-based VTO annotation (GPT-4o-mini) to derive reasoning supervision. We verify annotation quality through manual inspection of 500 samples (93.2% agreement with human annotators).

INSPIRED (Hayati et al., 2020) comprises 1,001 conversations emphasizing sociable recommendation strategies with social chitchat, personal anecdotes, and persuasion. The implicit preference signals present a challenging setting for preference extraction. **Preprocessing:** We apply identical VTO annotation and filter conversations with fewer than 4 turns.

MUSE (Wang et al., 2025) is a recently introduced multimodal dataset with 7,000 fashion conversations including product images. User profiles are derived from realistic shopping scenarios rather than manual design. **Preprocessing:** For text-only baselines, we extract image captions using BLIP-2 (Li et al., 2023). For HARPO, we process images through the base model's vision encoder when available, otherwise use caption surrogates.

D.2 Evaluation Protocol

Recommendation Quality (Primary): Following established CRS protocols (Li et al., 2018; Wang et al., 2022b; Zhou et al., 2020), we report Recall@K ($K \in \{1, 10, 50\}$), NDCG@K ($K \in \{10\}$), and MRR@K ($K \in \{10\}$). For ranking evaluation, we employ the standard negative sampling protocol with 99 randomly sampled negatives per positive item, computing ranks among 100 candidates. This protocol balances computational efficiency with evaluation validity (Krichene and Rendle, 2020).

User Satisfaction (Primary): We evaluate predicted satisfaction and engagement using two approaches: **(1) Model-based:** CHARM reward scores normalized to $[0, 1]$. To address circularity concerns, we report correlation with human judgments. **(2) Human evaluation:** Three expert annotators rate 200 test samples per dataset on recommendation quality (Rec.Q), explanation quality (Exp.Q), and overall satisfaction (1-5 scale). Inter-annotator agreement (Fleiss' κ) exceeds 0.72 for all dimensions.

Generation Quality: BLEU-n ($n \in \{1, 4\}$), ROUGE-L, and Distinct-n ($n \in \{1, 2\}$) assess fluency, relevance, and diversity.

VTO Accuracy (Secondary): Precision, Recall, and F1 for Virtual Tool Operations prediction against LLM-annotated ground truth.

Statistical Testing: All results report mean $\pm$ standard deviation over 3 runs with different random seeds (42, 123, 456). Statistical significance is assessed via paired t-tests with Bonferroni correction for multiple comparisons; $p < 0.01$ unless otherwise noted.

Ablation Methodology. Component ablations in Table 4 use the following protocol:

- **w/o CHARM**: Train Stages 1, 3, 4 without Stage 2; value network uses task-completion signal instead of quality signal.
- **w/o STAR**: Replace beam search with greedy decoding at inference; value network unused.
- **w/o VTOs**: Train Stage 1 without VTO prediction auxiliary loss ($\lambda_v = 0$ in Eq. 18).
- **w/o BRIDGE/MAVEN**: Remove respective modules entirely from training and inference.

Note that component interactions mean individual contribution estimates are not strictly additive.

D.3 Implementation Details

Model Architecture: We use `DeepSeek-R1-Distill-Qwen-7B` as the backbone, selected for its strong reasoning capabilities from R1 distillation. Parameter-efficient fine-tuning employs LoRA (Hu et al., 2022) with rank $r = 16$, $\alpha = 32$, applied to all attention projections (`q_proj`, `k_proj`, `v_proj`, `o_proj`) and MLP layers (`gate_proj`, `up_proj`, `down_proj`). Total trainable parameters: 42.7M (0.6% of base model).

Module Configurations: STAR: beam width $w = 3$, max depth $D = 3$, backtrack threshold $\tau = 0.3$. CHARM: 4 reward heads (relevance, diversity, satisfaction, engagement), $\beta = 0.5$ (preference strength), reference-free SimPO-style optimization. BRIDGE: 4 projection heads, adversarial $\alpha = 1.0$, gradient reversal for domain confusion. MAVEN: 3 agents (Recommender, Critic, Explainer), 2 communication rounds, attention-based weighting.

Training: AdamW optimizer with learning rates: SFT 5×10^{-5}, CHARM 2×10^{-5}, STAR/MAVEN 1×10^{-5}. Batch size 4 per GPU $\times$ 4 gradient accumulation $\times$ 2 GPUs = effective batch 32. Warmup ratio 10%, cosine decay. Max sequence length 512

Method	B-1	B-4	R-L	D-2	VTO-P	VTO-R	VTO-F1
ReDial							
UniCRS	21.6	3.4	19.2	0.32	–	–	–
DCRS	23.4	4.2	20.6	0.37	–	–	–
GPT-4	19.6	2.8	17.4	0.45	–	–	–
RecMind	21.4	3.5	19.0	0.40	0.56	0.52	0.54
HARPO	**25.2**	**4.8**	**22.1**	**0.47**	**0.81**	**0.77**	**0.79**
INSPIRED							
UniCRS	20.4	3.0	17.8	0.30	–	–	–
GPT-4	18.8	2.5	16.4	0.42	–	–	–
HARPO	**23.8**	**4.4**	**20.8**	**0.45**	**0.78**	**0.74**	**0.76**
MUSE							
Qwen2-VL-7B	23.2	4.0	20.1	0.39	–	–	–
GPT-4V	20.6	3.2	18.2	0.43	–	–	–
HARPO	**26.0**	**5.0**	**22.8**	**0.48**	**0.76**	**0.72**	**0.74**

Table 7: Generation quality and VTO accuracy. B-n: BLEU-n, R-L: ROUGE-L, D-n: Distinct-n, VTO-P/R/F1: VTO Precision/Recall/F1.

Method	Rec.Q	Exp.Q	Overall	κ
ReDial (n=200)				
UniCRS	3.18±0.12	2.86±0.14	3.04±0.11	0.73
DCRS	3.52±0.11	3.32±0.13	3.43±0.10	0.75
GPT-4	3.48±0.11	3.42±0.13	3.46±0.10	0.74
HARPO	**4.08±0.10**	**3.92±0.12**	**4.01±0.09**	0.78
INSPIRED (n=200)				
UniCRS	3.02±0.14	2.72±0.15	2.88±0.13	0.71
GPT-4	3.46±0.13	3.32±0.14	3.40±0.12	0.74
HARPO	**3.96±0.11**	**3.82±0.13**	**3.90±0.10**	0.77
MUSE (n=200)				
Qwen2-VL-7B	3.68±0.12	3.44±0.14	3.58±0.11	0.74
GPT-4V	3.52±0.13	3.38±0.14	3.46±0.12	0.73
HARPO	**4.16±0.10**	**4.00±0.12**	**4.09±0.09**	0.79

Table 8: Human evaluation (1-5 scale). Rec.Q: Recommendation Quality, Exp.Q: Explanation Quality. κ: Fleiss' kappa inter-annotator agreement. All HARPO improvements significant at $p < 0.01$ (Mann-Whitney U).

Setting	R@10	R@50	NDCG@10	Sat.	Eng.
Train: ReDial → Test: MUSE (Zero-Shot)					
UniCRS	6.2	16.4	4.8	0.28	0.24
HARPO w/o BRIDGE	17.8	34.6	12.1	0.42	0.38
HARPO (Full)	**25.4**	**46.2**	**17.2**	**0.56**	**0.52**
Δ vs. w/o BRIDGE	+42.7%%	+33.5%%	+42.1%%	+33.3%%	+36.8%%
Train: MUSE → Test: ReDial (Zero-Shot)					
UniCRS	7.8	20.2	5.6	0.30	0.26
HARPO w/o BRIDGE	15.4	32.8	10.4	0.44	0.40
HARPO (Full)	**22.2**	**44.6**	**15.0**	**0.58**	**0.54**
Δ vs. w/o BRIDGE	+44.2%%	+36.0%%	+44.2%%	+31.8%%	+35.0%%

Table 9: Cross-domain zero-shot transfer. BRIDGE enables 32–44% improvement across domains (Movies ↔ Fashion). All metrics in % except Sat./Eng. which are in [0,1].

tokens (covers 99.5% of samples without truncation). Training: 3 epochs (SFT), 2 epochs (other stages).

Hardware: 2× NVIDIA A100 80GB GPUs with BF16 mixed precision, Flash Attention 2 (Dao, 2023), and gradient checkpointing. Total training time: ∼2.5 hours.

Reproducibility: Code, datasets, and data processing scripts are available on the project page: https://harpo-bench.github.io. All hyperparameters were selected based on validation set performance; sensitivity analysis is provided in Section 4.6.

D.4 Reasoning Quality Analysis

Adaptive Depth: INSPIRED requires deeper reasoning (2.8 vs. 2.3 steps) due to implicit signals, demonstrating that STAR adapts search depth to problem complexity. **Productive Backtracking:** 14-21% backtrack rate indicates exploration; paths are abandoned when value predictions indicate poor outcomes. **Effective Value Learning:** 73-78% accuracy in predicting which paths lead to better recommendations demonstrates successful distillation from CHARM rewards.

D.5 Computational Analysis

Training Efficiency: The four-stage pipeline completes in 2.5 hours on 2×A100 GPUs, comparable to single-stage SFT baselines. **Inference Overhead:** STAR tree search adds 210ms latency (88ms → 298ms), acceptable for conversational settings as hown in Table 12. For latency-critical applications, STAR can be disabled with 9.4% R@10 degradation (Table 4).

D.6 Error Analysis

We analyze 100 failure cases where HARPO underperforms the best baseline.

Preference Hallucination (31%): When users provide minimal information, the model sometimes infers constraints not present. *Mitigation:* Improved uncertainty estimation could trigger clarification questions.

Over-diversification (24%): High diversity reward weight leads to unrelated recommendations. *Mitigation:* User-adaptive reward weighting based on explicit signals.

Context Truncation (19%): Long conversations (>512 tokens) lose early context. *Mitigation:* Efficient long-context handling or hierarchical context compression.

Parameter	Value	R@10	MRR	Sat.	Eng.
STAR Beam Width	$w = 1$ (greedy)	26.8	13.8	0.61	0.57
	$w = 3$ (default)	**29.8**	**15.6**	**0.68**	**0.64**
	$w = 5$	30.0	15.7	0.68	0.64
STAR Max Depth	$D = 1$	26.2	13.4	0.60	0.56
	$D = 3$ (default)	**29.8**	**15.6**	**0.68**	**0.64**
	$D = 5$	30.1	15.8	0.69	0.65
CHARM β	$\beta = 0.1$	27.6	14.2	0.62	0.58
	$\beta = 0.5$ (default)	**29.8**	**15.6**	**0.68**	**0.64**
	$\beta = 1.0$	28.6	14.8	0.66	0.62
LoRA Rank	$r = 8$	28.4	14.8	0.65	0.61
	$r = 16$ (default)	**29.8**	**15.6**	**0.68**	**0.64**
	$r = 32$	29.9	15.6	0.68	0.64

Table 10: Hyperparameter sensitivity on ReDial. Default values achieve near-optimal performance; increasing beam width/depth beyond defaults provides diminishing returns.

Metric	ReDial	INSPIRED	MUSE
Avg. Reasoning Depth	2.3	2.8	2.5
Avg. Branches Explored	4.6	6.2	5.4
Backtrack Rate (%)	14.2	20.6	17.4
Path Confidence (0-1)	0.74	0.67	0.71
Value Pred. Accuracy (%)	78.2	73.4	75.6
Quality Correlation (r)	0.72	0.66	0.69

Table 11: STAR reasoning analysis. INSPIRED requires deeper reasoning due to implicit preference signals; MUSE is intermediate with multimodal complexity.

Component	Time	Memory	Params	FLOPs
Training (2×A100-80GB)				
Stage 1: SFT	52m	68GB	42.7M	1.4T
Stage 2: CHARM	42m	62GB	8.2M	0.6T
Stage 3: STAR	46m	58GB	12.4M	0.7T
Stage 4: MAVEN	36m	56GB	6.8M	0.5T
Total Training	**2.9h**	68GB	70.1M	3.2T
Inference (per turn)				
Full HARPO	298ms	19GB	–	16.2B
w/o STAR	88ms	17GB	–	7.8B

Table 12: Computational requirements. Params: trainable LoRA parameters. FLOPs: floating-point operations. Training completed on 2×NVIDIA A100-80GB GPUs.

Error Type	%	Pattern
Preference hallucination	31	Inferring unstated constraints
Over-diversification	24	Too many unrelated options
Context truncation	19	Long history (>512 tokens)
VTO sequence error	14	Suboptimal reasoning order
Knowledge gap	12	Missing item attributes

Table 13: Error analysis on 100 failure cases.

E Discussion

We address key methodological considerations, limitations, and anticipated reviewer questions.

E.1 Evaluation Methodology

On Satisfaction Metrics and Circularity. Table 3 reports User Satisfaction and Engagement scores computed via CHARM's learned reward model. We acknowledge the potential circularity concern: using a trained component to evaluate the system containing it. We mitigate this through three mechanisms: (1) **Human correlation validation**—Pearson correlations between CHARM scores and independent human judgments (Table 8) range from $r = 0.64$ to $r = 0.73$, confirming that learned rewards capture meaningful quality dimensions rather than arbitrary artifacts; (2) **Held-out reward evaluation**—reward models are trained on preference pairs from training conversations only, then applied to test conversations; (3) **Proxy metric consistency**—improvements on CHARM-independent metrics (Recall, NDCG, MRR) parallel satisfaction gains, suggesting genuine quality improvements rather than reward hacking. Nevertheless, we emphasize human evaluation results (Table 8) as the primary evidence for user-aligned quality claims.

On Human Evaluation Scale. Our human evaluation comprises 200 samples per dataset rated by 3 expert annotators ($\kappa > 0.72$). While sufficient for statistical significance testing, larger-scale user studies with diverse annotator populations would strengthen ecological validity. We consider this important future work.

On Baseline Fairness. We deliberately preserve original backbones for published baselines rather than upgrading all methods to DeepSeek-R1-Distill-Qwen-7B. This choice reflects two considerations: (1) re-implementing complex systems like UniCRS or KGSF with different backbones risks introducing confounds that obscure method-specific contributions; (2) our SFT-only ablation (Table 4: R@10=21.6%) isolates backbone contribution—it performs comparably to UniCRS (21.2%) despite using a stronger backbone, indicating that HARPO's gains (+38% relative) stem from architectural innovations rather than backbone strength. For completeness, we note that upgrading all baselines to the same backbone would be a valuable but resource-intensive extension.

E.2 Novelty and Relationship to Prior Work

Relationship to Multi-Objective RL. CHARM's hierarchical reward decomposition relates to multi-objective reinforcement learning (Roijers et al., 2013), but differs in two key aspects: (1) **context-dependent weighting**—rather than fixed Pareto frontiers, our meta-learner adapts dimension weights based on conversational context; (2) **preference-based optimization**—we optimize via contrastive preference learning rather than policy gradients, avoiding reward scaling issues common in multi-objective RL.

Relationship to Tree Search Methods. STAR builds on tree-of-thought reasoning (Yao et al., 2023) but introduces domain-specific innovations: (1) the value network predicts *recommendation quality* rather than task completion; (2) quality is decomposed into interpretable dimensions enabling targeted backtracking; (3) VTO predictions at each node provide structured reasoning scaffolds absent in generic tree search.

On VTO Taxonomy Design. The 21 VTOs emerged from systematic analysis of 500 dialogues across three domains, not *a priori* design. We validated coverage by confirming that held-out dialogues required no additional operations. The taxonomy is intentionally general; domain-specific operations (e.g., `check_visual_similarity` for fashion) could extend it. Ablating VTOs entirely (-21.5% R@10; Table 4) confirms their utility, though sensitivity to taxonomy design warrants future investigation.

E.3 Limitations and Scope

Multimodal Claims. HARPO's strong MUSE performance uses BLIP-2 caption surrogates rather than native visual processing. We do not claim multimodal modeling; rather, we demonstrate that text-based reasoning abstractions transfer effectively to multimodal benchmarks when visual information is appropriately textualized. True multimodal HARPO variants integrating vision encoders are future work.

Cross-Domain Evaluation. We evaluate transfer between two domain pairs (Movies $\leftrightarrow$ Fashion). Broader evaluation across additional domains (e.g., music, books, restaurants) would better characterize BRIDGE's generalization. The current results are encouraging but not exhaustive.

Computational Overhead. STAR's tree search adds 210ms latency per turn (88ms $\rightarrow$ 298ms; Table 12). For real-time applications requiring $<$100ms responses, the "w/o STAR" variant retains 90.6% of full performance (R@10: 27.0 vs. 29.8) at baseline latency.

E.4 FAQ

Q: Why not use the same backbone for all baselines? Re-implementing published methods with different backbones risks introducing implementation artifacts. Our approach—preserving original implementations while adding an SFT-only ablation with the same backbone—isolates HARPO's contribution. The SFT-only baseline (R@10=21.6%) performs comparably to UniCRS (21.2%), confirming that gains are architectural.

Q: How sensitive is performance to the number of reward dimensions? We chose four dimensions (relevance, diversity, satisfaction, engagement) based on the recommendation quality literature (Jannach et al., 2021). Preliminary experiments with 2 dimensions (relevance + satisfaction) yielded 12% lower performance, while 6 dimensions (adding novelty, coverage) showed no improvement, suggesting four captures the essential quality aspects without redundancy.

Q: Could data contamination explain results on ReDial/INSPIRED? Three observations argue against this: (1) we evaluate *ranking* over candidate sets, not response memorization—contamination would not directly improve ranking ability; (2) SFT-only matches rather than exceeds prior SOTA, suggesting no unfair advantage from pre-training; (3) MUSE (December 2024) post-dates training data cutoffs, yet shows consistent improvements. Formal decontamination analysis remains future work.

Q: What is the annotation cost for VTOs? VTO annotation is a one-time preprocessing cost. Using GPT-4o-mini at \$0.15/1M tokens, annotating 300K turns cost approximately \$45. Human validation of 20% samples required $\sim$40 annotator-hours. This is comparable to other forms of training data curation.

Q: How do results change with different preference data quality? Stage 2 preference pairs are constructed via three methods: heuristic degradation, LLM-generated contrasts, and human annotation. Ablating human-annotated pairs reduces satisfaction correlation by 8%, while using only

heuristic pairs reduces it by 19%. This confirms that preference data quality matters, though the system remains effective with automated construction.

Q: Is HARPO applicable beyond conversational recommendation? The core innovations—hierarchical preference decomposition, quality-guided tree search, domain-agnostic reasoning abstractions—are not recommendation-specific. Applications to other subjective-quality tasks (e.g., dialogue systems, content generation) are plausible but unexplored.

F Human Evaluation Protocol

F.1 Annotation Task and Instructions

Human evaluation was conducted to assess recommendation quality along user-aligned dimensions such as relevance, satisfaction, and engagement. Annotators were presented with a conversational context and corresponding system responses, and were asked to provide either (i) scalar ratings on a 5-point Likert scale or (ii) pairwise preference judgments between two candidate responses.

Annotators were instructed to focus on how well the recommendations aligned with the user's expressed intent, how appropriate and diverse the suggested items were, and how likely the response would satisfy or engage a real user. The task involved no sensitive content, personal data, or deceptive elements. A brief task description and examples were provided prior to annotation to ensure consistency, and annotators could skip instances they found unclear.

F.2 Recruitment and Compensation

Annotators were recruited through a standard crowdsourcing platform and consisted of fluent English speakers with prior experience in conversational evaluation tasks. Participation was voluntary, and no personally identifying information was collected.

Annotators were compensated at rates aligned with or exceeding local minimum wage standards based on estimated task completion time. Compensation adequacy was monitored to ensure fair payment relative to annotator effort and regional norms.

F.3 Data Consent and Use

Annotators were informed that their judgments would be used solely for research purposes, including model evaluation and analysis, and that all collected data would be stored and reported in anonymized and aggregated form. Participation was voluntary, and annotators could withdraw at any time without penalty. No personally identifying information was collected or retained.