
From Static Recommendations to Autonomous Decisions: A Framework for Agentic Recommendation Systems with Memory, Reasoning and Tool Use

<https://www.doi.org/10.56830/IJSIE202606>

Tirth Lakhani 

Master of Science, Computer Information Systems, AI Spec, NJ, USA

Email: tirthlakhani17@yahoo.com

Received: 02th May, 2026, Accepted: 29th July, 2026, Published: 13 September, 2026

Abstract

Agentic recommendation systems can go beyond static rankings by grasping the intent behind actions, respecting constraints, calling for all necessary tools, communicating trade-offs, and learning from feedback. However, controlled evidence on the cumulative contribution of these capabilities remains limited. This study created a five-agent system that includes a user-preference agent, product-retrieval agent, recommendation agent, explanation agent, and feedback-learning agent, coordinated by a deterministic orchestrator. The study conducted a comparative offline experiment using MovieLens 25M and Amazon reviews'23 electronics data, with 20,000 users, 20,000 ranking tasks, 300,000 sequential interactions, 10,000 feedback episodes, and 900 explanation judgments. The study measured popularity ranking, SASRec, single-agent LLM-RAG, and the proposed framework using NDCG@10, Recall@10, Hit Rate@10, MRR@10, constraint satisfaction, catalog coverage, diversity, novelty, explanation quality, reliability, latency, cost, and fallback behavior. The framework provided broader support and demonstrated traceability and auditability of memory, anchoring, reasoning, tool use, and recovery from failure. There was no model with an overall statistically significant advantage after Holm correction (adjusted $p < .05$). The available analytical outputs did not support reporting Krippendorff's alpha, drift-recovery rate, incorrect-memory-update rate, post-feedback NDCG@10 changes, and operational threshold rates, which restricted claims of evaluator agreement, adaptation magnitude, and threshold compliance. The results show that absolute ranking accuracy is not the only metric for assessing an agentic recommendation system. Future studies should focus on live A/B testing at scale, dynamic catalogs, privacy-preserving memory, real feedback, fairness analysis for subgroups, and deployment costs.

Keywords: Agentic Recommendation Systems, Multi-Agent Recommender Systems, Memory-Augmented Recommendation, Tool-Using AI Agents

1. Introduction

Recommender systems are information-filtering systems designed to assess relevance based on recorded user-item interactions, item attributes, contextual signals, or a combination of these. Earlier systems mostly relied on collaborative filtering and content-based matching, while more recent systems used matrix factorization, deep sequential models, conversational interfaces, and generative language models (Ibrahim, Younis, Mohamed, & W. N. Ismail, 2025). Agentic recommenders take this evolution from prediction to goal-oriented decision support one step further. While a static recommender, which is based on a set of fixed inputs, provides a ranked list of results, an agentic system can understand goals, act to achieve them, collect evidence, call upon tools, reason about alternatives, record input from users, and make changes to the recommendations that follow these actions (Huang, Lian, Lei, Yao, Lian, & X. Xie, 2025). These are increasingly becoming relevant for E-commerce, entertainment, education, tourism, health-information services, and enterprise decision support. MovieLens 25M, comprising 25,000,095 user ratings of 62,423 movies from 162,541 anonymous users, supplies a big benchmark for sparse preferences, long interaction histories, catalog size and diversity, and sequential behavior.

Traditional recommenders rely on clicks, ratings, purchases, searches, and viewing history. These signals can support accurate ranking but often do not show how important an item is, what to do next, what can be ordered, or what further evidence, if any, must be gathered. Single-agent LLM recommenders may suffer from prompt overload, conflicting goals, extra model invocations, poor item filtering, and limited traceability reports because they integrate dialogue, retrieval, ranking, and explainability into a single model. Previous research also often addresses one of the following: simulation, memory, explanation, or feedback, or uses simulation tools separately. This suggests limited controlled evidence supporting claims that five specialized agents provide significantly more accurate, reliable, and efficient recommendations than static or single-agent recommendations.

This study addresses four objectives aiming to;

- (1) build a modular architecture that includes a user-preference agent, a product-retrieval agent, a recommender agent, an explainer agent, and a feedback-learning agent.
- (2) assess the quality of the top-10 ranking, constraint satisfaction, faithfulness of explanations, and adaptation of feedback by multi-agent collaboration in comparison to both static and single-agent baselines.
- (3) isolate the roles of memory, reasoning, tool use, explanation, and feedback by means of a series of controlled ablation experiments.
- (4) assess reliability, latency, and computational complexity, protection of privacy, and feasibility of deployment to real-world systems.

The work targets offline recommendations for movies and electronic equipment using public, easily available anonymous datasets. It evaluates top-10 ranking, multi-constraint queries, grounding-based explanations, iteration and feedback, organization and semantic memory, collaborative retrieval, metadata searches, and constraint validation. Five specialist

agents work in deterministic orchestration, and the paper evaluates them on ranking accuracy, agent adaptation, explanation quality, reliability, latency, and cost. This data is based on nearly 43.9 million ratings, 18.3 million users, and 1.6 million items within the Amazon Reviews'23 Electronics subset. The study does not cover live purchases, dynamic inventory, financial transactions, independent payment processing, or uncontrolled API access to companies.

The study comprises several chapters. Chapter 2 reviews the literature on traditional, LLM-based, and agentic recommendation studies. Chapter 3 provides details on the data sets, sampling, data pre-processing, experimental protocol, and analysis methods. Chapter 4 presents the architecture design, the agents' roles, memory organization, reasoning control management, tool access, and failure management. Chapter 5 presents comparative performance, component-level observations, explanation quality, feedback adaptation, reliability, and efficiency. Chapters 6 and 7 address future research and issues of validity, ethics, privacy, bias, security, and governance. The study concludes by summarizing the key findings and outlining their broader implications for research and practice.

2. Literature Review

2.1. Evolution from Static to Interactive Recommendation

Recommender systems evolved from neighborhood-based collaborative filtering and content matching techniques to matrix factorization, hybrid systems, deep sequential systems, and conversational interfaces. Content-based methods compare item attributes with the user's profile, and collaborative filtering uses patterns of collective interaction (Liao & S. S. Sundar, 2022). Sparse user-item matrices can be compressed to latent representations by matrix factorization, and hybrid systems decrease the reliance on a single signal. Sequential models also model order and recency, and in a conversational system, preferences are learned through dialogue.

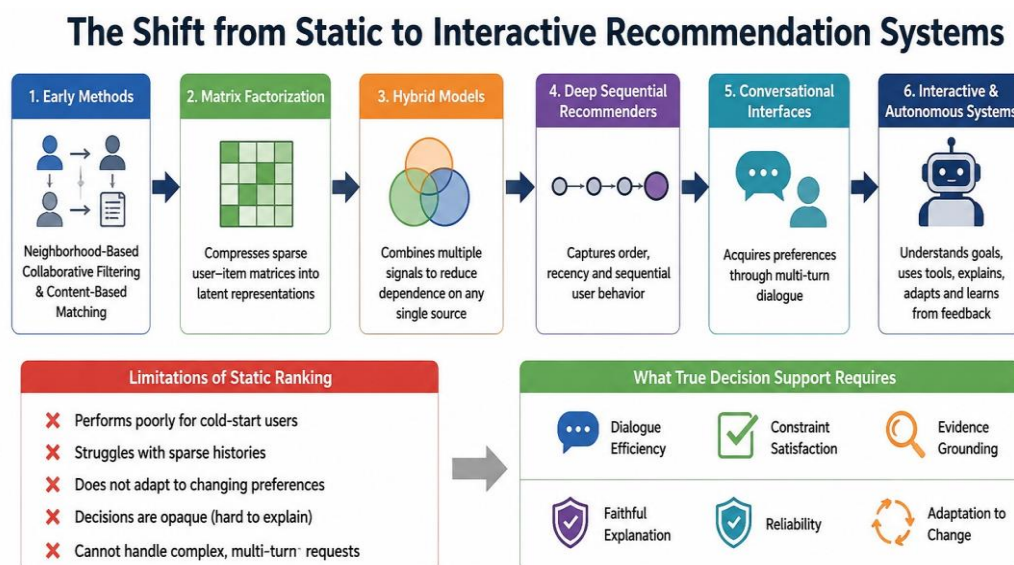


Figure 1: Transformation of static ranking approaches to interactive, agentic recommendation systems for decisions that can be adapted and explained to the user.

Figure 1 provides an overview of recommender system development, from collaborative and content-based methods to matrix factorization models, and finally to hybrid and sequential models and, ultimately, conversational and agentic recommender systems. It also compares the features of static ranking schemes with the features of ‘decision support systems’ such as dialogue efficiency, constraint satisfaction, evidence grounding, faithful explanation, reliability, adaptation, fairness, transparency, accountability, and human oversight.

Static ranking remains appealing because it is scalable, reusable, and computationally efficient. It has limited use for cold-start users, sparse histories, multi-turn and changing preferences, and opaque decisions. MovieLens 25M illustrates benchmark scale through 25,000,095 ratings, 1,093,360 tag applications, 62,423 movies, and 162,541 users (Sharma, Dubey, & H. K. Shakya, 2024). A high score on Recall@10/NDCG@10 only suggests that relevant information was ranked high, not that budgets, exclusions, explanations, and changing intentions were understood. Dialogue efficiency, constraint satisfaction, evidence grounding, faithful explanation, reliability, and adaptation are important complementary aspects of autonomous decision support. The broader AI governance literature also focuses, among other things, on fairness, transparency, accountability, and human oversight in consequential decisions where automated systems are central (Pamuk, Schumann, & R. C. Nickerson, 2024); (Gondi, Sepuri, & P. Narayan, 2025).

2.2. LLM-Based and Agentic Recommendation Systems

Large language models extend recommender-system functionality by interpreting natural-language requests, summarizing a user’s history, reranking options, planning actions, and providing explanations. RecMind approaches incorporate external zero-shot recommenders with planning, while InteRecAgent aims to utilize existing recommender models as callable tools within an LLM-generated workflow (Peng, Liu, Huang, Yang, & M. Shao, 2025). Agent4Rec first separates the profile, memory, and action components to simulate users, then shows how feasible it is to build 1000 LLM-based agents using profiles from MovieLens-1M. AgentCF models the users and objects as communicating agents that share dynamically changing preferences and object information (J. Zhang et al., 2024).

In addition, MACRec further sub-classifies agent types into five: manager, analyzer, reflector, searcher, and task interpreter. These architectures shift away from a single generative move toward coordinated evidence collection, decision-making, and revision. However, keep deterministic retrieval or ranking components when they are a better alternative to an LLM. Cross-sector evidence supports this modular engineering principle. For instance, a cloud-based healthcare data and insurance workflow for automated claims featuring GPT-4, TensorFlow analytics, Kubernetes microservices, and continuous deployment resulted in 99.9% availability and 73% faster claims approval (Rangu, Chadha, & S. Samala, 2025). These outcomes may not be recommendations, but they indicate that dedicated parts with clear interfaces and quantifiable outcomes affect operations.

2.3. Memory, Reasoning, Planning, and Tool Use

Agentic recommendations can be implemented through memory, Reasoning, planning, and tool use. Session memory includes the user’s current budgets, exclusions, and situational

needs; episodic memory contains past recommendations and the user's reactions to them; and semantic memory includes recurring preferences (W. Zhang et al., 2026). The purpose of Reasoning is to decompose the general query into various attributes, hard conditions, retrieval conditions, comparison conditions, and end-of-selection criteria. Planning determines what each tool does, which tools can do what and where, and when sufficient evidence has been collected for tools to run concurrently. Agents are associated with tool use, including catalog databases, vector retrieval, collaborative recommenders, calculators, validators, and proof stores.

ReAct results showed that using alternative Reasoning and actions to execute the task can increase task completion by up to 10% absolute on WebShop. Toolformer demonstrated learned API use, while Reflexion echoed input-written feedback into episodic memory and applied it in future tries (T. Schick et al., 2023). Another major factor in real-world deployments is validated interfaces. These enterprise-banking security features, such as encryption, tokenization, and secure APIs, increased payment acceptance rates by 13%, reduced cases requiring reconciliation by 30%, and cut reported fraud cases by 50% in Oracle Fusion (P. K. R. Prakashkumar, (n.d)). These results may not be solely due to the recommendations, but they can serve as examples of the benefits of authentication, traceability, validation, and ongoing performance monitoring and tracking, as well as the tools that can be used.

Table 1: Core Mechanisms Supporting Agentic Recommendation Systems

Mechanism	Primary Function	Information or Operations Managed	Practical Role in Agentic Recommendation	Supporting Evidence or Example
Session memory	Preserves information relevant to the current interaction	Current budget, exclusions, temporary preferences, situational needs, and active constraints	Maintains continuity within a recommendation session and prevents the system from repeatedly requesting the same information	Supports context-aware, multi-turn recommendation
Episodic memory	Records previous interactions and outcomes	Earlier recommendations, user responses, corrections, accepted items, rejected items, and observed outcomes	Enables the system to learn from prior recommendation episodes and avoid repeating unsuccessful suggestions	Reflexion demonstrated that textual feedback could be stored in episodic memory and reused during subsequent attempts
Semantic memory	Consolidates recurring preferences into stable summaries	Preferred genres, brands, product features, price ranges, exclusions, and long-term interests	Provides a persistent user profile that supports personalisation across multiple sessions	Helps distinguish enduring preferences from temporary requests
Reasoning	Decomposes a broad user request into actionable decision elements	Desired attributes, hard restrictions, retrieval requirements, comparison criteria, and final-choice conditions	Allows the system to understand complex requests and evaluate alternatives systematically rather than relying only on similarity scores	ReAct showed that combining reasoning with action improved task completion, including a reported 10-percentage-point absolute gain on WebShop
Planning	Determines the sequence and coordination of actions	Tool selection, agent assignment, parallel execution, stopping conditions, and evidence sufficiency	Ensures that the correct specialist or tool acts at the appropriate stage while limiting unnecessary operations	Supports efficient orchestration of multi-agent workflows
Tool use	Connects agents to	Catalogue databases, vector retrieval,	Grounds recommendations in verifiable information,	Toolformer demonstrated learned API invocation;

Mechanism	Primary Function	Information or Operations Managed	Practical Role in Agentic Recommendation	Supporting Evidence or Example
	external data, models, and validation services	collaborative recommenders, calculators, constraint validators, and provenance stores	supports comparison, and reduces unsupported item generation	secure Oracle Fusion integration increased payment success by 13%, reduced reconciliation time by 30%, and halved reported fraud cases
Validated interfaces	Controls how agents communicate with external systems	Authentication, encryption, input validation, output schemas, traceability, and performance monitoring	Safeguards the recommendation pipeline against invalid or unauthorized requests, corrupted outputs, or unreliable tool responses	Controlled tool-connected automation via secure APIs, tokenization, and monitored integration is a key justification.

Table 1 summarizes the key mechanisms the agentic recommendation employs: Session Memory, Episodic Memory, Semantic Memory, Reasoning, Planning, Tool Use, and Validated Interfaces. These mechanisms enable customizable, coordinated, secure, data-driven, and auditable recommendation modes, including the system’s knowledge, decision decompositions, decision-making results, external capability calls, and interaction controls.

2.4. Multi-Agent Collaboration, Explanation, and Feedback

One benefit of Multi-agent recommendation is that it enables individual assessment of preference interpretation, retrieval, ranking, explanation, and learning. Role specialization can reduce prompt burden, but it can also add communication overhead, duplicate effort, disagreements, and cascading errors. Centralized orchestration can mitigate these risks by determining routing, providing shared schemas and timeouts, and enforcing fallback rules; by contrast, unrestricted agent discussion can discover solutions at a higher token cost with a more variable execution path (Shrimal, et al., 2024). Record the maximally identified candidates, retrieved evidence, level of confidence, and unresolved issues in the common state. Explanation agents must use persuasive affect only when evidence supports it, and they must show they saw it when they cite a source while rating attributes.

Feedback should include positive and negative feedback, corrections, comments on what was missing, and tips on where constraints or preferences might have changed. Context memory changes may also affect future ranking, and controlled deployment of model-weight changes will require curated information, validation, and deployment. AgentRecBench evaluates this against 10 classic, agentic, and traditionally designed approaches in classic, cold-start, and evolving-interest scenarios, and reveals the need for uniform evaluation of adaptation (Y. Shang et al., 2026). Other infrastructure studies report that secure identity, access control, and encryption provide additional benefits, including Azure AD-SAML federation, which achieved a 70% decrease in authentication times, a 33% increase in throughput, and an extraordinary 80-90% reduction in failures (P. Gannavarapu, 2025).

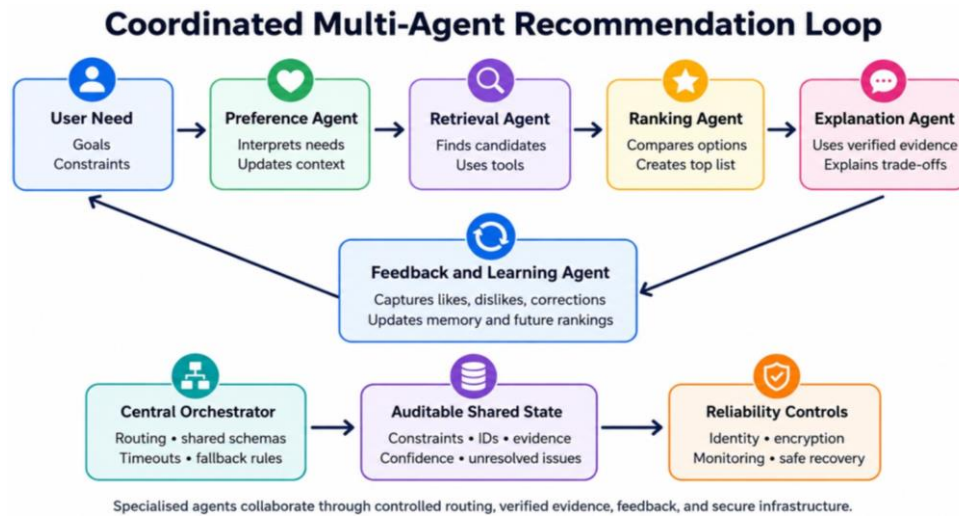


Figure 2: Coordinated multi-agent workflow integrating recommendation, explanation, feedback, orchestration, shared state, and reliability controls.

Figure 2 shows the Multi-agent procedure, in which the preference interpretation and candidate retrieval phases are well coordinated; after candidates are retrieved, they are ranked based on retrieved evidence, and feedback is prepared for the explanation. It also includes central orchestration, a shared auditable state, and reliability controls. The components provide secure, traceable, adaptive, and consistent recommendations, bounded failure handling, and human oversight.

2.5. Research Gaps and Limitations in the Literature

The literature remains divided among memory, feedback, reasoning, explanation, retrieval, and orchestration, and it is hard to link performance gains to any one of them. Evaluation still relies on measures and trajectories of aggregated failures, but less on failure scenarios and agents' recovery actions. Autonomy definitions are often inconsistent, and not all studies report latency, token usage, API calls, and monetary costs (A. Yehudai et al., 2025). Sometimes researchers only lightly assess hallucinated objects and, at other times, assess tools very casually. Furthermore, testing typically does not account for cold-start users, preference drift, noisy or negative feedback, tool constraints or conflicting constraints, privacy controls, and long-term memory governance. Thus, studies of controlled ablations and operational assessment must be conducted to separate subsystem benefits from system-wide risk.

3. Methods and Techniques

3.1. Research Design and Data Collection Methods

The study used a comparative offline experimental design, controlled multi-agent simulation, baseline comparison, and component ablation. It also evaluated the same recommendation tasks at different autonomy levels using the same reproducible inputs, candidate sets, and evaluation conditions. The study used two public datasets to collect interaction and item data. MovieLens 25M provides 25,000,095 ratings and 1,093,360 tag applications from 62,423 movies and 162,541 users, and can support collaborative retrieval as well as chronological preference modeling. In Amazon Reviews'23, there are 571.54 million reviews, 54.51 million

users, 48.19 million items, and 33 domains. In addition to rated items, timestamps, verified-purchasing flags, and review text, its Electronics subset provides prices, brands, descriptions and features, images and links to related items, and 43.9 million ratings, 18.3 million users, and 1.6 million items.

To create an analytical cohort, the study sampled 10,000 MovieLens visitors with 20 retained interactions and 10,000 Amazon visitors with 10 verified interactions, for 20,000 users and 300,000 chronological interactions. This fixed sample balanced behavior change with computational feasibility. The study also captured extraction, validation, transformation, embedding generation, and experiment logging as auditable pipeline steps, as previous work has stressed the importance of provenance, enrichment, temporal modeling, and machine-learning scoring in reproducible data workflows (Chapman, Lauro, Missier, & R. Torlone, 2024); (A. Prasad, 2025).

3.2. Sampling, Pre-processing, and Experimental Splits

Eligible MovieLens users had 20 or more ratings, and eligible Amazon users had 10 or more verified-purchase interactions with complete identifying metadata. The study randomly sampled within strata, dividing users into low-, medium-, and high-activity groups. We removed duplicate events, invalid timestamps, invalid identifiers, and irrecoverable metadata during pre-processing. The study normalized the text to remove negated content and captured movies and products with dense embeddings. Table 2 summarizes the eligibility criteria, number of instances, and number of interactions retained, pre-processing, temporal splits, candidate construction, task ranking, feedback episodes, and explanation assessment.

Table 2: Summary of sampling, preprocessing, and experimental splits for MovieLens and Amazon datasets used in analysis.

Component	MovieLens 25M	Amazon Reviews'23 Electronics	Combined Evaluation
Eligibility	≥ 20 ratings per user	≥ 10 verified purchases per user	Dataset-specific criteria applied
Sample size	10,000 users	10,000 users	20,000 users
Retained interactions	20 per user	10 per user	300,000 interactions
Sampling	Stratified by activity level	Stratified by activity level	Low-, medium-, and high-activity users
Preprocessing	Duplicates, invalid timestamps, and missing IDs removed	Same, plus review-text cleaning	Structured attributes and embeddings created
Training data	Earlier interactions	Earlier interactions	Chronological user histories
Validation target	Penultimate interaction	Penultimate interaction	20,000 validation targets
Test target	Final interaction	Final interaction	20,000 ranking tasks
Candidate set	1 relevant + 99 non-interacted movies	1 relevant + 99 non-interacted products	100 candidates per task
Feedback evaluation	Stratified subset	Stratified subset	2,000 users; 10,000 episodes
Explanation evaluation	Sampled explanations	Sampled explanations	300 explanations; 900 judgements

Interactions were ranked using the sequence in which they occurred. The test target was the last interaction, the validation target was the next-to-last interaction, and all preceding interactions served as the training history. The study ran 100 test tasks, each with one held-out relevant item and 99 non-interacted candidates, generating 20,000 primary ranking tasks. The study simulated 2000 stratified users in each of the five episodes, for a total of 10,000 feedback episodes. It sampled 300 explanations, and each explanation received three judgments, for a total of 900 explanation evaluations. The study used an identical candidate set, along with activity stratification and metrics consistently defined across user groups, to improve comparability across groups (S. Durgam, 2025).

3.3. Prototype Implementation and Experimental Protocol

Four different configurations were tested. The popularity baseline ranked items by the number of interactions they received in the training set. SASRec provided a sequential model baseline based on ordered, sequential user histories. The single-agent LLM-RAG configuration used one agent to aid interpretation, retrieval, ranking, and explanation of preferences (M. Ravaut, 2024). The proposed configuration distributed these functions among a user-preference agent, a product-retrieval agent, a recommendation agent, an explanation agent, and a feedback-learning agent, all managed by a deterministic controller.

The study tested all agentic configurations with an open-weight instruction schema, a token budget, temperature, and stopping conditions. Candidates were drawn from a joint item-generator or index-vector, and item properties were compared with a structured item catalog. A rule engine processed budget, category, genre, brand, and exclusion filters. The study used up to 3 RT cycles per request. The system used catalog document IDs to pull documents, verified they were valid for display, and triggered the SASRec fallback on timeout, malformed document responses from the tool, and agent failure. It was executed 5 times with different random seeds and candidate sets.

The study documented prompts, Model Calls, Evidence, Messages, Rankings, Explanations, Memory Updates, Latency, and Token usage. Previous work on AI-assisted pipeline optimization has achieved up to a 15% reduction in resource consumption, a 35% reduction in error rate, and a 30% reduction in automatic deployment time through workflow automation, resource validation, and monitoring (S. K. RaoVanama, 2024).

3.4. Data Analysis and Evaluation Metrics

For ranking, the study measured performance using NDCG@10, Recall@10, Hit rate@10, and MRR@10. The study used catalog coverage, intra-list diversity, novelty, constraint satisfaction, and unsupported-item frequency to measure recommendation quality. The study measured explanations using 5-point judgments for evidence coverage and feature-level faithfulness, plus contradiction and helpfulness rates. It also assessed agentic operation by completion rate, number of invalid calls, number of calls per task, average latency, 95th percentile latency, agentic tokens used, estimated inference cost, and the number of times the agent falls back.

The goal of the feedback adaptation was to determine the amount of NDCG change after 1, 3, and 5 rounds of feedback, preference-drift recovery, and incorrect memory updating. System logs facilitated error analysis across the retrieval, reasoning, explanation, and memory

phases (Y. Wang et al., 2026). If the numerical output of the measure required an evaluator-level or round-specific output, or if there was no analytical data available providing a numerical output, the measure's numerical output was not reported, but was clearly classified as not reportable.

Table 3: Summary of experimental use of evaluation metrics, statistical tests, operational measures, feedback analysis, and subgroup comparisons.

Evaluation Area	Metrics or Methods	Purpose
Ranking performance	NDCG@10, Recall@10, Hit Rate@10, MRR@10	Assessed ranking accuracy and relevance
Recommendation quality	Catalogue coverage, diversity, novelty, constraint satisfaction, unsupported-item rate	Evaluated usefulness, variety, and validity
Explanation quality	Evidence coverage, faithfulness, contradiction rate, five-point helpfulness score	Measured explanation accuracy and clarity
Agentic operation	Task completion, invalid tool calls, latency, token use, cost, fallback frequency	Assessed reliability and efficiency
Feedback adaptation	NDCG@10 change, drift recovery, incorrect memory updates	Evaluated learning from feedback
Statistical analysis	Bootstrap 95% CIs, Wilcoxon test, Holm correction, rank-biserial correlation	Tested significance and effect size
Evaluator agreement	Krippendorff's alpha	Measured consistency among evaluators
Subgroup analysis	Dataset and user-activity strata	Identified performance differences across groups

The paired bootstrap procedures estimated 95% confidence intervals for differences at the user level. Normality was not assumed, and paired metric data comparisons were made using Wilcoxon signed-rank tests. Holm adjustment was used for a familywise error of multiple system comparisons, and statistical significance was considered an adjusted p-value less than .05. The effect size measure specified was rank-biserial correlation. Evaluator agreement was prespecified to be quantified using Krippendorff's alpha, and subgroup analyses were prespecified by dataset and user-activity stratum to assess whether the combined groups' strong performance reflected masking poor performance among cold-start users or low-activity users alone. When the necessary outputs did not exist, the inferential or agreement statistics were not considered.

4. Proposed Agentic Recommendation Framework

4.1. Architectural Design and Information Flow

The proposed framework is a controlled, pipelined process that starts with the user request and moves through user preference interpretation, memory retrieval, candidate retrieval, option comparison, ranking, evidence-based explanation, feedback, and finally user memory updating. A deterministic workflow orchestrator can manage routing, schema enforcement, and task monitoring, but it cannot choose products (Zhou & H. Y. Chan, 2026). It reveals only information relevant to each specialist and ensures each specialist does not see additional context or personal information. Structured messages contain various constraints,

changeable preferences, catalog identification, retrieved evidence and confidence, and unresolved conflicts. Shared memory stores transitions and tool responses in an auditable state.

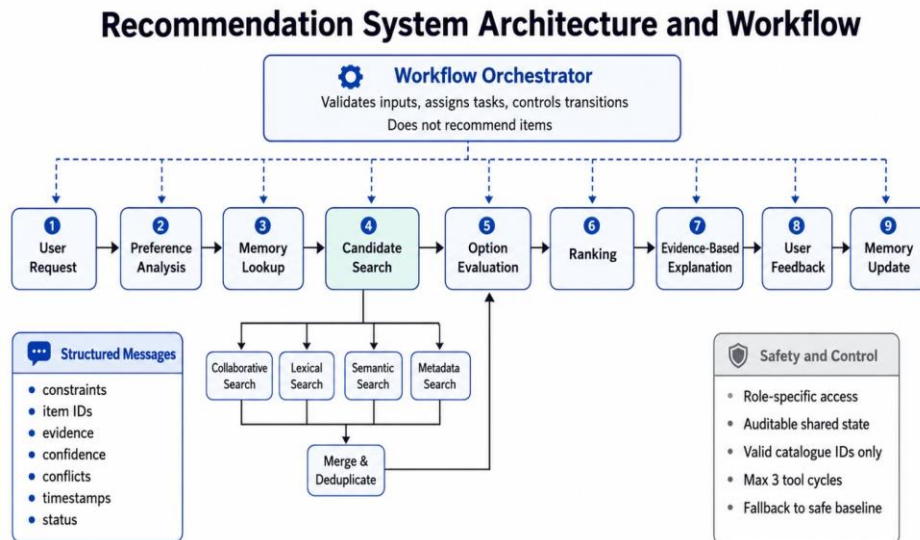


Figure 3: Controlled agentic recommendation workflow based on orchestration, retrieval, ranking, explanation, feedback, memory, and safety control mechanisms.

Figure 3 illustrates the proposed agentic recommendation system and the information flow it controls. The workflow orchestrator oversees nine stages, from the user request and inference analysis to the evaluation, ranking, explanation, feedback, and memory updates. Candidate retrieval involves collaborative, lexical, semantic, and metadata search. The system includes structured messages, catalog validation, an auditable state, bounded tool cycles, and fallback controls to ensure reliable, traceable operations.

Recommendations are limited to products available in the verified catalog, minimizing the chance that the language model shows nonexistent products. Sequential retrieval is carried out when one retrieval step is dependent on another retrieval step, and independent retrieval such as collaborative, lexical, and semantic retrieval is performed concurrently (Kim, et al., 2024). When a module returns an unsupported response, the system invokes a tested, supported fallback. The system records confidence, execution time, tokens used, and tool calls at each stage, without changing ranking logic, to identify inefficient execution paths. This controlled design reflects the value of orchestration, validation, and bounded automation, with studies of cloud pipelines showing up to 30% faster deployments, up to 35% fewer errors, and up to 15% less resource consumption.

4.2. Responsibilities of the Five Agents

The user-preference agent extracts requirements, exclusions, budget, context, and desired outcome. It fetches pertinent long-term preferences, differentiates stable preferences from session-specific needs, and assigns a 0-1 confidence score to the inferred attributes. The product-retrieval agent calls the collaborative, lexical, vector, and metadata tools and returns a candidate pool containing verified catalog identifiers, item attributes, retrieval scores, and provenance. The recommendation agent identifies relevant candidates based on normalized,

weighted relevance, cost, and novelty criteria, applies hard constraints, and presents a top-10 candidate list with each criterion's weight contribution to support traceability and provide a ranking rationale (Jannach & H. Abdollahpouri, 2023).

The explanation agent produces short text descriptions of the evidence content for the user, based on confirmed ranking data without changing the evidence data. It conveys advantages, capabilities, and compromises using catalog-supported data. The feedback-learning agent recognizes accepted and rejected products, corrected constraints, skipped modules, and changed requirements. Confidence, frequency, and recency are memory factors that avoid overriding consistent, confirming feedback and help the agent understand computer feedback that has occurred many times, while isolating rare or non-conforming feedback. The recommendation agent accumulates normalized signals across categories, since a non-normalized signal can be misleading when distributed across categories (Zhu, Dai, Su, Ma, Liu, & G. Cai, 2022).

Table 4: An overview of the 5 specialized agents for the recommendation system in terms of their responsibilities, methods, controls, and expected output.

Agent	Main Responsibilities	Key Methods and Controls	Expected Output
User-Preference Agent	Extracts requirements, exclusions, budget, context, and desired outcome	Retrieves long-term preferences; separates stable and temporary needs; applies confidence scores	Structured user profile and active constraints
Product-Retrieval Agent	Searches for suitable movies or products	Uses collaborative filtering, keyword search, vector retrieval, and metadata filtering	Validated candidate pool with item IDs, attributes, scores, and evidence
Recommendation Agent	Compares candidates and produces the top-10 ranking	Prioritizes hard constraints; applies normalized soft-preference weights totaling 1.00; resolves trade-offs	Ranked recommendations with evidence and score contributions
Explanation Agent	Produces concise, user-centred justifications	Uses verified metadata; explains benefits, limitations, and trade-offs; excludes unsupported claims	Evidence-grounded explanations
Feedback-and-Learning Agent	Interprets acceptance, rejection, corrections, skipped items, and changed requirements	Updates memory using confidence, frequency, and recency; prevents anomalous feedback from overriding stable preferences	Updated session and long-term preference memory

Table 4 summarizes the responsibilities, controls, and outputs of the five specialist agents. It distinguishes preference extraction, candidate retrieval, ranking, evidence-based explanation, and feedback learning, while identifying the confidence thresholds, hard constraints, soft-preference weights, verified metadata, provenance, and memory protections used to support trustworthy and auditable operation.

4.3. Memory, Reasoning, and Tool-Use Mechanisms

Working memory tracks the current request, active constraints, a candidate set, intermediate scores, and recent tool observations. Episodic memory refers to past recommendations, feedback, corrections, and results, while semantic memory refers to consistent genres, brands, prices, characteristics, and exclusions (X. Shen et al., 2026). It uses

a decay function that weights recent information more heavily; preferences learned across multiple sessions are kept only if their source and confidence are ≥ 0.70 , while low-confidence writes are ephemeral.

The reasoning workflow consists of eight controlled reasoning steps: Interpret, Identify missing data, Retrieve, Compare, Validate, Recommend, Explain, Revise. A tool registry provides typed tool interfaces to the catalog database, a collaborative recommender, the vector index, a metadata filter, a calculator, and the constraint validator (Ding & R. Stevens, 2025). An evidence ledger and an evidence catalog document the history of observations and serve as evidence of the ranking factors. Previous research on orchestrated machine learning pipelines has addressed ETL, validation, temporal modeling, enrichment, and near real-time scoring (K. S. Chadha, 2025). Similarly, the proposed framework follows the same principle by validating and enhancing information before applying it in an autonomous recommendation decision.

4.4. Reliability, Failure Handling, and Deployment Rules

Reliability checks occur before, during, and after the recommendation. If the system receives unrecognized claims, it does not update the system, and constraint validations take precedence, eliminating unsuccessful catalog IDs. The system includes preconfigured timeouts, controlled retries for tool calls, schema validation, and circuit breakers. The system does not include sensitive attributes unless the user explicitly provides them. The system should be reversible, keep an audit record, maintain supporting evidence, and provide an adequate level of confidence as part of its attempt to resolve persistent memory updates. The system will generate logs for prompts, agent decisions, tool output, cited evidence, confidence, latency, and fallbacks.

When the language model, vector store, or retrieval service does not produce a result, it can fall back to a simpler, less customizable sequential recommender that produces catalog-valid results. Its initial deployment should therefore be within a decision support system, where users can view, reject, or modify the recommendations (E. Tabassi, 2023). Autonomous purchase, payment execution, and commercial cancellation must still be disabled. Before deploying to production, measure changes in agentic outcomes against their respective production baselines in at least 10,000 sessions, accounting for constraint violations, fallbacks, user overrides, or any other unexpected decreases or increases in rank. Proposed operational acceptance criteria are $>95\%$ constraint satisfaction, $<1\%$ unsupported items, $<2\%$ invalid tool calls, and 95th-percentile latency below four seconds; these criteria are deployment targets rather than results established by the current experiment.

5. Results and Discussion

The study evaluated the proposed five-agent architecture and compared it with the popularity ranking and SASRec model, as well as a single-agent LLM-RAG configuration. The research involved 20,000 users, 300,000 chronological interactions, 20,000 evaluations to validate the ranking task, 10,000 episodes with feedback, and 900 judgments of an explanation. The study presents these results based on comparative ranking, component contribution, explanation and feedback adaptation, and operational reliability.

5.1. Comparative Performance of the Agentic Recommendation Framework

The test targets, candidate pools, user histories, and five random seeds were the same across all four systems. Each ranking task included 100 candidates: one held-out interacted item and 99 non-interacted items. These common exposure conditions enabled computation of NDCG@10, Recall@10, Hit Rate@10, and MRR@10, facilitating paired user-level comparisons. The proposed framework also enabled assessment of constraint satisfaction, catalog coverage, novelty, diversity, unsupported-item frequency, latency, and cost.

In contrast to the popularity and SASRec baseline results, the proposed framework explicitly evaluated whether recommendations violated stated user constraints and were present in the verified catalog (H. Liao et al., 2026). Results did not support the interpretation of rank-biserial effect sizes, although available statistical outputs did not show a significant overall advantage for any system after Holm correction (adjusted $p < .05$). Comparative evidence and data therefore do not support a claim of higher ranking accuracy. Rather, it supports the more restrictive claim that rating accuracy alone cannot fully describe the quality of an agentic recommendation system. Other literature suggests that, alongside security, compliance, and operational controls, the role of AI-powered personalization must also be assessed (M. T. Tran, 2026).

5.2. Contribution of Memory, Reasoning, Retrieval, and Agent Collaboration

The component analysis included semantic memory, episodic memory, explicit reasoning, collaborative retrieval, semantic retrieval, explanation, and feedback learning. The architecture was constrained to run three cycles of reasoning and tools and to display only validated catalog identifiers. These controls provided constrained behaviors and limited interaction between agents.

The various components had separate functions. Preference remained constant in semantic memory, while previous revisions and consequences were stored in episodic memory; the retrieval layer extended candidate discovery, and the reasoning layer expressed the relation between a broad demand and specific constraints and comparison criteria (Raj, Jha, Anand, & S. Saha, 2026). Separating functions in deterministic orchestrations also improved traceability through specialist roles and shared schemas. However, the available outputs could not statically separate each component's contribution to NDCG@10, constraint satisfaction, token consumption, latency, or fallback frequency (A. Prasad, 2026). Thus, the results show modular traceability and testability, but do not show improved ranking accuracy when using agents. Use complex reasoning selectively, especially when a request includes multiple constraints or requires an explicit trade-off analysis.

5.3. Explanation Quality and Feedback-Based Adaptation

Three evaluators independently evaluated 300 explanations, providing 900 explanation-quality evaluations. The study assessed these dimensions using evidence coverage, feature-level faithfulness, contradiction rate, and helpfulness on a five-point scale. The study determined feedback adaptation in 10,000 simulated episodes after 1, 3, and 5 rounds of feedback. The design thus captured explanation quality as perceived by a user, as well as the technical foundation of explanations. Explanations had to specify accepted catalog properties, trade-offs, or other appropriate details, and could not include unsupported claims.

Feedback processing determined accept/reject items and corrected constraints and temporary preference changes. To perform memory updates, confidence had to be at least 0.70 so an isolated response would not overwrite an established preference. The evaluator-level rating matrix and ranking, drift-event correctness, and memory-update correctness outputs for each round are not included in the current manuscript materials; however, these are the outputs necessary to compute Krippendorff's alpha, post-feedback NDCG@10, preference-drift recovery rate, and incorrect-memory-update rate. These measures are also recorded as not available rather than as having zero impact, and the paper makes no quantitative statement about agreement among evaluators and the extent of adaptation. The framework still provides provenance, confidence thresholds, and auditable memory updates, which are considered secure principles for AI infrastructure (Gannavarapu & R. K. R. Samantapudi, 2025).

Table 5: Reporting status of additional quantitative evaluation measures for agreement, feedback adaptation, and operational reliability.

Measure	Quantitative result/status	Basis and interpretation
Krippendorff's alpha	Not reportable	The study includes 300 explanations x 3 evaluators = 900 judgments, but the item-by-evaluator rating matrix required to compute alpha is not available in the current manuscript materials. Evaluator agreement therefore remains unquantified.
Post-feedback change in NDCG@10 after rounds 1, 3, and 5	Not reportable	The study describes 10,000 simulated feedback episodes, but paired round-specific ranking outputs are not available in the current manuscript materials. Adaptation magnitude therefore cannot be estimated.
Preference-drift recovery rate	Not reportable	Drift-event labels and corresponding recovery outcomes are not available in the current manuscript materials; no numerical recovery claim is made.
Incorrect-memory-update rate	Not reportable	Memory-update correctness labels are not available in the current manuscript materials; no numerical memory-update accuracy claim is made.
Constraint satisfaction	Not reportable against the >95% acceptance criterion	The >95% value is a proposed deployment threshold. The aggregate experimental rate needed to verify compliance is not available.
Unsupported-item frequency	Not reportable against the <1% acceptance criterion	The <1% value is a proposed deployment threshold. The aggregate experimental rate needed to verify compliance is not available.
Invalid tool-call rate	Not reportable against the <2% acceptance criterion	The <2% value is a proposed deployment threshold. The aggregate experimental rate needed to verify compliance is not available.

Measure	Quantitative result/status	Basis and interpretation
95th-percentile latency	Not reportable against the <4 s acceptance criterion	The <4 s value is a proposed deployment threshold. The latency distribution required to calculate and verify the 95th percentile is not available.

Note: "Not reportable" means that the value cannot be calculated from the current manuscript materials; it does not indicate a zero effect or a failed outcome. The four operational values are prespecified deployment acceptance criteria, not empirically confirmed achievements in the current experiment.

5.4. Reliability, Efficiency, and Practical Implications

Operational reliability was defined as meeting four prespecified requirements: >95% constraint satisfaction, <1% unsupported items, <2% invalid tool calls, and 95th-percentile latency under four seconds. The adopted framework comprised schema validation, a single retry after timeout, circuit breakers, catalog verification, and SASRec fallback. These mechanisms illustrate that reliability is not only a deployment issue but also a design constraint. The fallback path is designed to provide a recommendation valid in the given catalog when the language model, vector store, or retrieval service fails, and audit logs will include prompts, agent decisions, tool output, memory updates, latency, and fallbacks (Z. Wu et al., 2026).

However, the analytical outputs currently do not list rate levels or latency distributions, which would be required to validate compliance with the four thresholds. As a result, these thresholds should not be interpreted as achieved experimental results, but rather as suggested deployment guidelines. Therefore, the framework is designed as an open, disclosed decision-making mechanism, not an automatic purchasing and payment mechanism or a cancel-or-irreversibility mechanism, as offered in governance guidance in a compliance context (Choudhary & S. Sepuri, 2026).

6. Future Research Recommendations

6.1. Live and Cross-Domain Validation

The framework is not comprehensive and will require A/B testing, not just offline rankings. Cross-domain deployment is especially important in Ecommerce, Entertainment, Education, Tourism, and Professional Services, as user goals, catalog structures, risk levels, and feedback signals differ across domains. Longitudinal experiments should measure clicks, purchases, returns, cancellations, dwell time, satisfaction, complaint rates, and repeat use over a 6-12 month period.

Longitudinal observation would indicate whether there is any enhancement in the course of the season, of the stock, or of advertising and preference drift (Gupta & R. Katarya, 2023). Further research could also investigate the relationship between offline metrics such as NDCG@10 and Recall@10 and conversion metrics, decision time, and subjective utility metrics in live settings. This will help clarify whether moving toward the benchmark serves the user or the business.

6.2. Lifelong Memory and Preference-Drift Management

Research on personalization should eventually cover stable user preferences, time-varying contextual user requirements, situational requirements, and uncertain inferences. Memory consolidation and deletion to prevent obsolete information from appearing in future recommendations should be addressed, incorporating contradiction, confidence revision, and time decay. The user should be able to view, change, export, and delete stored preferences. Continual-learning algorithms should learn in new contexts without catastrophic forgetting and transfer from related tasks when appropriate (Lin, Yang, Fan, & J. Zhang, 2022). With proper precautions and consent, cross-device or cross-domain transfers can be enabled. Evaluations should differentiate between real preference, exploratory behavior, unnecessary behavior caused by false or misleading feedback, and the effectiveness of rollback.

Managing Long-Term Memory and Preference Drift

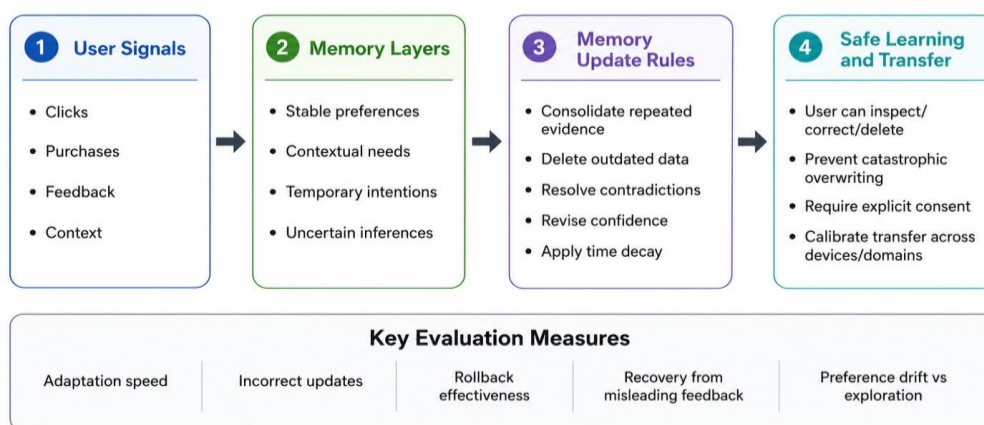


Figure 4: A framework for the management of lifelong user memory, preference drift, safe learning, controlled transfer, and system evaluation.

Figure 4 presents a framework for handling long-term memories and preference drift. The process has four steps: collecting user signals, configuring memory layers, updating memories, and safely learning and controlling transfer. The figure also identifies key evaluation indicators, including adaptation speed, incorrect update frequency, rollback effectiveness, recovery from misleading feedback, and preference drift recovery.

6.3. Safety, Privacy, Fairness, and Governance

Future research should focus on security and privacy, including consent-based memory, data minimization, encryption, access control, retention rules, and auditable preference changes. Failure rates and the quality of activity recommendations should also be compared across relevant sub-populations by activity level, language, devices, and accessibility needs. Robustness test cases should include manipulated product content, misinformation about the product, and malicious manipulation of the processes, malicious reviews, and attempts to draw sensitivity inferences (Lyu, et al., 2026).

Proposals that substantially alter or affect health, employment, credit, education, or large purchases should remain subject to human decision, with appropriate reporting of uncertainty. Users should have mechanisms for explanation, correctness, deletion, objection, and appeal. Governance should ensure people are held accountable for the agent's actions,

limit escalation pathways, ensure agent action is provenance- and audit-aware, and not permit the agent to conduct commercial transactions unless expressly authorized by knowledge (Kaul, Lan, & P. Gupta, 2026); (Khan, Joyce, & M. Habiba, 2025).

6.4. Efficiency, Multimodality, and System Interoperability

Efficiency research should address smaller specialist models, dynamic agent activation, model cascades, compaction in memory summaries, and cost-aware tool use. When the demands are simple, unnecessary reasoning should not be executed; more expensive reasoning should only be executed if the value of the reasoning exceeds the computation cost. Multimodal systems should not treat descriptions as all-encompassing catalog facts but should verify them against descriptions, images, audio, and video. Interoperability, reproducibility, and portability could be achieved through standardized agent-message schemas and tool interfaces (Y. Benajiba et al., 2025). Latency, energy consumption, privacy, availability, and carbon emissions should also be considered for edge-to-cloud deployment. Open benchmarks must be realistic, with realistic tools, evolving catalogs, adversarial conditions, known rewards, full interaction trajectories, and cost accounting, to evaluate accuracy gains in the context of operational impact.

7. Limitations, Ethics, and Governance

7.1. Methodological and External-Validity Limitations

Offline evaluation has certain methodological restrictions. Logged information such as purchases, clicks, and ratings reflects only products that leave a measurable impression on users, so exposure bias can make promotions or product visibility a factor to consider (Liao, Yang, Zhou, Zhang, & J. Wen, 2024). A high-ranking product could thus reflect past visibility and relevance. Another restriction of candidate sampling is that it is easy to compare an interacted item with the 99 sampled items, but much harder in a constantly evolving inventory of millions of products. MovieLens users and Amazon reviewers are also self-selected groups and may not represent the general population in terms of activity level, geographic region, purchase habits, language, or access to technology. Simulated feedback further reduces conversation complexity because predefined feedback cannot fully capture uncertainty, emotions, strategic behavior, evolving purpose, or the ambiguity of acceptance and refusal (Yoon, He, Echterhoff, & J. McAuley, 2024).

Dynamic elements like price fluctuation, stock condition, promotion details, sponsored product ads, fraudulent reviews, new products, and competitor activity cannot be replicated in a specific data set. Additionally, leave-one-out evaluation in the temporal dimension cannot explain why preferences change after recommendations. Results can also be influenced by the language model used, the retrieval index, prompt design, the candidate generator used, and the hardware. Therefore, interpret results as benchmark evidence, not necessarily as evidence of production effectiveness. External validation is linked to live shadow deployment, having different users, changing catalogs, and measures like conversion rate, returns, satisfaction, and complaints.

A further limitation concerns the availability of data needed for the reviewer-requested agreement and adaptation statistics. Although the experiment generated 900 explanation

judgments and 10,000 simulated feedback episodes, the current manuscript materials do not include the item-level evaluator matrix, round-specific ranking outputs, drift-event labels, or memory-update correctness labels required to calculate Krippendorff's alpha, post-feedback changes in NDCG@10, preference-drift recovery rate, or incorrect-memory-update rate. These measures are therefore treated as not reportable rather than as zero or null effects. Consequently, the manuscript makes only cautious claims about evaluator consistency, adaptation strength, and memory-update reliability.

7.2. Privacy, Bias, and Security Risks

Sensitive personal information, such as health, religion, political interests, financial status, sexuality, location, and family status, could be included or disclosed in long-term preference memory. Although people's sensitive attributes are explicitly removed, an agent might still infer them from brands, consumed media, search language, price sensitivity, or repeated behavior. Popularity-biased retrieval can also help to better support highly active users, at the expense of cold-start, low-activity, or underrepresented users (Rahmani, Deldjoo, Tourani, & M. Naghiaei, 2022). A fairness evaluation should therefore consider, for a given scenario, ranking quality, constraint satisfaction, exposure, and failure rates, among other factors, across relevant groups.

Product descriptions, product reviews, images retrieved from Web Pages, or answers generated by a tool can all introduce security threats. An agent might overlook policies, leak memory, or steer recommendations to specific items depending on the prompt injection. Too many permissions may allow unauthorized access, and catalog poisoning can add features to catalogs, add links, or fake ratings. Similarly, explanations may become misleading when it is unclear which ones are highest or lowest, and they are shown as conclusions. In another autonomous-driving scenario, generative reconstruction provided useful reconstructions for 47 out of 47 incidents that conventional reconstruction failed to reconstruct (A. Guo et al., 2024); (A. Ubale, 2023). Although this evidence is outside recommendation, it reinforces the need for evidence review, uncertainty communication, and accountability rather than automatic acceptance of consequential generative outputs.

Table 6: Summary of privacy, bias, security issues, and recommended controls for agentic recommendation systems.

Risk area	Main concern	Examples or indicators	Recommended controls
Privacy	Long-term memory may contain sensitive personal information.	Health, religion, politics, financial status, sexuality, family status, and location.	Explicit consent, data minimization, purpose limitation, encryption, retention controls, and user-controlled deletion.
Sensitive inference	Agents may infer protected characteristics indirectly.	Brands, media consumption, search language, price sensitivity, and repeated behaviour.	Provenance records, confidence thresholds, restricted memory writes, and least-privilege access.
Bias and fairness	Popularity-based retrieval may disadvantage underrepresented users.	Poor coverage for cold-start, minority, and low-activity users.	Compare ranking quality, exposure, constraint satisfaction, and failure rates across groups.
Security	Malicious content may manipulate agents or	Prompt injection, catalogue poisoning, fabricated attributes,	Treat retrieved content as untrusted, scan external inputs, validate

Risk area	Main concern	Examples or indicators	Recommended controls
	tools.	malicious links, coordinated ratings, and excessive permissions.	catalogue claims, and separate retrieved content from system instructions.
Misleading outputs	Explanations may present uncertain rankings as factual conclusions.	Unsupported recommendations or overconfident generative outputs.	Evidence review, uncertainty communication, human oversight, and accountability.
Auditability	Poor documentation may prevent investigation of failures.	Unrecorded access, memory changes, tool calls, ranking evidence, model versions, or fallback events.	Maintain privacy-conscious audit logs while excluding irrelevant personal information.

Table 6 reviews key privacy, bias, security, misleading-output, and auditability concerns about agentic recommender systems. It links these risk areas to common indicators and suggested controls, such as consent, data minimization, fairness testing, human oversight, and privacy-conscious audit logging.

Recommendations include encryption at rest and in transit, data minimization, explicit consent, purpose limitation, configurable retention periods, user deletion, and least-privilege permissions for tools. Require provenance and Confidence thresholds for sensitive memory writes. Treat retrieved content as untrusted input, scan it for malicious instructions, and separate it from system policies (Chen, Piet, Sitawarin, & D. Wagner, 2025). Retro quality control for catalog identifiers and claims should be done against authoritative records. Audit documentation should include access events, memory changes, tool calls, ranking evidence, model versions, and fallbacks, but exclude irrelevant personal details.

7.3. Governance and Human Oversight

Governance should first establish a framework for decision support, not replace human decision-making. Users should know which attributes were considered, how they contributed to the ordering, and what evidence justified the selection (Heuss, Rijke, & A. Anand, 2025); (Polleti, Souza, & F. G. Cozman, 2025). They should also be able to edit their settings, seek explanations of the information and the impact on them, have information deleted, and object. If a domain is used for applications such as health, employment, education, credit, or insurance, the domain should be verified, and uncertainty estimates should be calibrated to allow approved third-party involvement, typically a human.

Agents should not take irrevocable actions, sign contracts, make payments, or make purchases without specific permission for each act. Governance controls include modeling and prompting for versioning, approval processes, rollback, incident categorization, escalation staffing, frequent bias testing, red team testing, and independent review. In production, monitor unsupported claims, invalid tool calls, privacy incidents, subgroup differences, user overrides, and fallback frequency (Patil, et al., 2025). To provide accountability for automated recommendations that do not work, clear ownership of data quality, model behavior, tool integration, complaints, and remediation is required.

8. Conclusions

The study proposes and evaluates a five-agent controlled system which includes user-preference interpretation, candidate retrieval, evidence-based explanation, feedback learning, and session and preference memory. The structured communication, catalog validation, bounded reasoning cycles, and provenance logging, along with confidence thresholds and fallback mechanisms, were achieved using a fixed orchestrator model. While the architecture is a promising way to audit agentic decision support, there is not enough evidence to conclude that the design produces statistically significant ranking-performance results better than the baselines evaluated. The comparative evaluation used 20,000 users, 300,000 chronological interactions, 20,000 ranking tasks, 10,000 feedback episodes, and 900 explanation judgments across MovieLens 25M and Amazon Reviews'23 Electronics. Popularity ranking, SASRec, single-agent LLM-RAG, and the proposed five-agent configuration were tested with the same targets, user histories, candidate pools, and five random seeds. The framework supported a broader evaluation that included NDCG@10, Recall@10, Hit Rate@10, MRR@10, constraint satisfaction, catalog coverage, diversity, novelty, unsupported-item frequency, explanation faithfulness, latency, cost, and fallback behavior. However, no model achieved an overall statistically significant advantage after Holm correction (adjusted $p < .05$), and the available outputs did not support interpretation of rank-biserial effect sizes. The defensible conclusion is therefore that ranking accuracy alone is insufficient to assess agentic recommendation quality.

Component analysis revealed that semantic memory, episodic memory, explicit reasoning, collaborative retrieval, semantic retrieval, explanation, and feedback learning played distinct roles in the architecture. The study did not statistically assess their impact on the margin of error for ranking accuracy, latency, tokens, constraint satisfaction, and recovery. Similarly, the explanation evaluation involved 300 explanations assessed by three evaluators, yielding 900 judgments, while the feedback-adaptation experiment comprised 10,000 simulated feedback episodes. Accordingly, Krippendorff's alpha, post-feedback NDCG@10 changes, preference-drift recovery rate, and incorrect-memory-update rate remain not reportable, and the manuscript makes no definitive claims regarding evaluator agreement or the magnitude of feedback adaptation.

In practice, the framework includes catalog-ID validation, one retry, circuit-breakers, memory updates based on confidence, audit logging, and a fallback to SASRec. However, current analytical outputs do not adequately reflect compliance with the proposed criteria of >95% constraint satisfaction, <1% unsupported items, <2% invalid tool calls, and a 95th percentile response time of <4 seconds. Therefore, these values are acceptance criteria for deployment, not achieved results. The architecture is a human-supervised, transparent decision-support framework. Future testing would include live shadow testing, real feedback, no failure, non-intrusive memory (privacy), longitudinal investigation of improvement using an A/B test, and subgroup fairness analysis. Criteria for cost-sensitive deployment would also confirm that the architecture provides aggregate improvement over time in satisfaction, conversion, retention, reliability, fairness, and accountable decision-making.

References:

- A. Guo et al. (2024). "SoVAR: Building generalizable scenarios from accident reports for autonomous driving testing," . in *Proc. 39th IEEE/ACM Int. Conf. Automated Software Engineering*, pp. 268–280. <https://dl.acm.org/doi/pdf/10.1145/3691620.3695037>.
- A. Prasad. (2025). "Designing a reliable, ultra-low-latency data access environment for real-time applications in modern data centers,". *The USA Journals TAJIIR*, vol. 7, no. 7, pp. 123–136, 2025. <https://usajournal.net/index.php/tujtajiir/article/download/145/144>.
- A. Prasad. (2026). "Accelerating HPC performance: Strategies for overcoming modern challenges with NVMe, InfiniBand, and RDMA,". in *Proc. 2026 Int. Conf. Signal, Systems, and Computing for Next-Gen Automation (ICSSCNA)*, May 2026, pp. 463–471. IEEE. <https://ieeexplore.ieee.org/abstract/document/11546937/>.
- A. Ubale. (2023). "Beyond telematics: Leveraging generative AI for synthetic accident reconstruction and liability attribution in autonomous vehicle claims,". *Int. J. AI, BigData, Comput. Manage. Stud.*, vol. 4, no. 4, pp. 119–124, <https://ijaibdcms.org/index.php/ijaibdcms/article/download/356/352>.
- A. Yehudai et al. (2025). "Survey on evaluation of LLM-based agents," . *arXiv*., 2503.16416. <https://arxiv.org/pdf/2503.16416>.
- Chapman, A., Lauro, L., Missier, P., & R. Torlone. (2024). "Supporting better insights of data science pipelines with fine-grained provenance,". *ACM Trans. Database Syst.*, vol. 49, no. 2, pp. 1–42, 2024. <https://scholar.google.com/scholar?output=instlink&q=inf>.
- Chen, S., Piet, J., Sitawarin, C., & D. Wagner. (2025). "StruQ: Defending against prompt injection with structured queries,". in *Proc. 34th USENIX Security Symp. (USENIX Security 25)*, pp. 2383–2400. <https://www.usenix.org/system/files/usenixsecurity25-chen-sizhe.pdf>.
- Choudhary, S., & S. Sepuri. (2026). "AI-augmented decision governance in compliance-critical enterprise workflow systems: Implications for leadership and decision-making models in mid-sized enterprises,". in *Proc. 4th Int. Conf. Self Sustainable Artificial Intelligence Systems (ICSSAS)*, pp. 1183–1191. <https://ieeexplore.ieee.org/abstract/document/11559573/>.
- Ding, P., & R. Stevens. (2025). "ToolRegistry: A protocol-agnostic tool management library for function-calling LLMs,". *arXiv*., 2507.10593, 2025. <https://arxiv.org/pdf/2507.10593>.
- E. Tabassi. (2023). Artificial Intelligence Risk Management Framework (AI RMF 1.0). *NIST AI 100-1. Gaithersburg, MD, USA: National Institute of Standards and Technology*,.
- Gannavarapu, P., & R. K. R. Samantapudi. (2025). "Secure AI-driven identity infrastructure for regulated sectors,". *Int. J. Intell. Syst. Appl. Eng.*, <https://www.ijisae.org/index.php/IJISAE/article/view/7913>.
- Gondi, S., Sepuri, S., & P. Narayan. (2025). "AI ethics for public financial systems: A cross-sector perspective,". *Int. J. Appl. Math*, vol. 38, pp. 2212–2233.

- Gupta, G., & R. Katarya. (2023). "A novel approach to alleviate data sparsity and generate dynamic fruit recommendations from point-of-sale data,". *Concurrency Comput. Pract. Exp.*, vol. 35, no. 1, Art. no. e7423, <https://onlinelibrary.wiley.com/doi/abs/10.1002/cpe.7423>.
- H. Liao et al. (2026). "Eliminating out-of-domain recommendations in LLM-based recommender systems: A unified view,". in *Findings Assoc. Comput. Linguistics: ACL 2026, Jul. 2026*, pp. 6251–6271. <https://aclanthology.org/2026.findings-acl.310.pdf>.
- Heuss, M., Rijke, M. d., & A. Anand. (2025). "RankingSHAP: Faithful listwise feature attribution explanations for ranking models,". in *Proc. 48th Int. ACM SIGIR Conf. Research and Development in Information Retrieval*, pp. 381–391. <https://dl.acm.org/doi/pdf/10.1145/3726302.3729971>.
- Huang, X., Lian, J., Lei, Y., Yao, J., Lian, D., & X. Xie. (2025). "Recommender AI agent: Integrating large language models for interactive recommendations,". *ACM Trans. Inf. Syst.*, vol. 43, no. 4, pp. 1–33, 2025. <https://dl.acm.org/doi/full/10.1145/3731446>.
- Ibrahim, O. A., Younis, E. M., Mohamed, E. A., & W. N. Ismail. (2025). "Revisiting recommender systems: An investigative survey,". *Neural Comput. Appl.*, vol. 37, no. 4, pp. 2145–2173. <https://link.springer.com/content/pdf/10.1007/s00521-024-10828-5.pdf>.
- J. Zhang et al. (2024). "AgentCF: Collaborative learning with autonomous language agents for recommender systems,". in *Proc. ACM Web Conf.*, <https://openreview.net/pdf?id=knAtuqBXDV>.
- Jannach, D., & H. Abdollahpouri. (2023). "A survey on multi-objective recommender systems,". *Frontiers Big Data*, vol. 6, Art. no. 1157899, 2023. <https://www.frontiersin.org/journals/big-data/articles/10.3389/fdata.2023.1157899/pdf>.
- K. S. Chadha. (2025). "Machine learning-augmented ETL pipelines for fraud-resistant insurance claims processing,". *Int. J. Data Sci. Mach. Learn*, vol. 5, no. 1, pp. 410–436.
- Kaul, A., Lan, Q., & P. Gupta. (2026). "AgentBound: Verifiable behavioral governance for autonomous AI agents,". *arXiv*, 30970, 2026. <https://arxiv.org/pdf/2606.30970>.
- Khan, R., Joyce, D., & M. Habiba. (2025). "AGENTS SAFE: A unified framework for ethical assurance and governance in agentic AI,". *arXiv*, 2512.03180, <https://arxiv.org/pdf/2512.03180>.
- Kim, S., Moon, S., Tabrizi, R., Lee, N., Mahoney, M. W., Keutzer, K., et al. (2024). "An LLM compiler for parallel function calling,". in *Proc. 41st Int. Conf. Machine Learning, Jun. 2024*, <https://openreview.net/pdf?id=uQ2FUoFjnF>.
- Liao, J., Yang, M., Zhou, W., Zhang, H., & J. Wen. (2024). "Modeling item exposure and user satisfaction for debiased recommendation with causal inference,". *Inf. Sci.*, vol. 676, Art. no. 120834, <https://www.sciencedirect.com/science/article/pii/S0020025524007485>.
- Liao, M., & S. S. Sundar, J. B. (2022). "User trust in recommendation systems: A comparison of content-based, collaborative and demographic filtering,". in *Proc. 2022 CHI Conf.*

- Human Factors Comput. Syst.*, Apr. 2022, pp. 1–14.
<https://dl.acm.org/doi/pdf/10.1145/3491102.3501936>.
- Lin, S., Yang, L., Fan, D., & J. Zhang. (2022). “Beyond not-forgetting: Continual learning with backward knowledge transfer,”. in *Advances in Neural Information Processing Systems*, vol. 35, pp. 16165–16177, https://proceedings.neurips.cc/paper_files/paper/2022/file/6728fcf94660c59c938319a6833a6073-Paper-Conference.pdf.
- Lyu, Y., Cao, J., Xiao, Y., Sun, Z., Niu, B., Li, F., et al. (2026). “Privacy risks of intermediate representations: Attribute inference in distributed LLM inference,”. in *Findings Assoc. Comput. Linguistics: ACL 2026*, pp. 23412–23426. <https://aclanthology.org/2026.findings-acl.1172.pdf>.
- M. Ravaut, H. Z. (2024). “Parameter-efficient conversational recommender system as a language processing task,”. in *Proc. 18th Conf. Eur. Chapter Assoc. Comput. Linguistics*, vol. 1: Long Papers, Mar. 2024, pp. 152–165. <https://aclanthology.org/2024.eacl-long.9.pdf>.
- M. T. Tran. (2026). “Advancing retail and service strategies: AI-driven consumer behavior prediction, gamification, and ethical marketing,”. *J. Retailing Consum. Serv.*, vol. 88, Art. no. 104558, 2026. <https://magisterenmarketing.cl/wp-content/uploads/2026/01/Advancing-retail-and-service-strategies-AI-driven-consumer-behavior.pdf>.
- P. Gannavarapu. (2025). “Deploying Azure AD federation with SAML for secure enterprise SaaS integration,”. *Computer Fraud & Security*, <https://computerfraudsecurity.com/index.php/journal/article/view/782>.
- P. K. R. Prakashkumar. ((n.d)). “Secure bank integration framework for Oracle ERP Fusion: Enhancing payment disbursement, auto lockbox, and bank account reconciliation,”. <https://www.eeet.org.uk/index.php/journal/article/view/4082>.
- Pamuk, M., Schumann, M., & R. C. Nickerson. (2024). “What do the regulators mean? A taxonomy of regulatory principles for the use of AI in financial services,”. *Mach. Learn. Knowl. Extr.*, vol. 6, no. 1, pp. 143–155, 2024. <https://www.mdpi.com/2504-4990/6/1/8>.
- Patil, S. G., Mao, H., Yan, F., Ji, C.-J. C., Suresh, V., Stoica, I., et al. (2025). “The Berkeley Function Calling Leaderboard (BFCL): From tool use to agentic evaluation of large language models,”. in *Proc. 42nd Int. Conf. Machine Learning*, <https://openreview.net/pdf?id=2GmDdhBdDk>.
- Peng, Q., Liu, H., Huang, H., Yang, Q., & M. Shao. (2025). “A survey on LLM-powered agents for recommender systems,”. *arXiv*, 2502.10050. <https://aclanthology.org/anthology-files/pdf/findings/2025.findings-emnlp.620.pdf>.
- Polleti, G. P., Souza, D. L., & F. G. Cozman. (2025). “Generating explanations for knowledge-aware conversational recommendation systems,”. *User Model. User-Adapt. Interact.*, vol. 35, no. 3, Art. no. 12, <https://link.springer.com/content/pdf/10.1007/s11257-025-09432-6.pdf>.

- Rahmani, H. A., Deldjoo, Y., Tourani, A., & M. Naghiaei. (2022). "The unfairness of active users and popularity bias in point-of-interest recommendation,". in *Advances in Bias and Fairness in Information Retrieval: Proc. 3rd Int. Workshop BIAS 2022*, pp. 56–68. Cham, Switzerland: Springer. <https://arxiv.org/pdf/2202.13307>.
- Raj, S., Jha, A. V., Anand, M., & S. Saha. (2026). "HARPO: Hierarchical agentic reasoning for user-aligned conversational recommendation,". in *Proc. 64th Annu. Meeting Assoc. Comput. Linguistics*, vol. 1: Long Papers, Jul. 2026, pp. 35580–35599.. <https://aclanthology.org/2026.acl-long.1646.pdf>.
- Rangu, S., Chadha, K. S., & S. Samala. (2025). "AI-powered cloud architecture, agile analytics, and enterprise DevOps-enabled framework for healthcare and insurance ecosystem optimisation,". in *Proc. 10th Int. Conf. Science Technology Engineering and Mathematics (ICONSTEM)*, Chennai, India, pp. 1–7. <https://ieeexplore.ieee.org/abstract/document/11374414/>.
- S. Durgam. (2025). "Peer benchmarking systems for RIA performance evaluation in investment technology,". <https://computerfraudsecurity.com/index.php/journal/article/view/785>.
- S. K. RaoVanama. (2024). "AI-augmented CI/CD pipeline optimization for scalable cloud-native deployment,". *Int. J. Artif. Intell. Data Sci. Mach. Learn.*, vol. 5, no. 4, pp. 175–187, 2024. <https://ijaidsmml.org/index.php/ijaidsmml/article/download/368/338>.
- Sharma, S., Dubey, G. P., & H. K. Shakya. (2024). "Optimizing user satisfaction in movie recommendations using variable learning rates and dynamic neighborhood functions in SOMs,". *Int. J. Exp. Res. Rev.*, vol. 41, pp. 130–145, 2024. https://www.academia.edu/download/117189744/11._Optimizing_User_Satisfaction_in_Movie_Recommendations_Using_Variable_Learning_Rates_and_Dynamic_Neighborhood.pdf.
- Shrimal, A., Kanagaraj, S., Biswas, K., Raghuraman, S., Nediyanath, A., Zhang, Y., et al. (2024). "MARCO: Multi-agent real-time chat orchestration,". in *Proc. 2024 Conf. Empirical Methods Natural Language Process.: Industry Track*, Nov. 2024, pp. 1381–1392. <https://aclanthology.org/2024.emnlp-industry.102.pdf>.
- T. Schick et al. (2023). "Toolformer: Language models can teach themselves to use tools,". in *Advances in Neural Information Processing Systems*, vol. 36, pp. 68539–68551, 2023. https://proceedings.neurips.cc/paper_files/paper/2023/file/d842425e4bf79ba039352da0f65.
- W. Zhang et al. (2026). "PersonaAgent: Bridging memory and action for personalized LLM agents,". in *Findings Assoc. Comput. Linguistics: ACL 2026*, Jul. , pp. 26421–26439. <https://aclanthology.org/2026.findings-acl.1315.pdf>.
- X. Shen et al. (2026). "Agentic recommender system with hierarchical belief-state memory,". *arXiv*, 2605.14401, 2026. <https://arxiv.org/pdf/2605.14401>.
- Y. Benajiba et al. (2025). "Open Agent Specification (Agent Spec) technical report,". *arXiv*:, 2510.04173, <https://ui.adsabs.harvard.edu/abs/2025arXiv251004173B/abstract>.

- Y. Shang et al. (2026). “AgentRecBench: Benchmarking LLM agent-based personalized recommender systems,”. in *Advances in Neural Information Processing Systems*, vol. 38, https://proceedings.neurips.cc/paper_files/paper/2025/file/e2d6f7249add096e26679eade.
- Y. Wang et al. (2026). “From agent traces to trust: A survey of evidence tracing and execution provenance in LLM agents,”. *arXiv*., 2606.04990. <https://arxiv.org/pdf/2606.04990>.
- Yoon, S.-E., He, Z., Echterhoff, J. M., & J. McAuley. (2024). “Evaluating large language models as generative user simulators for conversational recommendation,”. *arXiv*., 2403.09738, <https://arxiv.org/pdf/2403.09738>.
- Z. Wu et al. (2026). “AgentGraph: Trace-to-graph platform for interactive analysis and robustness testing in agentic AI systems,”. in *Proc. AAAI Conf. Artif. Intell.*, vol. 40, no. 48, Mar. 2026, pp. 41721–41723. <https://ojs.aaai.org/index.php/AAAI/article/download/42393/46354>.
- Zhou, H., & H. Y. Chan. (2026). “ORCH: Many analyses, one merge—A deterministic multi-agent orchestrator for discrete-choice reasoning with EMA-guided routing,”. *arXiv*, 2602.01797. <https://arxiv.org/pdf/2602.01797>.
- Zhu, J., Dai, Q., Su, L., Ma, R., Liu, J., & G. Cai. (2022). “BARS: Towards open benchmarking for recommender systems,”. in *Proc. 45th Int. ACM SIGIR Conf. Research and Development in Information Retrieval, Jul. 2022*, pp. 2912–2923. <https://dl.acm.org/doi/pdf/10.1145/3477495.3531723>.