CANDIDATE-CONDITIONED IMPLICIT DEBATE FOR RECOMMENDATION AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Recommendation agents can refine candidate lists through multi-role discussion, but shared summaries may obscure candidate-specific evidence, ranking changes may misalign states and items, and previous experience must be reused without interfering with the current interaction. We propose *Candidate-Conditioned Implicit Debate* (CCID), a recurrent framework that addresses these challenges through continuous candidate states, persistent candidate identities, and conditional reflection memory. A proposer initializes the candidates and their states; utility and risk roles assess candidate fit and the consequences of replacing the current leader; and an adjudicator updates both rankings and states for further assessment. Before discussion, relevant lessons from completed offline trajectories are retrieved from a frozen memory snapshot and remain fixed throughout the request. Experiments on four recommendation benchmarks show that CCID consistently outperforms single-round discussion. Ablations support the contributions of candidate correspondence, communication capacity, and reflection memory. Paired top-1 analysis shows that recurrent discussion corrects more errors than it introduces on all four datasets. Additional rounds improve recommendation quality with diminishing gains, while adaptive stopping achieves quality close to fixed four-round discussion with fewer model calls. Together, these results support iterative candidate-level refinement that preserves evidence, incorporates complementary assessments, and reuses applicable experience.

1 INTRODUCTION

Recommendation agents use language models to interpret user histories, retrieve candidates, and revise recommendations through specialized reasoning roles. Their ability to reconsider an initial list is useful when a user's interests provide competing signals: long-term preferences may favor one item while recent behavior favors another. To resolve such conflicts, collaborating agents need to exchange the evidence behind candidate-level judgments and retain it as the list changes.

Text communication makes this difficult. As the upper path in Figure 1 illustrates, a short summary such as *smartphone + portable tech* captures a broad interest but collapses distinctions between individual phone candidates. A receiving role then has little basis for deciding which candidate to promote. The lower path instead assigns each candidate a continuous state, allowing the receiver to use information specific to that candidate. This motivates implicit communication through learned representations during recommendation refinement.

Rich messages also need stable references. If a state is attached only to a rank position, moving candidate A upward can associate its evidence with a different item. Subsequent roles would then revise the list using the wrong evidence. In addition, each revision changes the context for the next decision: once a candidate becomes the leader, promoting another item must be assessed against that updated list. Effective collaboration therefore requires both persistent candidate identity and feedback from the preceding revision.

Experience poses a related challenge across requests. A revision that helped one user may be inappropriate when another user's recent behavior conflicts with the same broad preference. Reusable lessons therefore need applicability conditions that connect past outcomes to the current decision. Retrieving these lessons before debate lets the ranking and candidate states evolve against a fixed body of prior experience.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

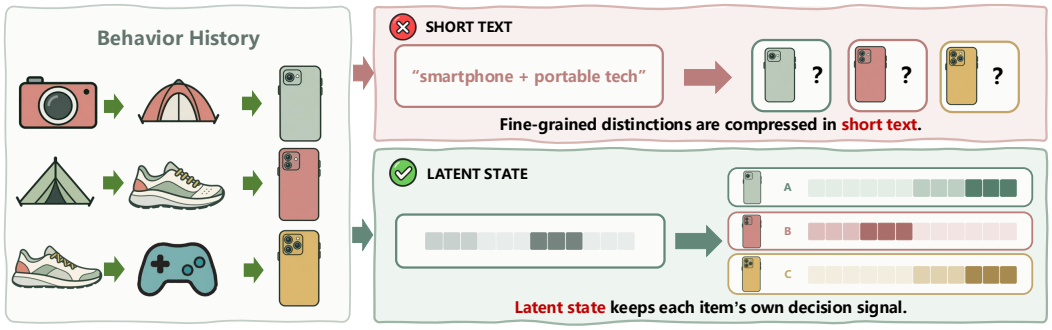

Figure 1: Candidate-specific evidence in recommendation. In the upper path, different histories collapse into the same short interest summary, obscuring distinctions between candidates. In the lower path, separate continuous states carry candidate-specific information to the receiving role.

To address these challenges, we propose *Candidate-Conditioned Implicit Debate* (CCID), an implicit-communication framework for recommendation agents. To preserve distinctions lost in shared summaries, we maintain a separate continuous state for every candidate. To keep evidence aligned through reranking, a shared candidate state board binds states and role replies to candidate identities. Utility and risk roles assess candidate fit and the risk of displacing the current leader; a candidate-wise adjudicator updates scores and states, then returns them for another assessment. To reuse experience across requests, reflection extracts conditional lessons from completed offline trajectories, and each new request retrieves relevant lessons from a frozen snapshot. Together, these mechanisms let agents revise a recommendation using both its evolving evidence and experience from earlier decisions. Our code is openly available at `https://anonymous.4open.science/r/CCID/`.

Experiments on multiple public datasets support this design. CCID improves NDCG@5 over text communication by 4.46%–11.46%, demonstrating the benefit of its continuous communication interface. Candidate reassignment and pooling comparisons show why preserving the association between evidence and individual candidates matters. Memory comparisons further show that selecting lessons by their applicability improves on retrieving them by similarity alone. Finally, adaptive stopping retains quality close to four-round discussion with fewer model calls, making repeated assessment more economical.

Our contributions are threefold:

- We propose Candidate-Conditioned Implicit Debate, a recommendation-agent framework that exchanges continuous candidate states and revisits decisions through recurrent utility–risk assessment of individual candidates.
- We develop a candidate identity-based communication interface and conditional reflection memory, preserving evidence through ranking changes and transferring applicable lessons across requests with related decision contexts.
- We evaluate the framework on multiple public datasets. The results establish the value of candidate-aligned communication and conditional memory, while adaptive stopping reduces calls by 26.77%–36.00% relative to fixed four-round discussion.

2 Related Work

Sequential and generative recommendation. Sequential models learn evolving preferences from interaction histories (Hidasi et al., 2015; Kang & McAuley, 2018; Sun et al., 2019), with extensions for multiple interests and heterogeneous preference signals (Chen et al., 2021; Lin et al., 2025; Hu et al., 2023; Xie et al., 2022). Generative recommenders represent items as tokens or semantic codes and produce recommendations autoregressively (Geng et al., 2022; Rajput et al., 2023; Wang et al., 2024a; Deng et al., 2025). These approaches provide the candidate lists on which subsequent agents can perform further candidate-level assessment.

Language models and recommendation agents. Language-model recommenders use prompting, alignment, retrieval augmentation, and reranking (Bao et al., 2023; Lyu et al., 2024; Liao et al., 2024; Chen et al., 2024; Gao et al., 2025). ReLLa improves comprehension of long behavior histories (Lin et al., 2023), and LC-Rec integrates language and collaborative semantics through alignment tasks (Zheng et al., 2024). Agent frameworks add planning, tools, and collaboration (Shi et al., 2024; Zhao et al., 2024; Huang et al., 2025; Wang et al., 2024b). Multi-agent recommenders distribute the workflow across specialized roles (Fang et al., 2024; Wang et al., 2024c). STARec equips user agents with fast responses and deliberate reasoning (Wu et al., 2025).

Learned communication. DIAL learns differentiable communication, CommNet uses continuous messages for cooperation, and TarMAC learns targeted communication (Foerster et al., 2016; Sukhbaatar et al., 2016; Das et al., 2019). Recent language-model systems explore latent collaboration and hidden-state exchange (Du et al., 2025; Zou et al., 2025; Peng et al., 2026; Shi et al., 2025). Recommendation adds an identity constraint: after a list is reordered, each message must still refer to the candidate it describes.

Verification and reusable experience. Candidate verification can refine retrieved recommendations (Zhang et al., 2026). InvariRank addresses sensitivity to candidate order in listwise reranking (Bito et al., 2026). LIGE-GR studies listwise generation and evaluation within an industrial ranking system (Srinivas et al., 2026). CoVeMem uses candidate-conditioned collaborative vector memory (Chen et al., 2026); AgentMMRec integrates reusable multimodal knowledge (Xu et al., 2026); SAGER develops evolving user-policy skills (Tao et al., 2026). Our focus is a request-level revision loop in which candidate states remain aligned after adjudication, with conditional lessons retrieved from a frozen memory for later requests.

3 Problem Setup

For request u, a proposer maps the history H_u available before a chronological cutoff to candidates $C_u = \{c_{u,1}, \ldots, c_{u,K_u}\}$ and initial ranking $\rho_u^{(0)}$. Candidate identities stay fixed during discussion; their rank positions can change. Index j always denotes a candidate identity.

At the start of round t, the shared board is

$$B_u^{(t)} = (C_u, \rho_u^{(t)}, Z_u^{(t)}, D_u^{(<t)}, M_u),$$

where $t = 0$ initializes the first round, Z contains candidate states, $D^{(<t)}$ contains earlier role responses (empty at $t = 0$), and M_u is retrieved memory. A role's message contains tuples $(c_{u,j}, z_{u,v,j}^{(t)}, q_{u,v,j}^{(t)})$, where v indexes the sending role: candidate identity, continuous payload, and sender/round metadata. Identity routes the payload when ranking positions change.

Roles V_v assess the board, and adjudicator F_ϕ, where ϕ denotes learnable parameters, updates it:

$$D_u^{(t)} = \{V_v(H_u, B_u^{(t)}) : v \in \mathcal{V}\}, \tag{1}$$

$$(\ell_u^{(t+1)}, Z_u^{(t+1)}) = F_\phi(H_u, B_u^{(t)}, D_u^{(t)}). \tag{2}$$

After adjudication, sorting $\ell_u^{(t+1)}$ in descending order gives $\rho_u^{(t+1)}$. The updated states and accumulated replies then become the input to the next round, allowing its roles to assess the revised list. Meanwhile, M_u remains fixed within the request: offline reflection constructs it from earlier training and validation trajectories. The held-out target enters evaluation after inference.

Metrics. Let r_u be the final rank of the single held-out target, with $r_u = \infty$ when absent. All N test requests contribute to

$$\text{P@1} = N^{-1} \sum_u \mathbb{1}[r_u = 1], \quad \text{HR@5} = N^{-1} \sum_u \mathbb{1}[r_u \le 5], \tag{3}$$

$$\text{NDCG@5} = N^{-1} \sum_u \frac{\mathbb{1}[r_u \le 5]}{\log_2(r_u + 1)}. \tag{4}$$

Absent targets contribute zero. Correction and damage compare two settings' final top-1 outcomes on the same requests; both rates use this complete denominator.

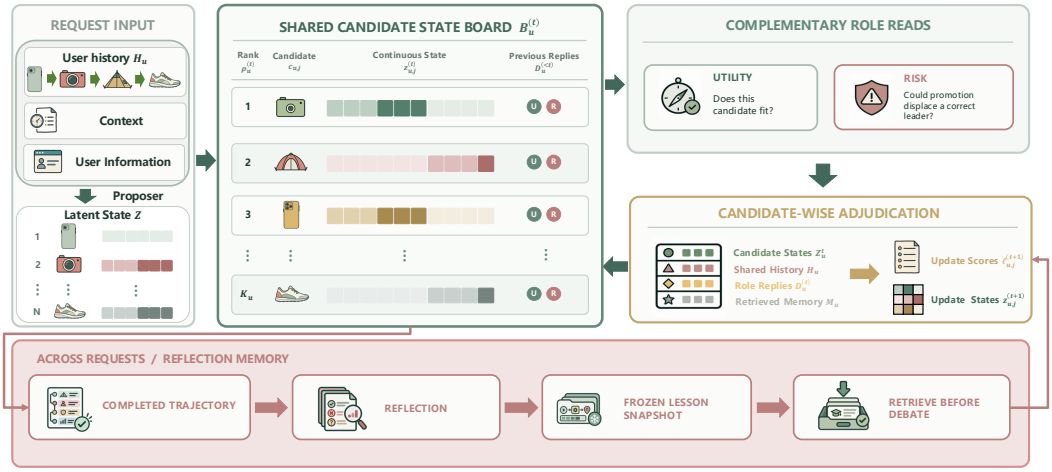

Figure 2: Candidate-Conditioned Implicit Debate. The proposer initializes candidate states on the shared board. Utility and risk roles read these states and contribute complementary replies; candidate-wise adjudication updates scores and states for the next round. Across requests, reflection turns completed trajectories into frozen lesson snapshots, from which relevant experience is retrieved before debate.

4 Candidate-Conditioned Implicit Debate

The challenges in Section 1 motivate three connected operations: retaining candidate-specific evidence, revisiting decisions after the ranking changes, and reusing relevant experience. Figure 2 shows how CCID implements them. A shared candidate state board connects the proposer, complementary utility and risk roles, and candidate-wise adjudication. Reflection memory supplies conditional lessons to the same adjudication step, so request-level refinement and cross-request experience contribute to a single decision process.

4.1 Starting from candidate-specific evidence

A shared interest summary can erase the evidence that distinguishes two plausible candidates. We therefore initialize the board directly from a frozen proposer's candidate representations. After it generates the initial list, we replay the generation sequence and retain the final-layer representation $h_{u,j}^{(0)} \in \mathbb{R}^{d_h}$ at each candidate position. An adapter maps this representation to a common communication space, $z_{u,j}^{(0)} = A(h_{u,j}^{(0)}) \in \mathbb{R}^d$. Padding is masked and duplicate codes are removed, giving each candidate one identity and one initial state.

These continuous states form the main message payload exchanged by the roles. Because a receiver can transform the full state, it can assess candidate evidence beyond a single ranking score. Short text can accompany a message for inspection. The continuous payload provides the information used by subsequent role computations, keeping the communication interface tied to the ranking task.

4.2 Keeping each signal with its candidate

As debate changes the ranking, evidence must remain attached to the item it describes. The shared candidate state board therefore indexes states and replies by candidate identity, while recording current rank separately. Role-specific adapters map states into the common communication space; sender and round metadata identify the source and version of each reply.

For example, if $c_{u,j}$ moves from third to first place, its index j continues to retrieve the same candidate's state and previous replies. The new rank records the effect of adjudication without changing the identity used to route messages. This separation lets later roles reconsider a promotion in the context of the updated list while retaining the evidence that motivated it.

4.3 COMPLEMENTARY ASSESSMENT AND CANDIDATE-WISE ADJUDICATION

Promoting an appealing item can still displace a better leader. CCID addresses these two aspects through complementary role reads. The utility role asks, "Does this candidate fit?" The risk role asks, "Could promotion displace a correct leader?" Their offline supervision targets are $\mathbb{1}[c_{u,j} = y_u]$ and $\mathbb{1}[a_u^{(t)} = y_u,\ c_{u,j} \neq y_u]$, respectively, where $a_u^{(t)}$ is the current leader. At inference, each role uses the visible history, candidate states, and prior replies to estimate its target. Candidate references and state versions are checked before the replies enter adjudication.

The adjudicator then combines the current state of each candidate with its associated role replies, shared history, and retrieved memory. It produces updated scores and states through F_ϕ as defined in Section 3. The resulting candidate representation $h_{u,j}^{(t+1)}$ has two readouts:

$$\ell_{u,j}^{(t+1)} = w_{\text{score}}^\top h_{u,j}^{(t+1)}, \qquad z_{u,j}^{(t+1)} = \text{Proj}(h_{u,j}^{(t+1)}).$$

The score determines the new ordering, while the projected state returns the adjudicated evidence to the shared board. Since the board also retains earlier replies, the next round evaluates the consequences of the preceding revision. Thus, repeated assessment operates on an evolving recommendation instead of repeatedly judging the unchanged initial proposal.

4.4 STOPPING AFTER A STABLE DECISION

Additional rounds are useful while the ranking or the role assessments continue to change. After each completed round, the controller compares the new top-k order with the preceding order and checks the remaining role disagreement. It terminates when the order is unchanged and the monitored disagreement is at or below a validation-selected threshold ϵ. A single completed round can satisfy this condition. Otherwise, discussion continues until stability or the configured round or cost limit, and the controller returns the latest valid ranking. This rule allocates further assessment to requests whose decisions remain unsettled.

4.5 REFLECTION MEMORY ACROSS REQUESTS

Request-level feedback explains how a decision evolved, but similar conflicts can recur for other users. The reflection path converts completed offline trajectories into reusable lessons. Each trajectory includes the initial proposal, role replies, ranking changes, and final adjudication. Its observed outcome distinguishes preserved decisions, successful corrections, harmful revisions, and unresolved cases. Reflection associates these outcomes with the decision context in which each revision was made.

A lesson stores a trigger condition, conditional recommendation, suggested action, scope, confidence, and supporting trajectory references. For example, conflicting long-term and recent interests can trigger a lesson to recheck a proposed promotion. These records form a frozen lesson snapshot. Before debate, retrieval selects lessons applicable to the current history and candidates to construct M_u. Candidate-wise adjudication reads this memory alongside the role replies, allowing earlier experience to inform the current revision.

The retrieved lessons remain fixed throughout a request, so the changing inputs across rounds are its ranking, states, and accumulated replies. The snapshot is also frozen during evaluation. In this way, reflection supplies prior experience without mixing the current evaluation outcome into the recommendation for that same request.

4.6 LEARNING THE COMMUNICATION INTERFACE

Candidate-level and listwise supervision train the role readouts and communication adapters on offline trajectories. Utility supervision rewards recognition of the held-out target, while risk supervision identifies promotions that would displace a correct leader. Validation selects continuation and retrieval settings. With parameters fixed at inference, adaptation within a request comes from updating the shared board; reuse across requests comes from the retrieved lesson snapshot.

Table 1: Dataset statistics. Average length is interactions per user.

Dataset	Users	Items	Interactions	Avg. length
Movies	1,534,020	646,522	11,195,641	7.30
Beauty	9,159	19,450	39,427	4.30
Toys	1,446,010	703,307	8,136,525	5.63
Yelp	530,330	149,659	5,211,090	9.83

5 EXPERIMENTS

Our evaluation tests whether candidate-specific continuous communication improves recommendation, which parts of the interface account for the gain, and how much discussion is needed to retain these recommendation gains.

5.1 EVALUATION SETUP

We evaluate three Amazon categories—Movies&TV (Movies), Beauty, and Toys&Games (Toys)[1]—and Yelp.[2] Table 1 summarizes the datasets. Evaluation uses chronological histories and a single held-out target per request. P@1, HR@5, and NDCG@5 use all requests, including those whose target is absent from the candidate list, as defined in Section 3.

Implementation and comparisons. The frozen proposer is OneRec-1.7B (Deng et al., 2025). Its 2048-dimensional representation is mapped to a 256-dimensional communication state. The controller monitors the top five positions, permits at most five rounds, and stops after one stable round. Component comparisons retain the request and candidate protocol while changing discussion length, communicated information, or memory. Reference methods use an earlier fixed-top-20 evaluation with different candidate construction and evaluation populations; Table 2 separates these historical results from the current-protocol comparison. The reference methods cover sequential, generative, language-model, and agent-based recommendation; their citations are listed in Appendix C.

5.2 RECOMMENDATION QUALITY

Table 2 reports all three metrics across the four datasets. Under the current protocol, CCID improves NDCG@5 over text communication by 4.46% on Movies, 6.59% on Beauty, 7.01% on Toys, and 11.46% on Yelp. These improvements are significant at $p < 0.05$ across repeated runs. P@1 and HR@5 also increase on every dataset, showing that the gains extend to both the first recommendation and target inclusion in the top five. Thus, the continuous communication interface improves several aspects of the final ranked list. The component studies below examine whether its benefit comes from richer messages, candidate correspondence, repeated assessment, or retrieved experience.

5.3 CORE COMPONENTS

Table 3 tests the three central design choices: recurrent assessment, continuous messages, and reflection memory. Restricting the method to one round removes the opportunity to reconsider an adjudicated list. Full CCID improves NDCG@5 over that setting by 6.13%, 7.32%, 9.85%, and 10.88% on Movies, Beauty, Toys, and Yelp, respectively. The additional rounds also improve P@1 and HR@5 in every domain, supporting the value of continued refinement after collecting the initial utility and risk assessments.

Replacing continuous states with scalar scores causes the largest NDCG@5 reduction among the three ablations. A score communicates a ranking judgment but discards much of the representation that another role could reassess. Removing reflection memory also lowers all three metrics across all datasets. These comparisons show complementary contributions: continuous states carry useful candidate evidence, recurrence gives roles another opportunity to use it, and retrieved lessons improve the resulting decisions.

[1] http://snap.stanford.edu/data/amazon/productGraph/categoryFiles/

[2] https://www.yelp.com/dataset

Table 2: Recommendation results. P1, H5, and N5 denote P@1, HR@5, and NDCG@5; leading zeros are omitted. Reference and current-protocol results are grouped separately. CCID improves NDCG@5 over text communication at $p < 0.05$ across repeated runs.

Method	Movies			Beauty			Toys			Yelp		
	P1	H5	N5	P1	H5	N5	P1	H5	N5	P1	H5	N5
Sequential recommendation												
SASRec	.0068	.0267	.0167	.0922	.1227	.1091	.0041	.0085	.0064	.0035	.0121	.0078
BSARec	.0106	.0161	.0135	.1182	.1329	.1266	.0086	.0114	.0101	.0027	.0091	.0059
FMLP-Rec	.0114	.0236	.0179	.1192	.1376	.1292	.0086	.0149	.0120	.0042	.0115	.0079
Mamba4Rec	.0115	.0228	.0176	.1094	.1282	.1197	.0081	.0140	.0113	.0041	.0094	.0068
HSTU	.0122	.0195	.0161	.1199	.1372	.1294	.0079	.0136	.0103	.0030	.0114	.0072
Generative recommendation												
TIGER	.0119	.0201	.0171	.1231	.1385	.1291	.0085	.0150	.0122	.0086	.0132	.0108
OneRec	.0201	.0278	.0241	.1302	.1698	.1547	.0187	.0249	.0211	.0103	.0181	.0149
LLM-based recommendation												
TALLRec	.0123	.0286	.0207	.1234	.1533	.1400	.0121	.0204	.0164	.0064	.0082	.0073
HLLM	.0256	.0448	.0349	.1462	.1708	.1566	.0179	.0317	.0265	.0117	.0195	.0152
ItemRAG	.0209	.0454	.0311	.1306	.1656	.1467	.0148	.0314	.0235	.0127	.0176	.0155
Recommendation agents												
InterRecAgent	.0193	.0357	.0257	.1067	.1345	.1221	.0135	.0234	.0178	.0076	.0201	.0155
STARec	.0211	.0374	.0281	.1237	.1577	.1409	.0183	.0213	.0196	.0114	.0182	.0142
AgentRec	.0199	.0378	.0271	.1137	.1342	.1248	.0175	.0201	.0192	.0091	.0182	.0141
ToolRec	.0195	.0372	.0263	.1051	.1137	.1101	.0148	.0177	.0166	.0080	.0154	.0127
Current CCID evaluation protocol												
Text communication	.0234	.0517	.0381	.1674	.1857	.1746	.0186	.0322	.0271	.0123	.0240	.0192
CCID (Ours)	**.0266**	**.0560**	**.0398**	**.1707**	**.1951**	**.1861**	**.0212**	**.0336**	**.0290**	**.0132**	**.0287**	**.0214**

Table 3: Core component ablations. P1, H5, and N5 denote P@1, HR@5, and NDCG@5; leading zeros are omitted.

Method	Movies			Beauty			Toys			Yelp		
	P1	H5	N5	P1	H5	N5	P1	H5	N5	P1	H5	N5
Single round	.0248	.0500	.0375	.1590	.1832	.1734	.0190	.0316	.0264	.0127	.0266	.0193
Score-only	.0220	.0460	.0347	.1495	.1752	.1640	.0184	.0290	.0245	.0113	.0231	.0178
No memory	.0254	.0545	.0384	.1671	.1912	.1774	.0205	.0331	.0274	.0124	.0276	.0203
CCID (Ours)	**.0266**	**.0560**	**.0398**	**.1707**	**.1951**	**.1861**	**.0212**	**.0336**	**.0290**	**.0132**	**.0287**	**.0214**

5.4 Communication content and candidate correspondence

Figure 3 separates the information in a message from the candidate to which it is attached. Pooling the states preserves more useful information than scalar scores, but retaining separate candidate states adds another 11.04%–14.44% in NDCG@5. Candidate-level detail therefore matters beyond the availability of a continuous representation alone.

The reassignment controls explain why. Shuffling states among a user's candidates preserves their overall information while breaking their association with individual items; performance then falls below score-only communication on every dataset. Replacing states with those of another user performs worst. Both controls retain the message format, yet lose the benefit of the original interface. The consistent ordering supports our design choice to preserve user-relevant evidence and bind it to the candidate being judged.

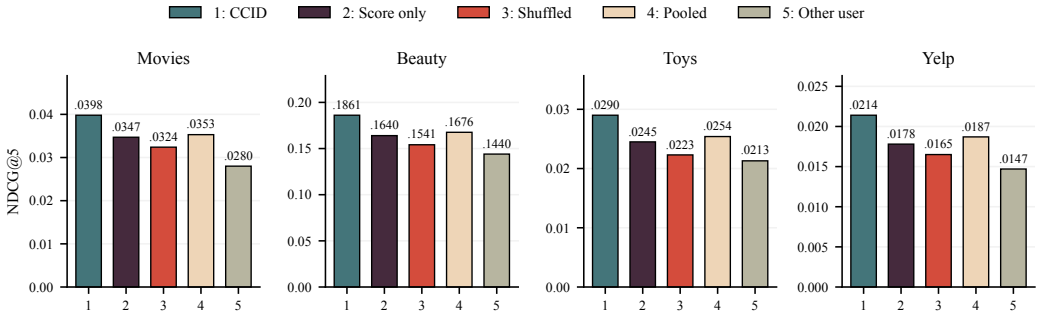

Figure 3: Communication content and candidate correspondence. Each panel compares five variants; all bars start at zero. Shuffled states come from the same user with candidate associations permuted; other-user states replace the current user's states. Exact values are in Table 7.

Table 4: Memory content and retrieval (NDCG@5).

Setting	Movies	Beauty	Toys	Yelp
CCID (Ours)	**0.0398**	**0.1861**	**0.0290**	**0.0214**
No memory	0.0384	0.1774	0.0274	0.0203
User-preference memory	0.0390	0.1823	0.0286	0.0205
Similarity-only retrieval	0.0394	0.1832	0.0280	0.0203
Random lesson retrieval	0.0391	0.1806	0.0281	0.0204

State capacity. Reducing the communication width from 256 to 128 retains 98.49%–100.00% of default NDCG@5 (Table 8). The relative loss grows to 3.60%–6.54% at 64 dimensions and 9.30%–13.08% at 32. This gradual decline shows that the interface tolerates moderate compression, while stronger compression removes information useful for ranking. Together with the reassignment controls, the result supports preserving both the content and the identity of each candidate's state.

5.5 Memory content and retrieval

Table 4 asks whether reflection helps because it provides any historical information or because it retrieves applicable decision lessons. CCID exceeds no memory by 3.65%–5.84% in NDCG@5. Preference summaries and random lessons also help, so part of the benefit comes from access to past information. However, conditioning retrieval on a lesson's applicability adds 1.02%–5.42% over similarity-only retrieval and produces the best score in every domain.

Similarity-only retrieval exceeds preference memory on Movies and Beauty but falls below it on Toys and Yelp. A superficially similar trajectory can therefore be a less useful guide than a simpler preference summary. The consistent advantage of conditional reflection supports storing when a lesson applies, alongside what action it recommends. This connects the memory design to the current decision instead of relying solely on resemblance between requests.

5.6 Corrections and harmful revisions

Table 5 compares the final predictions of CCID and single-round discussion on paired requests. It tests whether continued refinement corrects more first-choice errors than it introduces. On Beauty, 2.14% of all requests change from incorrect to correct and 0.97% change in the reverse direction, giving a net gain of 1.17 percentage points. The correction-to-damage ratios are 1.60, 2.21, 2.10, and 1.31 on Movies, Beauty, Toys, and Yelp: corrections exceed harmful revisions on every dataset.

Yelp has the smallest margin, with 0.21% corrections and 0.16% harmful revisions yielding a 0.05-point gain. The positive net change supports continued assessment, while its domain variation shows why candidate fit and displacement risk should be considered together. Counting both directions makes the aggregate P@1 gain interpretable as a balance of successful and unsuccessful revisions.

Table 5: Final top-1 outcomes relative to single-round discussion. Transition rates use all requests; the net change is in percentage points.

Setting	Movies	Beauty	Toys	Yelp
Single-round P@1 (%)	2.48	15.90	1.90	1.27
CCID P@1 (%)	2.66	17.07	2.12	1.32
Incorrect → correct (%)	0.48	2.14	0.42	0.21
Correct → incorrect (%)	0.30	0.97	0.20	0.16
Net change (pp)	0.18	1.17	0.22	0.05

Table 6: NDCG@5 by fixed discussion length and default policy.

Setting	Movies	Beauty	Toys	Yelp
Fixed 1	0.0375	0.1734	0.0264	0.0193
Fixed 2	0.0385	0.1815	0.0277	0.0202
Fixed 3	0.0394	0.1848	0.0286	0.0210
Fixed 4	**0.0401**	0.1860	**0.0290**	0.0213
Default adaptive	0.0398	**0.1861**	**0.0290**	**0.0214**

5.7 Discussion length, stopping, and cost

Table 6 tests whether additional discussion keeps improving the list. Quality rises from one to four fixed rounds, with smaller later gains. The adaptive policy differs from fixed four-round discussion by at most 0.0003 NDCG@5: it is slightly lower on Movies, equal on Toys, and slightly higher on Beauty and Yelp. Thus, its benefit is achieving comparable quality with fewer iterations.

The stopping distribution explains this saving (Table 9 and Figure 4). Mean discussion lengths are 2.84, 2.44, 2.66, and 2.71 rounds, and 72%–83% of requests finish within three rounds. Stability accounts for 94%–97% of stopping events. Most requests therefore reach the stopping condition before spending the full budget, while a smaller subset receives further assessment.

Table 10 connects this allocation to resource use. Compared with fixed four-round discussion, adaptive stopping reduces model calls by 26.77%–36.00%, estimated full-pipeline time by 27.90%–39.75%, and estimated inter-role payload by 23.49%–38.96%. Adaptive stopping therefore retains the benefit of repeated assessment while avoiding a uniform four-round cost for every request.

6 Discussion and Limitations

The results support communication that preserves both candidate evidence and its association with an item. The shared board keeps these inputs accessible as decisions change, while conditional memory adds experience relevant to the current revision.

Evaluation uses offline held-out targets, and candidate generation determines which items discussion can recover. The single-round comparison measures the combined benefit of additional discussion and computation; candidate reassignment and memory variants isolate more specific design choices. Time and payload are estimates, with deployment latency also depending on hardware, batching, and software implementation details.

7 Conclusion

We presented Candidate-Conditioned Implicit Debate, which connects recommendation roles through continuous candidate states, recurrent adjudication, and conditional reflection memory. Experiments on multiple public datasets support preserving rich evidence and its candidate association, with applicable lessons adding further gains. Adaptive stopping retains quality close to four-round discussion with fewer calls. Together, these results establish candidate-aligned implicit communication as an effective interface for iterative recommendation refinement.

AI USE STATEMENT

AI tools assisted with manuscript writing and language polishing, literature retrieval and discovery (including related-work searches), and research ideation or execution. The authors take responsibility for the final manuscript, experimental claims, and cited sources.

ETHICS STATEMENT

This work studies data-driven recommendation agents. The experiments use public datasets, chronological held-out targets, and fixed evaluation protocols. Responsible use should account for biases in historical feedback and for the possibility that automated ranking changes amplify unwanted recommendations or exposure disparities.

REPRODUCIBILITY STATEMENT

The code is openly available at `https://anonymous.4open.science/r/CCID/`. Sections 3–4 define the method and evaluation interface. Section 5 describes the experimental setup and comparisons; Appendices A and B provide additional results on communication, stopping behavior, and resource use.

REFERENCES

Keqin Bao, Jizhi Zhang, Yang Zhang, Wenjie Wang, Fuli Feng, and Xiangnan He. TALLRec: An Effective and Efficient Tuning Framework to Align Large Language Model with Recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems*, pp. 1007–1014, 2023. doi: 10.1145/3604915.3608857. URL https://doi.org/10.1145/3604915.3608857.

Ethan Bito, Yongli Ren, and Estrid He. One Pass, Any Order: Position-Invariant Listwise Reranking for LLM-Based Recommendation. In *Proceedings of the 49th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 3625–3629, 2026. doi: 10.1145/3805712.3809952. URL https://doi.org/10.1145/3805712.3809952.

Hanchong Chen, Xing Tang, Lingjie Li, Xiongfeng Shan, and Xiuqiang He. When Memory Takes Gradients: Collaborative Vector Memory for Agentic Recommender Systems. *arXiv preprint arXiv:2608.26895*, 2026. URL https://arxiv.org/abs/2608.26895.

Junyi Chen, Lu Chi, Bingyue Peng, and Zehuan Yuan. HLLM: Enhancing Sequential Recommendations via Hierarchical Large Language Models for Item and User Modeling. *arXiv preprint arXiv:2409.12740*, 2024. URL https://arxiv.org/abs/2409.12740.

Wanyu Chen, Pengjie Ren, Fei Cai, Fei Sun, and Maarten De Rijke. Multi-interest Diversification for End-to-end Sequential Recommendation. *ACM Transactions on Information Systems*, 40(1):1–30, 2021. doi: 10.1145/3475768. URL https://doi.org/10.1145/3475768.

Abhishek Das, Théophile Gervet, Joshua Romoff, Dhruv Batra, Devi Parikh, Mike Rabbat, and Joelle Pineau. TarMAC: Targeted multi-agent communication. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 1538–1546. PMLR, 2019. URL https://proceedings.mlr.press/v97/das19a.html.

Jiaxin Deng, Shiyao Wang, Kuo Cai, Lejian Ren, Qigen Hu, Weifeng Ding, Qiang Luo, and Guorui Zhou. OneRec: Unifying Retrieve and Rank with Generative Recommender and Iterative Preference Alignment. *arXiv preprint arXiv:2502.18965*, 2025. URL https://arxiv.org/abs/2502.18965.

Zhuoyun Du, Runze Wang, Huiyu Bai, Zouying Cao, Xiaoyong Zhu, Yu Cheng, Bo Zheng, Wei Chen, and Haochao Ying. Enabling Agents to Communicate Entirely in Latent Space. *arXiv preprint arXiv:2511.09149*, 2025. URL https://arxiv.org/abs/2511.09149.

Jiabao Fang, Shen Gao, Pengjie Ren, Xiuying Chen, Suzan Verberne, and Zhaochun Ren. A Multi-Agent Conversational Recommender System. *arXiv preprint arXiv:2402.01135*, 2024. URL https://arxiv.org/abs/2402.01135.

Jakob Foerster, Ioannis Alexandros Assael, Nando de Freitas, and Shimon Whiteson. Learning to communicate with deep multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper/2016/hash/c7635bfd99248a2cdef8249ef7bfbef4-Abstract.html.

Jingtong Gao, Bo Chen, Xiangyu Zhao, Weiwen Liu, Xiangyang Li, Yichao Wang, Wanyu Wang, Huifeng Guo, and Ruiming Tang. LLM4Rerank: LLM-based Auto-Reranking Framework for Recommendations. In *Proceedings of the ACM on Web Conference 2025*, pp. 228–239, 2025. doi: 10.1145/3696410.3714922. URL https://doi.org/10.1145/3696410.3714922.

Shijie Geng, Shuchang Liu, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. Recommendation as Language Processing (RLP): A Unified Pretrain, Personalized Prompt & Predict Paradigm (P5). In *Proceedings of the 16th ACM Conference on Recommender Systems*, pp. 299–315, 2022. doi: 10.1145/3523227.3546767. URL https://doi.org/10.1145/3523227.3546767.

Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. Session-based Recommendations with Recurrent Neural Networks. *arXiv preprint arXiv:1511.06939*, 2015. URL https://arxiv.org/abs/1511.06939.

Hengchang Hu, Wei Guo, Yong Liu, and Min-Yen Kan. Adaptive Multi-Modalities Fusion in Sequential Recommendation Systems. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pp. 843–853, 2023. doi: 10.1145/3583780.3614775. URL https://doi.org/10.1145/3583780.3614775.

Xu Huang, Jianxun Lian, Yuxuan Lei, Jing Yao, Defu Lian, and Xing Xie. Recommender AI Agent: Integrating Large Language Models for Interactive Recommendations. *ACM Transactions on Information Systems*, 43 (4):1–33, 2025. doi: 10.1145/3731446. URL https://doi.org/10.1145/3731446.

Wang-Cheng Kang and Julian McAuley. Self-Attentive Sequential Recommendation. In *2018 IEEE International Conference on Data Mining (ICDM)*, pp. 197–206. IEEE, 2018. doi: 10.1109/icdm.2018.00035. URL https://doi.org/10.1109/icdm.2018.00035.

Sunwoo Kim, Geon Lee, Kyungho Kim, Jaemin Yoo, and Kijung Shin. ItemRAG: Item-Based Retrieval-Augmented Generation for LLM-Based Recommendation. *arXiv preprint arXiv:2511.15141*, 2025. URL https://arxiv.org/abs/2511.15141.

Jiayi Liao, Sihang Li, Zhengyi Yang, Jiancan Wu, Yancheng Yuan, Xiang Wang, and Xiangnan He. LLaRA: Large Language-Recommendation Assistant. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1785–1795, 2024. doi: 10.1145/3626772.3657690. URL https://doi.org/10.1145/3626772.3657690.

Jianghao Lin, Rong Shan, Chenxu Zhu, Kounianhua Du, Bo Chen, Shigang Quan, Ruiming Tang, Yong Yu, and Weinan Zhang. ReLLa: Retrieval-enhanced Large Language Models for Lifelong Sequential Behavior Comprehension in Recommendation. *arXiv preprint arXiv:2308.11131*, 2023. URL https://arxiv.org/abs/2308.11131.

Xiaolin Lin, Weike Pan, and Zhong Ming. Towards Interest Drift-driven User Representation Learning in Sequential Recommendation. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1541–1551, 2025. doi: 10.1145/3726302.3730099. URL https://doi.org/10.1145/3726302.3730099.

Chengkai Liu, Jianghao Lin, Jianling Wang, Hanzhou Liu, and James Caverlee. Mamba4Rec: Towards Efficient Sequential Recommendation with Selective State Space Models. *arXiv preprint arXiv:2403.03900*, 2024. URL https://arxiv.org/abs/2403.03900.

Hanjia Lyu, Song Jiang, Hanqing Zeng, Yinglong Xia, Qifan Wang, Si Zhang, Ren Chen, Chris Leung, Jiajie Tang, and Jiebo Luo. LLM-Rec: Personalized Recommendation via Prompting Large Language Models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pp. 583–612, 2024. doi: 10.18653/v1/2024.findings-naacl.39. URL https://doi.org/10.18653/v1/2024.findings-naacl.39.

Bo Ma, Hang Li, ZeHua Hu, XiaoFan Gui, LuYao Liu, and Simon Lau. AgentRec: Next-Generation LLM-Powered Multi-Agent Collaborative Recommendation with Adaptive Intelligence. *arXiv preprint arXiv:2510.01609*, 2025. URL https://arxiv.org/abs/2510.01609.

Yanwen Peng, Delvin Ce Zhang, Xi Wang, and Nikolaos Aletras. StateBridge: Training-free Hidden-state Alignment for Latent Communication in LLM Multi-Agent Systems. *arXiv preprint arXiv:2608.13317*, 2026. URL https://arxiv.org/abs/2608.13317.

Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan Hulikal Keshavan, Trung Vu, Lukasz Heldt, Lichan Hong, Yi Tay, Vinh Tran, Jonah Samost, Maciej Kula, Ed Chi, and Maheswaran Sathiamoorthy. Recommender Systems with Generative Retrieval. *Advances in Neural Information Processing Systems 36*, 36:10299–10315, 2023. doi: 10.52202/075280-0452. URL https://doi.org/10.52202/075280-0452.

Wentao Shi, Xiangnan He, Yang Zhang, Chongming Gao, Xinyue Li, Jizhi Zhang, Qifan Wang, and Fuli Feng. Large Language Models are Learnable Planners for Long-Term Recommendation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1893–1903, 2024. doi: 10.1145/3626772.3657683. URL https://doi.org/10.1145/3626772.3657683.

Xiangyu Shi, Marco Chiesa, Gerald Q. Maguire, Jr., and Dejan Kostic. KVComm: Enabling Efficient LLM Communication through Selective KV Sharing. *arXiv preprint arXiv:2510.03346*, 2025. URL https://arxiv.org/abs/2510.03346.

Yehjin Shin, Jeongwhan Choi, Hyowon Wi, and Noseong Park. An Attentive Inductive Bias for Sequential Recommendation beyond the Self-Attention. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 8984–8992, 2024. doi: 10.1609/aaai.v38i8.28747. URL https://doi.org/10.1609/aaai.v38i8.28747.

Venkat Srinivas, Chenzhang He, Sam Woodmansee, Shawn Lian, Wenjie Hu, Renjie Jiang, Ziheng Huang, Xinyuan Zhang, Zhihao Zheng, Zhuoran Yu, Rui Li, Lei Yuan, Ziwei Li, Jimmy Jia, Mert Terzihan, Ekrem Kocaguneli, Yiming Liao, Zhichen Zhao, Yue Yin, Yue Weng, Wanli Ma, Xufeng Cai, Weimiao Wu, Yezhou Huang, Du Zhang, Yukun Ding, Aaron Johnston, Yueming Wang, Zhaojie Gong, Yuting Zhang, Serena Li, Adithya Ganesh, Boying Liu, Haichuan Yang, Xialu Li, Matt Ma, Qunshu Zhang, Andrew Ton, Nathan Berrebbi, Neel Pawar, John Joshua Miller, Jayant Subramanian, Praveen Rathinavelu, Cheng Huang, Aadhar Sachdeva, Josh Karns, Andres Aaron Gutierrez, Neil Agarwal, Gustas Pladis, Vladimir Batygin, Gopal Ray, Aditya Priyadarshi, Shantanu Patil, Zhe Wang, Penny Pan, Yiping Han, Arun Singh, Guangdeng Liao, Bi Xue, Xinyao Hu, Yang Song, Yisong Song, Meihong Wang, Haotian Wu, Deepak Agarwal, and Ji Liu. LIGE-GR: A Smooth Leap from Ranking to Generative Recommendation in the LLM Era. *arXiv preprint arXiv:2609.18148*, 2026. URL https://arxiv.org/abs/2609.18148.

Sainbayar Sukhbaatar, Arthur Szlam, and Rob Fergus. Learning multiagent communication with backpropagation. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper/2016/hash/55b1927fdafef39c48e5b73b5d61ea60-Abstract.html.

Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pp. 1441–1450, 2019. doi: 10.1145/3357384.3357895. URL https://doi.org/10.1145/3357384.3357895.

Zhen Tao, Riwei Lai, Chenyun Yu, Weixin Chen, Li Chen, Beibei Kong, Lei Cheng, Chengxiang Zhuo, Zang Li, and Qingqiang Sun. SAGER: Self-Evolving User Policy Skills for Recommendation Agent. *arXiv preprint arXiv:2604.14972*, 2026. URL https://arxiv.org/abs/2604.14972.

Wenjie Wang, Honghui Bao, Xinyu Lin, Jizhi Zhang, Yongqi Li, Fuli Feng, See-Kiong Ng, and Tat-Seng Chua. Learnable Item Tokenization for Generative Recommendation. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pp. 2400–2409, 2024a. doi: 10.1145/3627673.3679569. URL https://doi.org/10.1145/3627673.3679569.

Yancheng Wang, Ziyan Jiang, Zheng Chen, Fan Yang, Yingxue Zhou, Eunah Cho, Xing Fan, Yanbin Lu, Xiaojiang Huang, and Yingzhen Yang. RecMind: Large Language Model Powered Agent For Recommendation. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pp. 4351–4364, 2024b. doi: 10.18653/v1/2024.findings-naacl.271. URL https://doi.org/10.18653/v1/2024.findings-naacl.271.

Zhefan Wang, Yuanqing Yu, Wendi Zheng, Weizhi Ma, and Min Zhang. MACRec: A Multi-Agent Collaboration Framework for Recommendation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 2760–2764, 2024c. doi: 10.1145/3626772.3657669. URL https://doi.org/10.1145/3626772.3657669.

Chenghao Wu, Ruiyang Ren, Junjie Zhang, Ruirui Wang, Zhongrui Ma, Qi Ye, and Wayne Xin Zhao. STARec: An Efficient Agent Framework for Recommender Systems via Autonomous Deliberate Reasoning. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, pp. 3355–3365, 2025. doi: 10.1145/3746252.3760995. URL https://doi.org/10.1145/3746252.3760995.

Yueqi Xie, Peilin Zhou, and Sunghun Kim. Decoupled Side Information Fusion for Sequential Recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1611–1621, 2022. doi: 10.1145/3477495.3531963. URL https://doi.org/10.1145/3477495.3531963.

Jinfeng Xu, Zheyu Chen, Shuo Yang, Jinze Li, Puzhen Wu, Zewei Liu, Zheng Lin, Jianheng Tang, Jing Yang, Wei Wang, Xiping Hu, and Edith Ngai. Agents as Knowledge Integrator and Utilizer in Multimodal Recommendation. *arXiv preprint arXiv:2608.29410*, 2026. URL https://arxiv.org/abs/2608.29410.

Jiaqi Zhai, Lucy Liao, Xing Liu, Yueming Wang, Rui Li, Xuan Cao, Leon Gao, Zhaojie Gong, Fangda Gu, Michael He, Yinghai Lu, and Yu Shi. Actions Speak Louder than Words: Trillion-Parameter Sequential Transducers for Generative Recommendations. *arXiv preprint arXiv:2402.17152*, 2024. URL https://arxiv.org/abs/2402.17152.

Benyu Zhang, Qiang Zhang, Rui Li, Qunshu Zhang, Devansh Tandon, and Neeraj Bhatia. Recommendation Retrievers Need Verifiers: Universal Generative Reranking for Sequential Recommendations. *arXiv preprint arXiv:2609.12270*, 2026. URL https://arxiv.org/abs/2609.12270.

Yuyue Zhao, Jiancan Wu, Xiang Wang, Wei Tang, Dingxian Wang, and Maarten de Rijke. Let Me Do It For You: Towards LLM Empowered Recommendation via Tool Learning. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1796–1806, 2024. doi: 10.1145/3626772.3657828. URL https://doi.org/10.1145/3626772.3657828.

Bowen Zheng, Yupeng Hou, Hongyu Lu, Yu Chen, Wayne Xin Zhao, Ming Chen, and Ji-Rong Wen. Adapting Large Language Models by Integrating Collaborative Semantics for Recommendation. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*, pp. 1435–1448. IEEE, 2024. doi: 10.1109/icde60146.2024.00118. URL https://doi.org/10.1109/icde60146.2024.00118.

Kun Zhou, Hui Yu, Wayne Xin Zhao, and Ji-Rong Wen. Filter-enhanced MLP is All You Need for Sequential Recommendation. In *Proceedings of the ACM Web Conference 2022*, pp. 2388–2399, 2022. doi: 10.1145/3485447.3512111. URL https://doi.org/10.1145/3485447.3512111.

Jiaru Zou, Ruizhong Qiu, Gaotang Li, Xiyuan Yang, Katherine Tieu, Pan Lu, Ke Shen, Hanghang Tong, Yejin Choi, Jingrui He, James Zou, Mengdi Wang, and Ling Yang. Latent Collaboration in Multi-Agent Systems. *arXiv preprint arXiv:2511.20639*, 2025. URL https://arxiv.org/abs/2511.20639.

Table 7: State content and candidate correspondence (NDCG@5).

Setting	Movies	Beauty	Toys	Yelp
CCID (Ours)	**0.0398**	**0.1861**	**0.0290**	**0.0214**
Score only	0.0347	0.1640	0.0245	0.0178
Within-user state shuffling	0.0324	0.1541	0.0223	0.0165
Shared pooled state	0.0353	0.1676	0.0254	0.0187
Cross-user replacement	0.0280	0.1440	0.0213	0.0147

Table 8: Communication dimension and NDCG@5.

Setting	Movies	Beauty	Toys	Yelp
256	**0.0398**	**0.1861**	**0.0290**	**0.0214**
128	0.0392	0.1835	**0.0290**	0.0212
64	0.0380	0.1794	0.0277	0.0200
32	0.0350	0.1688	0.0255	0.0186

A Additional Communication Results

Table 7 reports the exact values underlying Figure 3. The same ordering holds on all four datasets: CCID, pooled state, score-only, shuffled states, and other-user states. Pooling retains useful information, but attaching states to the wrong candidate harms ranking even when they come from the correct user. This supports candidate identity as part of the communication interface. The contrast between pooling and shuffling is informative: pooling removes candidate-level detail, whereas shuffling assigns evidence to the wrong item. Their different outcomes show why richer messages need correct routing to the intended candidate.

Table 8 varies only the communication width. Its near-matching scores at 128 dimensions and larger reductions at 64 and 32 distinguish moderate compression from compression that substantially degrades ranking quality. The default width preserves the strongest overall results, while the 128-dimensional setting offers a smaller state with little quality loss. Toys retains the same displayed score at 128 dimensions, while the other domains decline slightly. The larger losses under further compression suggest that useful ranking information is distributed across the state.

B Stopping Behavior and Resource Use

Figure 4 and Table 9 describe how the adaptive policy allocates discussion rounds. Each request contributes to one stopping-round category and one terminal-reason category. Stability can be reached on the last permitted round, so stopping at the round limit does not by itself imply budget exhaustion. The concentration at two and three rounds, together with the high stability share, explains how the policy maintains the quality in Table 6 while reducing average discussion length.

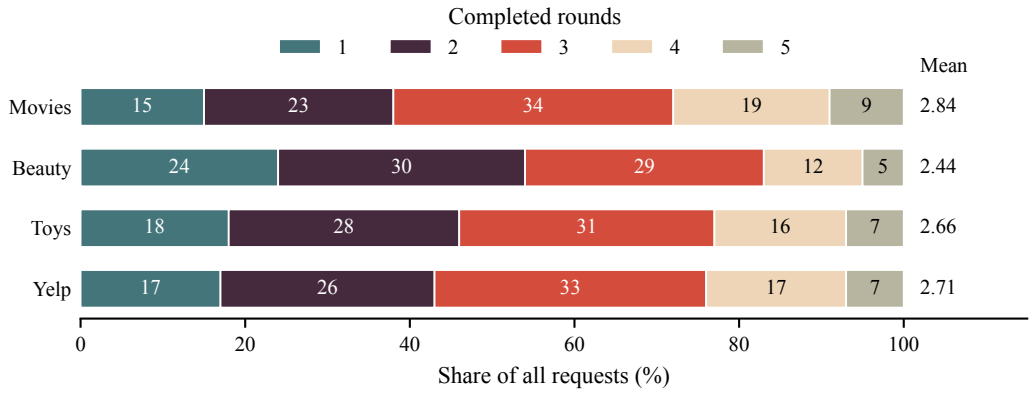

Figure 4: Adaptive stopping distribution. Each bar sums to 100% of requests; segment labels give percentages and the right column gives mean completed rounds.

Table 9: Default stopping statistics. Round shares and reason shares each use all requests; stopping can satisfy stability on round five.

Setting	Movies	Beauty	Toys	Yelp
Maximum rounds	5.00	5.00	5.00	5.00
Stop after round 1 (%)	15.00	24.00	18.00	17.00
Stop after round 2 (%)	23.00	30.00	28.00	26.00
Stop after round 3 (%)	34.00	29.00	31.00	33.00
Stop after round 4 (%)	19.00	12.00	16.00	17.00
Stop after round 5 (%)	9.00	5.00	7.00	7.00
Mean rounds	2.84	2.44	2.66	2.71
Stable (%)	94.00	97.00	96.00	95.00
Budget exhausted (%)	6.00	3.00	4.00	5.00

Beauty has the shortest mean discussion length and the largest first-round stopping share. Movies requires more rounds on average, showing that a common stopping rule yields different discussion lengths across domains.

Quality and resource tradeoff. Table 10 compares single-round, fixed four-round, and adaptive discussion. One round uses the fewest calls but gives lower NDCG@5 on all four datasets. Adaptive discussion approaches four-round quality while reducing calls and both estimated cost measures. It therefore offers an intermediate resource level that retains most of the benefit of extended discussion.

Table 10: Recommendation quality, model calls, and estimated costs per request. Time covers the full recommendation pipeline; payload is an estimated logical inter-role message volume.

Setting	Movies	Beauty	Toys	Yelp
Single: NDCG@5	0.0375	0.1734	0.0264	0.0193
Single: calls/request	4.00	4.00	4.00	4.00
Single: estimated time (ms)	464	477	473	489
Single: estimated payload (KiB)	3.57	4.00	3.59	4.24
Fixed four: NDCG@5	0.0401	0.1860	0.0290	0.0213
Fixed four: calls/request	13.00	13.00	13.00	13.00
Fixed four: estimated time (ms)	1484	1459	1452	1427
Fixed four: estimated payload (KiB)	14.69	16.71	16.76	14.95
Default: NDCG@5	0.0398	0.1861	0.0290	0.0214
Default: calls/request	9.52	8.32	8.98	9.13
Default: estimated time (ms)	1070	879	934	970
Default: estimated payload (KiB)	11.24	10.56	10.23	10.45

Time covers the complete recommendation pipeline and is estimated. Payload is the estimated logical volume exchanged between roles, in KiB (1024 bytes). Each saving is computed independently as $(\text{fixed four} - \text{adaptive})/\text{fixed four}$, while quality comparisons use absolute NDCG@5 differences. These definitions keep model calls, elapsed-time estimates, and communication volume distinct. The three resource measures improve together, making the saving consistent across several accounting views. Their different reductions also show why a reduction in calls should be interpreted alongside the separate time and payload estimates.

C Reference Methods and Reflection Records

The historical reference rows in Table 2 use a fixed-top-20 evaluation. Their candidate construction and evaluation populations differ from the CCID component studies. The within-protocol comparisons in Section 5 provide the direct evidence for the communication and memory design. The reference methods are SASRec (Kang & McAuley, 2018), BSARec (Shin et al., 2024), FMLP-Rec (Zhou et al., 2022), Mamba4Rec (Liu et al., 2024), HSTU (Zhai et al., 2024), TIGER (Rajput et al., 2023), OneRec (Deng et al., 2025), TallRec (Bao et al., 2023), HLLM (Chen et al., 2024), ItemRAG (Kim et al., 2025), InterRecAgent (Huang et al., 2025), STARec (Wu et al., 2025), AgentRec (Ma et al., 2025), ToolRec (Zhao et al., 2024).

Reflection records. Each record stores a trigger, lesson, suggested action, scope, confidence, and supporting trajectories. Retrieval selects applicable records before debate; they stay fixed while the request's ranking and states evolve.