Multi-Agentic Recommender Systems: Foundations, Perspectives, and Lessons from Large Scale Deployments





Reza Yousefi Maragheh[1] Yashar Deldjoo[2] Benjamin Coleman [3] Jason Cho[4] Chi Wang[5]

[1] Walmart Global Tech, USA
[2] Polytechnic University of Bari, Italy
[3] Google DeepMind, USA
[3] Pretask.AI, USA
[5] AG2AI, USA

The 49th International ACM SIGIR Conference on Research and Development in Information Retrieval , (SIGIR'26)

Melbourne | Naarm, Australia, 20-24 July 2026

Multi-Agentic Recommender Systems: Foundations, Perspectives, and Lessons from Large Scale Deployments

Yashar Deldjoo[1]

The 49th International ACM SIGIR Conference on Research and Development in Information Retrieval , (SIGIR'26)

[1] Polytechnic University of Bari, Italy

Melbourne | Naarm, Australia, 20-24 July 2026

Outline

- Part I: Introduction, Foundation (Yashar)
- Part II: Industrial Use- Case from Agentic RecSys (Reza)
- Part III: Transitioning to Autonomos Discovery Loops and Practical Hurdles (Jason)
  Break
- Part IV: Self-Evolving Memory and Learned Experiences (Ben)
- Part III: AgentOS AG2 and MassGen (Chi)

Agentic Recommender Systems (Agentic-RecSys)

GPTs and Assistants are precursors to agents. They will gradually be able to plan and perform more complex actions on your behalf. These are our first step toward AI Agents

Check Out Our Recent Monographs on **Agentic RecSys**!

arXiv > cs > arXiv:2507.02097

Search...

Help

Computer Science > Information Retrieval

[Submitted on 2 Jul 2025 (v1), last revised 10 Jul 2025 (this version, v2)]

The Future is Agentic: Definitions, Perspectives, and Open Challenges of Multi-Agent Recommender Systems

Reza Yousefi Maragheh, Yashar Deldjoo

Large language models (LLMs) are rapidly evolving from passive engines of text generation into agentic entities that can plan, remember, invoke external tools, and co-operate with one another. This perspective paper investigates how such LLM agents (and societies thereof) can transform the design space of recommender systems.
We introduce a unified formalism that (i) models an individual agent as a tuple comprising its language core, tool set, and hierarchical memory, and (ii) captures a multi-agent recommender as a triple of agents, shared environment, and communication protocol. Within this framework, we present four end-to-end use cases-interactive party planning, synthetic user-simulation for offline evaluation, multi-modal furniture recommendation, and brand-aligned explanation generation-each illustrating a distinct capability unlocked by agentic orchestration.
We then surface five cross-cutting challenge families: protocol complexity, scalability, hallucination and error propagation, emergent misalignment (including covert collusion), and brand compliance.
For each, we formalize the problem, review nascent mitigation strategies, and outline open research questions. The result is both a blueprint and an agenda: a blueprint that shows how memory-augmented, tool-using LLM agents can be composed into robust recommendation pipelines, and an agenda inviting the RecSys community to develop benchmarks, theoretical guarantees, and governance tools that keep pace with this new degree of autonomy. By

Maragheh, Reza Yousefi, and Yashar Deldjoo. "The future is agentic: Definitions, perspectives, and open challenges of multi-agent recommender systems." *arXiv preprint arXiv:2507.02097* (2025). https://arxiv.org/abs/2507.02097

04433v1 [cs.IR] 5 Jul 2026

Autonomous Information Seeking: A Roadmap for Agentic Recommender Systems

Xinyu Lin[1] Yashar Deldjoo[2✉] Sunhao Dai[3] Honghui Bao[1] Xiaopeng Ye[3] Fatemeh Nazary[2] Wenjie Wang[4] Tommaso Di Noia[2] Jun Xu[3] Tat-Seng Chua[1]
[1]National University of Singapore [2]Polytechnic University of Bari
[3]Renmin University of China [4]University of Science and Technology of China
xylin1028@gmail.com, deldjooy@acm.org

The rapid integration of large language model-based agents into recommender systems has driven a shift from static, ranking-based pipelines toward autonomous and interactive systems that can reason, plan, and act. This survey provides a comprehensive overview of this emerging landscape by introducing a unified taxonomy grounded in the level of autonomy and three core paradigms of agentic recommender systems: agent-assisted recommendation, agent-as-recommender, and agent-as-user-simulator. The autonomy framework organizes existing methods along increasing capabilities in proactivity, context awareness, interaction flexibility, and adaptivity. Building on this framework, the survey analyzes how each paradigm adopts different agentic architectures and how agents enhance key components such as profiles, memory, tool use, workflows, and optimization mechanisms. We further examine evaluation methodologies for agentic recommendation, covering automated metrics, LLM-based judging, and simulation-based assessment, and discuss their limitations in capturing reasoning quality, user experience, and system behavior. Beyond existing evaluation protocols, we further discuss unresolved issues in evaluating agentic recommender systems, including trajectory-level assessment, agent contribution analysis, and calibration of user simulation. Lastly, the survey outlines open challenges in lifelong user modeling, contextual abstraction, multimodal alignment, controllability, trustworthiness, privacy, scalability, and efficiency. Together, these analyses establish a unified foundation for understanding the current progress of agentic

Lin, Xinyu, **Yashar Deldjoo**, Sunhao Dai, Honghui Bao, Xiaopeng Ye, Fatemeh Nazary, Wenjie Wang, Tommaso Di Noia, Jun Xu, and Tat-Seng Chua. "Autonomous Information Seeking: A Roadmap for Agentic Recommender Systems." *arXiv preprint arXiv:2607.04433* (2026). https://arxiv.org/abs/2607.04433

Check out our ACM TORS 2025 Perspective paper!

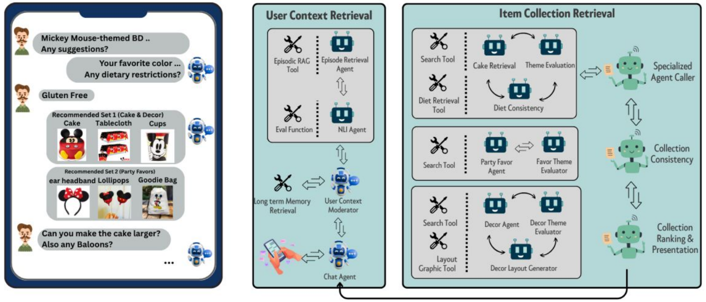
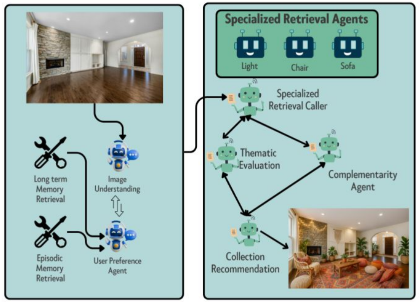


Table 1. Core Capabilities of Agentic Recommender Systems

Icon	Capability	Description
	Planning & Task Decomposition	Breaks down complex goals into sub-tasks, enabling multi-step reasoning and planning (e.g., candidate selection, bundle retrieval, re-ranking) [12, 44, 79].
	Tool Use & Action Execution	Invokes external tools/APIs or performs real-world actions (e.g., database queries, web searches, retrieving live inventory) [76].
	Memory & State Management	Maintains state across steps using memory modules (semantic, vector, knowledge graph), enabling context retention and personalized, multi-turn recommendations [4, 31, 34, 49, 53, 72, 74].
	Autonomy & Goal-Driven Behavior	Operates autonomously in a closed-loop, observing the environment, self-refining, and continuing until the objective is reached [42, 64].



arXiv > cs > arXiv:2507.02097

Search...

Help | Adv

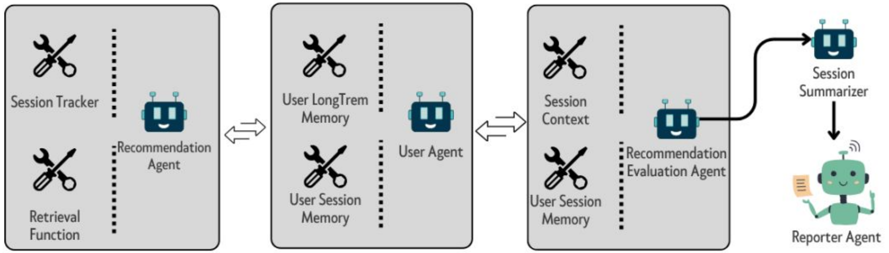
Computer Science > Information Retrieval

[Submitted on 2 Jul 2025 (v1), last revised 10 Jul 2025 (this version, v2)]

The Future is Agentic: Definitions, Perspectives, and Open Challenges of Multi-Agent Recommender Systems

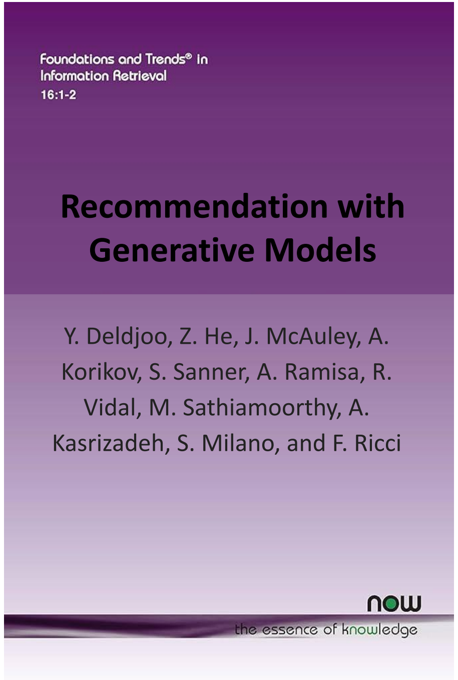
Reza Yousefi Maragheh, Yashar Deldjoo

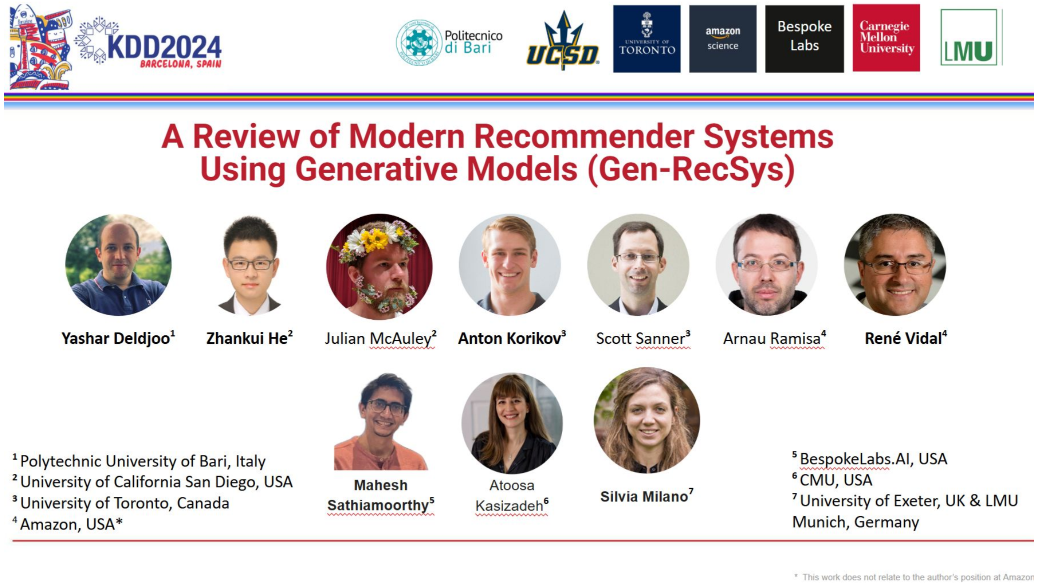
Large language models (LLMs) are rapidly evolving from passive engines of text generation into agentic entities that can plan, remember, invoke external tools, and co-operate with one another. This perspective paper investigates how such LLM agents (and societies thereof) can transform the design space of recommender systems.
We introduce a unified formalism that (i) models an individual agent as a tuple comprising its language core, tool set, and hierarchical memory, and (ii) captures a multi-agent recommender as a triple of agents, shared environment, and communication protocol. Within this framework, we present four end-to-end use cases-interactive party planning, synthetic user-simulation for offline evaluation, multi-modal furniture recommendation, and brand-aligned explanation generation-each illustrating a distinct capability unlocked by agentic orchestration.
We then surface five cross-cutting challenge families: protocol complexity, scalability, hallucination and error propagation, emergent misalignment (including covert collusion), and brand compliance.
For each, we formalize the problem, review nascent mitigation strategies, and outline open research questions. The result is both a blueprint and an agenda: a blueprint that shows how memory-augmented, tool-using LLM agents can be composed into robust recommendation pipelines, and an agenda inviting the RecSys community to develop benchmarks, theoretical guarantees, and governance tools that keep pace with this new degree of autonomy. By



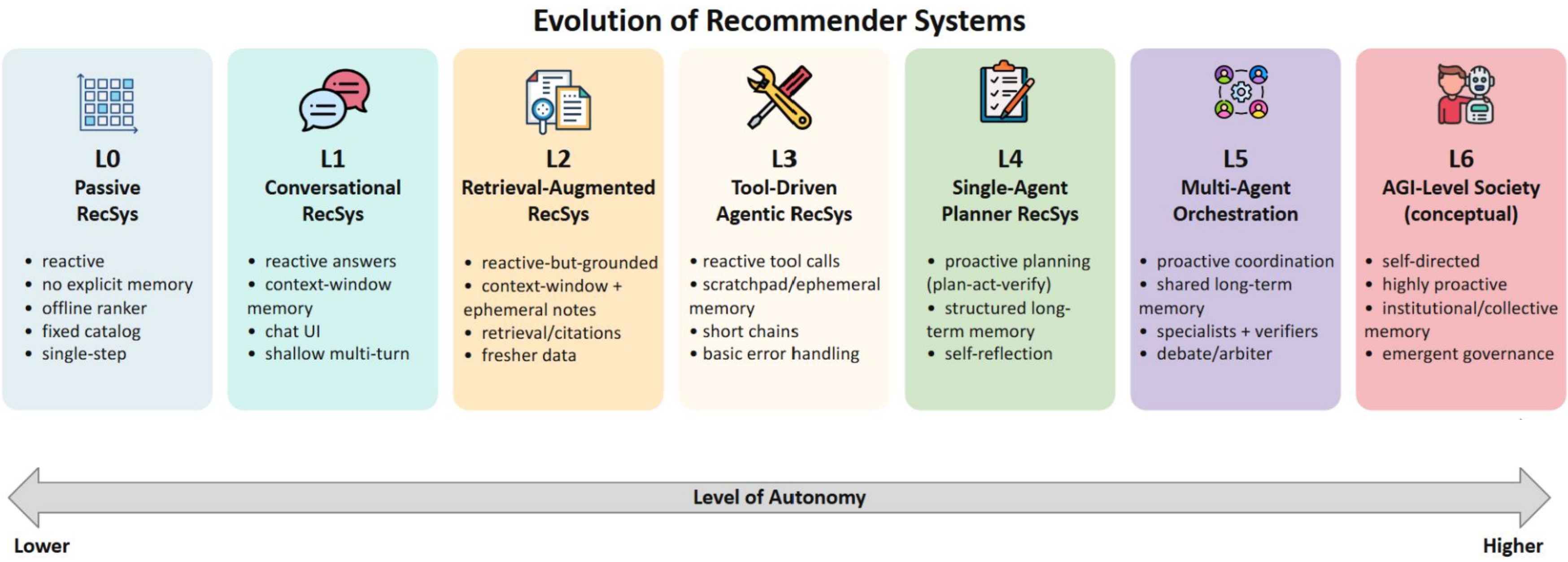
Link: https://arxiv.org/abs/2507.02097

Also Take a look at our Gen-RecSys Work

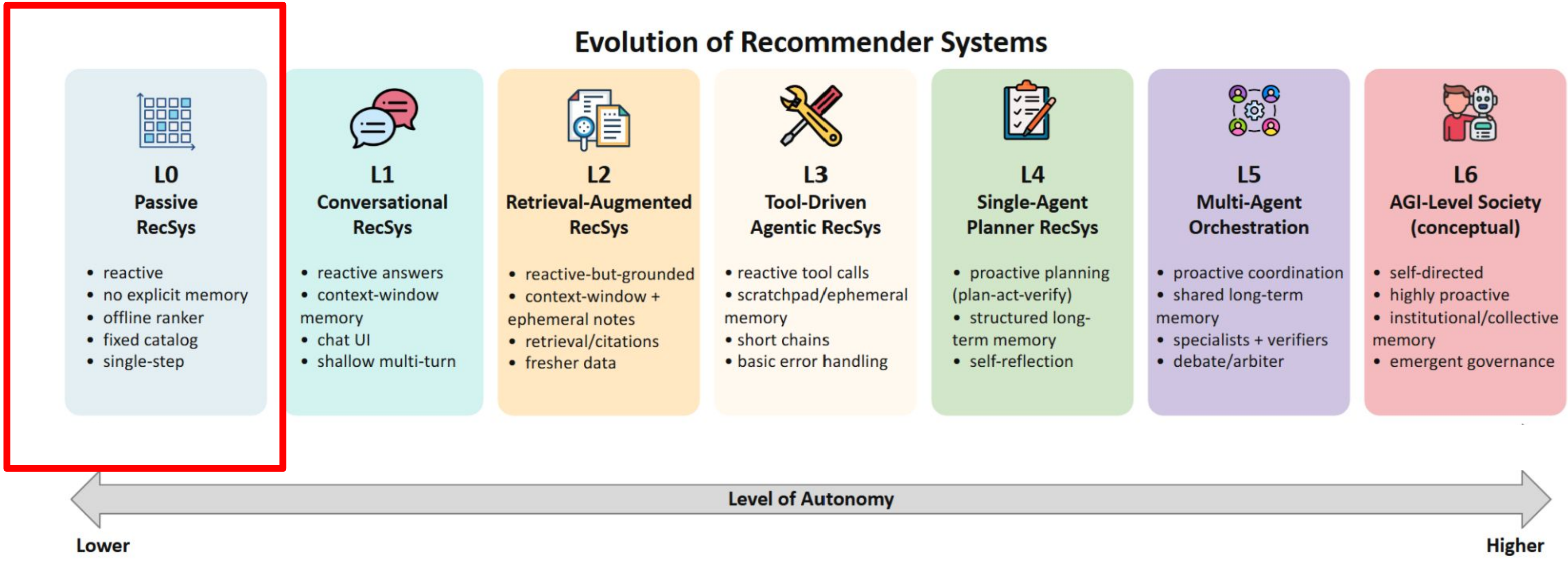




Link: https://arxiv.org/abs/2409.15173

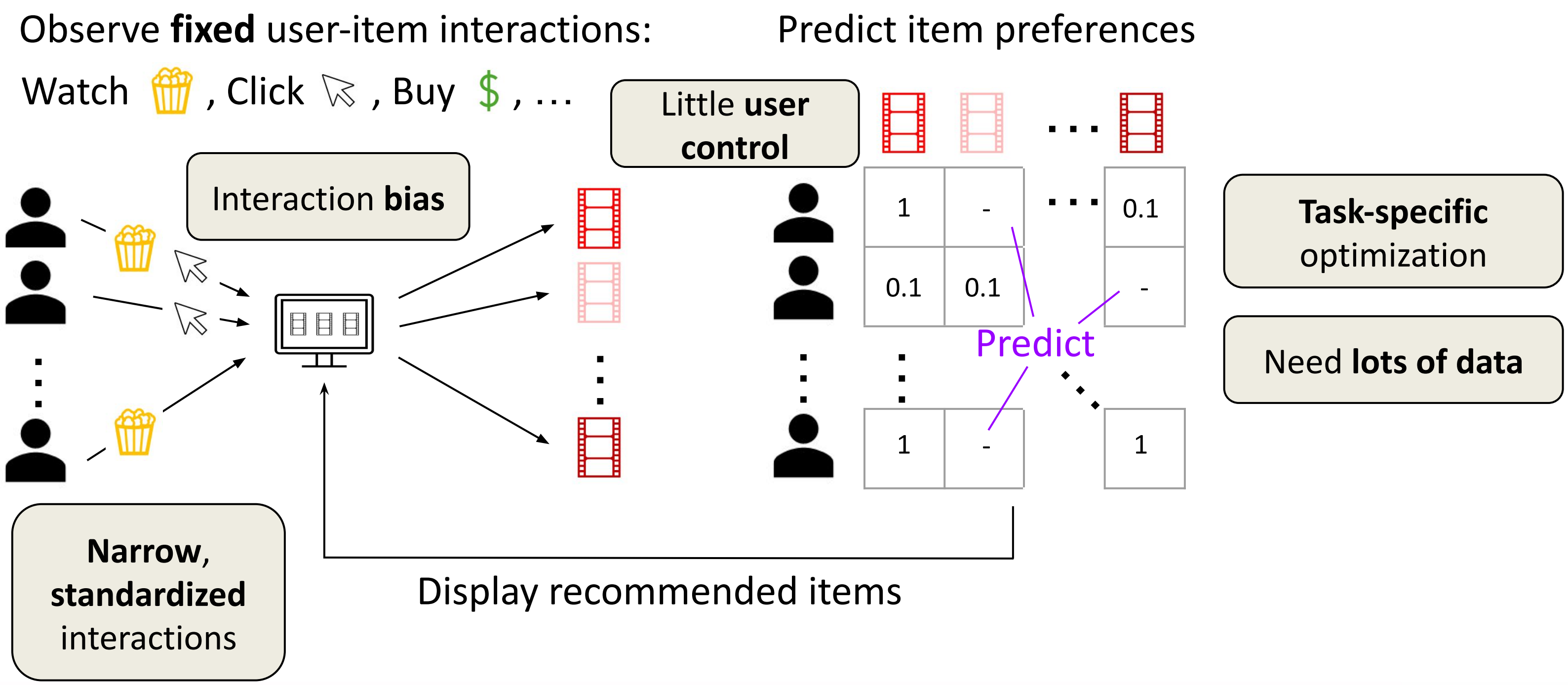


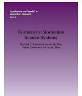
Link: https://arxiv.org/abs/2409.15173

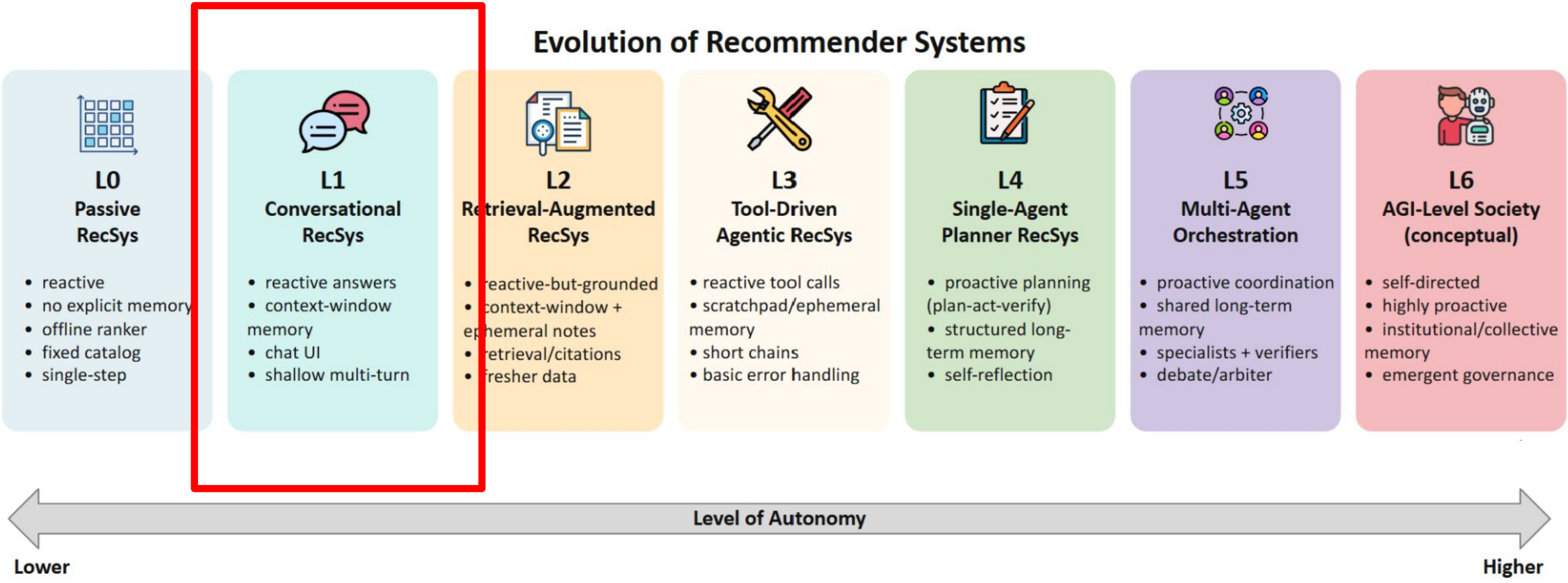


Link: https://arxiv.org/abs/2409.15173

Traditional Recommendation (Recap)





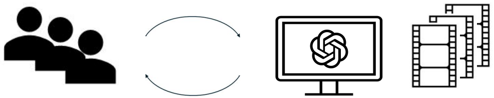


Link: https://arxiv.org/abs/2409.15173

LLMs Unlock NL in Recommender Systems

LLMs **unlock NL** as a medium to represent:

User preferences

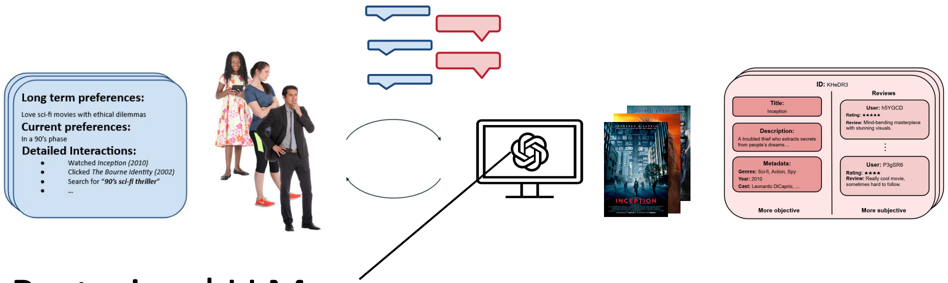


Item descriptions

User-system interactions

Opportunities of LLM-Driven Recommendation

Rich NL data + LLM **general reasoning** abilities =



Pretrained LLMs:

- **Internalized knowledge** about many items and human preferences
- Can reason about **diverse tasks** with **little** or **no new training data**

Opportunities:

- Nuanced **personalization** in **diverse contexts**
- **Interactive, real-time** recommendation
- Faster system design and deployment

NL-based Models

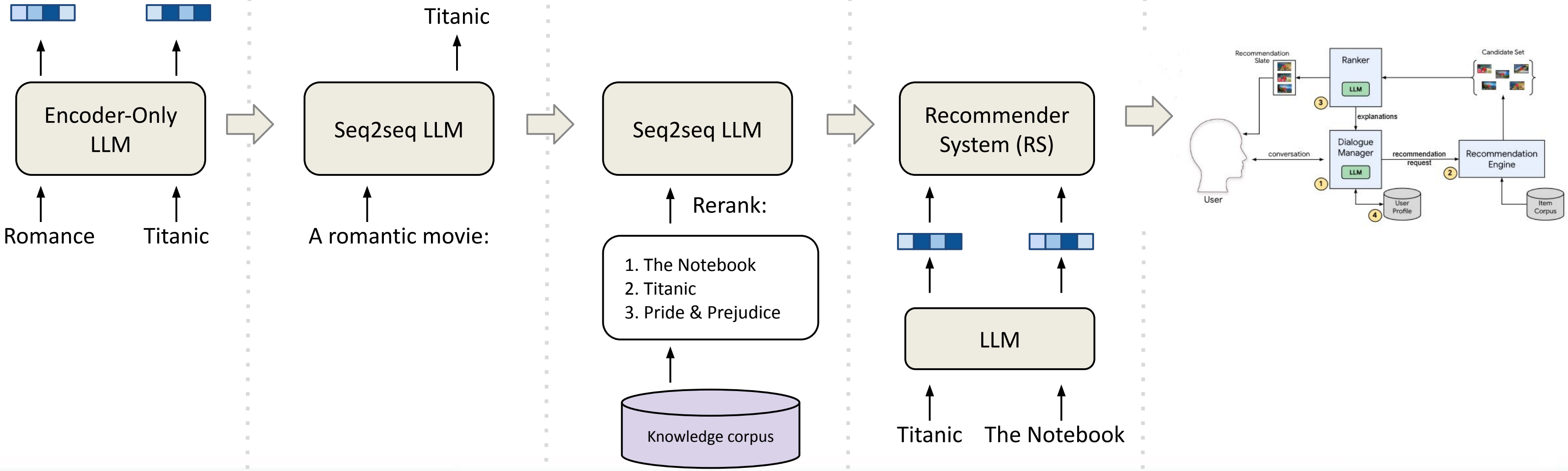
Encoder-Only LLMs

Sequence to Sequence (Seq2seq) LLMs

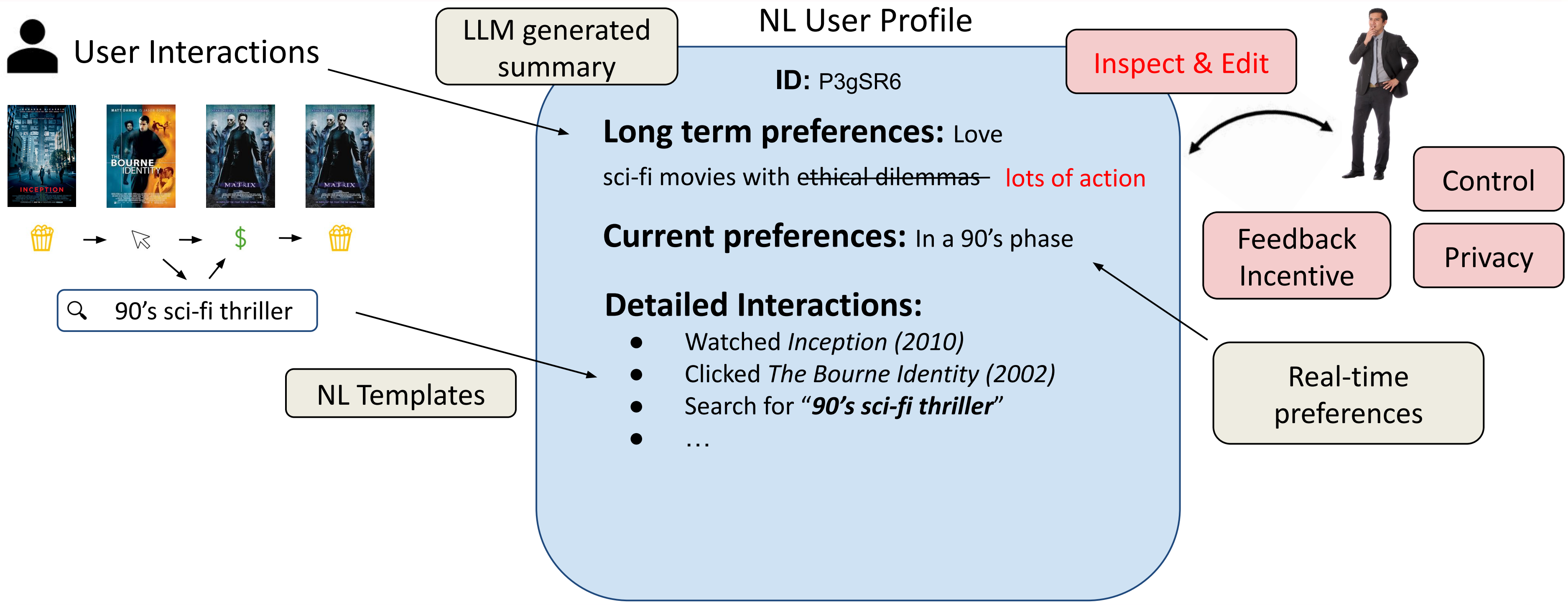
Retrieval-Augmented Generation (RAG)

LLM Representation Generation

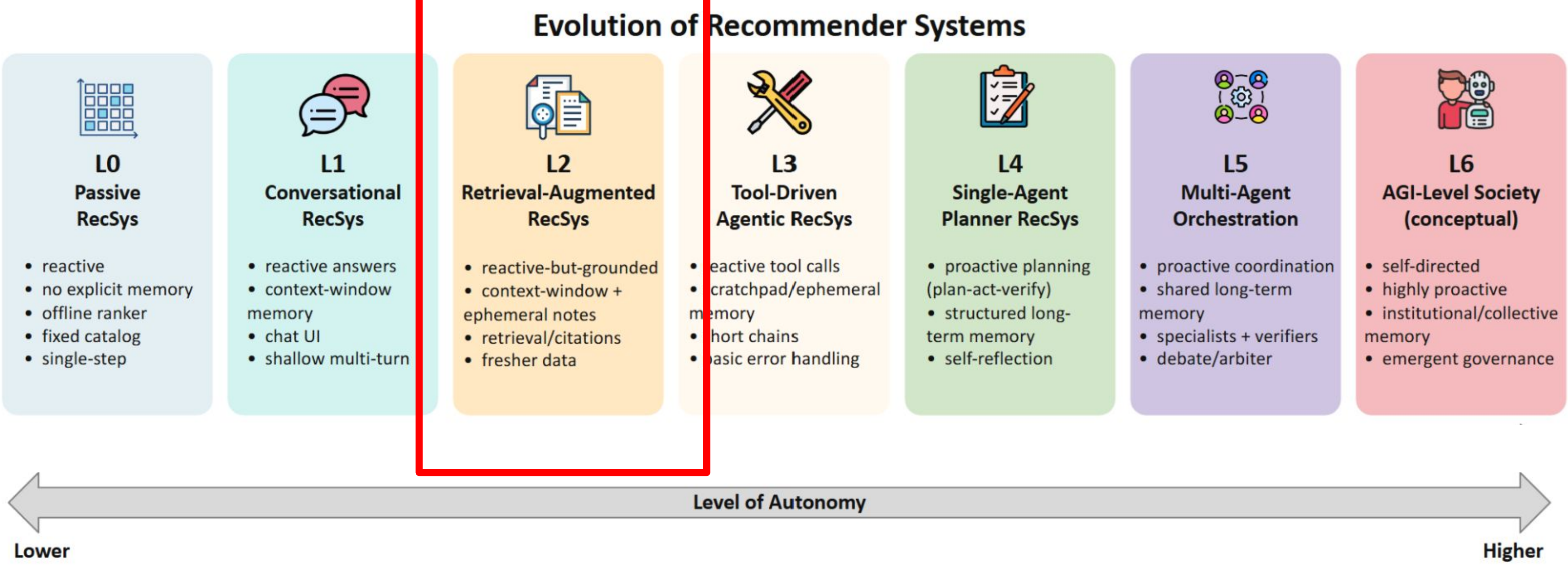
Conversational Recommendation



Editable NL User Profiles



[2] F. Radlinski, K. Balog, F.Diaz, L.Dixon, and B. Wedin. 2022. On Natural Language User Profiles for Transparent and Scrutinable Recommendation. SIGIR 2022. https://doi.org/10.1145/3477495.3531873



Link: https://arxiv.org/abs/2409.15173

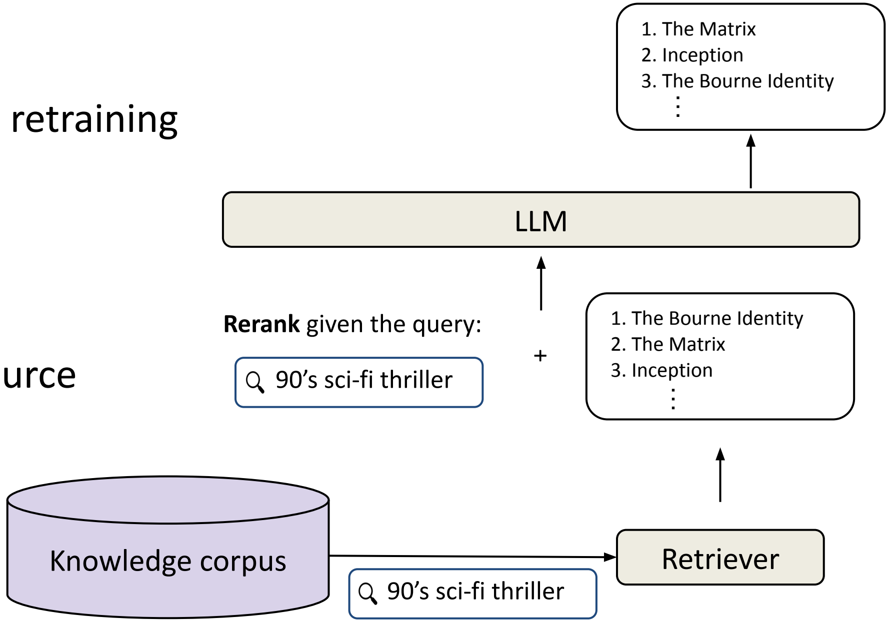
Retrieval Augmented Generation (RAG)

Limitations of LLM internal knowledge:

- Cannot add or revise knowledge without retraining
- More knowledge → more parameters
- No source attribution, hallucination

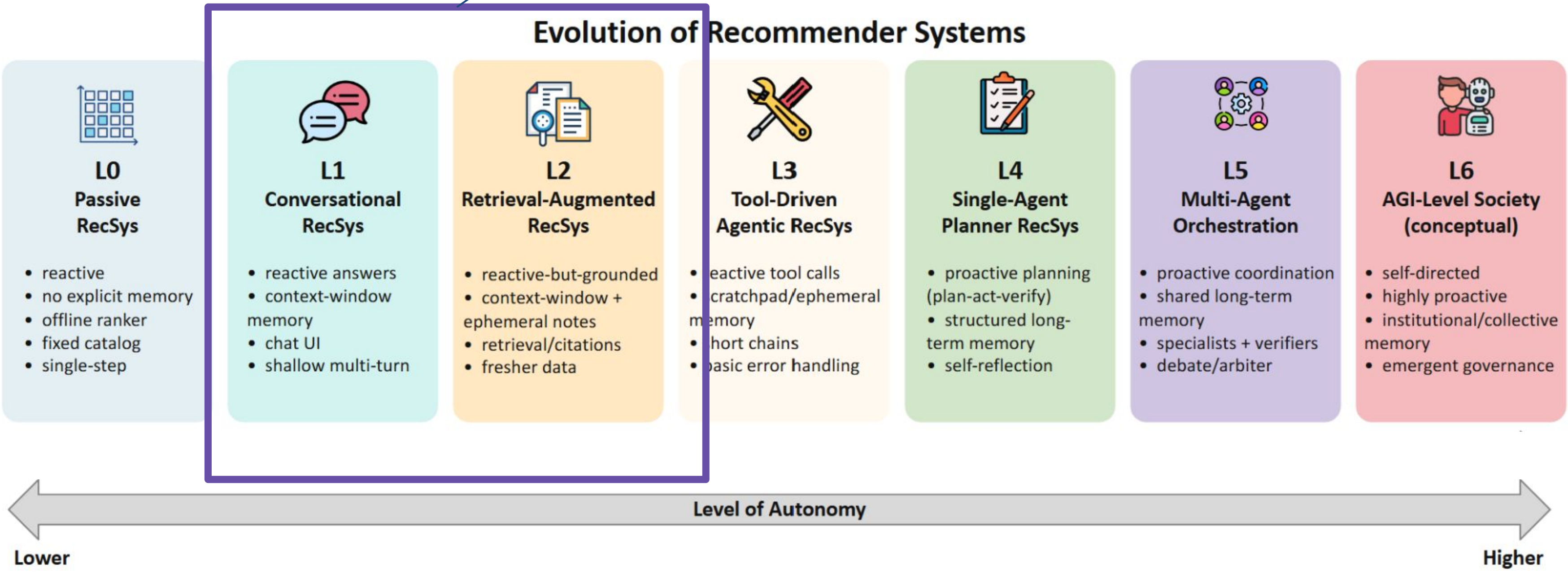
RAG:

- Retrieve information from an external source
- Prompt LLM with retrieved information
- Advantages:
  - Smaller LLM
  - Opportunity for source attribution



Example: RAG for recommendation reranking

Gen-RecSys Zone



Link: https://arxiv.org/abs/2409.15173

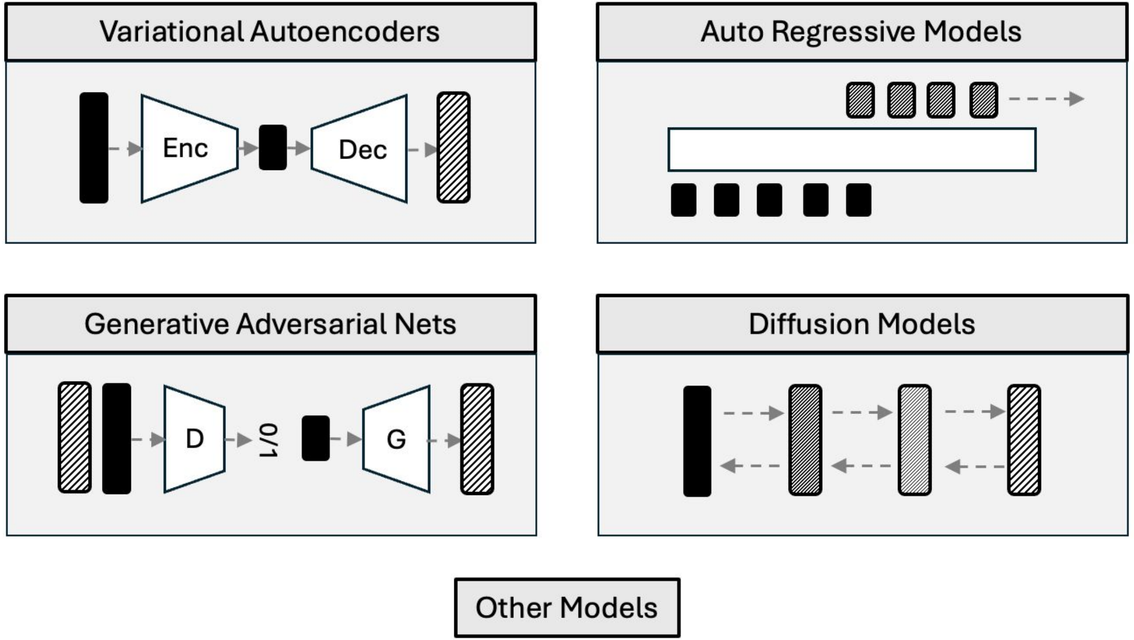
Broad Categories of Gen-RecSys

Gen-RecSys based on modality

- ID-based models → Deep Generative Models
- NLP-based model→Large Language Models
- Multimodal (Foundation) Models

Deep Generative Models (DGMs)

- DGMs
  - VAEs
  - AR Models
  - GANs
  - Diffusion Models
  - Others



* figure adopted from https://lilianweng.github.io/posts/2021-07-11-diffusion-models/

Deep Generative Models (DGMs)

- DGMs

Unconditional: $p_{\text{DGM}}(\mathbf{x}) \approx p(\mathbf{x})$

Conditional: $p_{\text{DGM}}(\mathbf{x}|C) \approx p(\mathbf{x}|C)$

- Attention!

1. **Different** $\underbrace{\boxed{\mathbf{X}}}_{\text{high-dimensional}}$ in *RecSys w/ UI Data*
2. **Different** reasons to model $\boxed{p(\mathbf{x}|\cdot)}$ in *RecSys w/ UI Data*

* For simplicity, we will omit the user identifier *u* in condition.

Deep Generative Models (DGMs)

- **Different** reasons ?

Direct

- Generate **an interaction vector** as recommendations
- Generate **item lists** or **pages** as recommendations
- ……

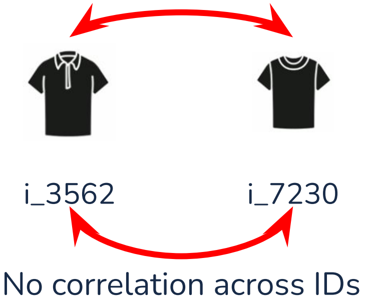
Indirect

- Augment training data to train better **scoring function** $f_\theta(u, i)$
- Better sequential encoding for **next-item prediction** $p_\theta(\mathbf{x}_t | \mathbf{x}_{<t}, \cdot)$
- ……

Item IDs vs Semantic IDs

Item IDs:

- Contain **no prior knowledge** about items;
- **#Parameters** grows proportionally to #items

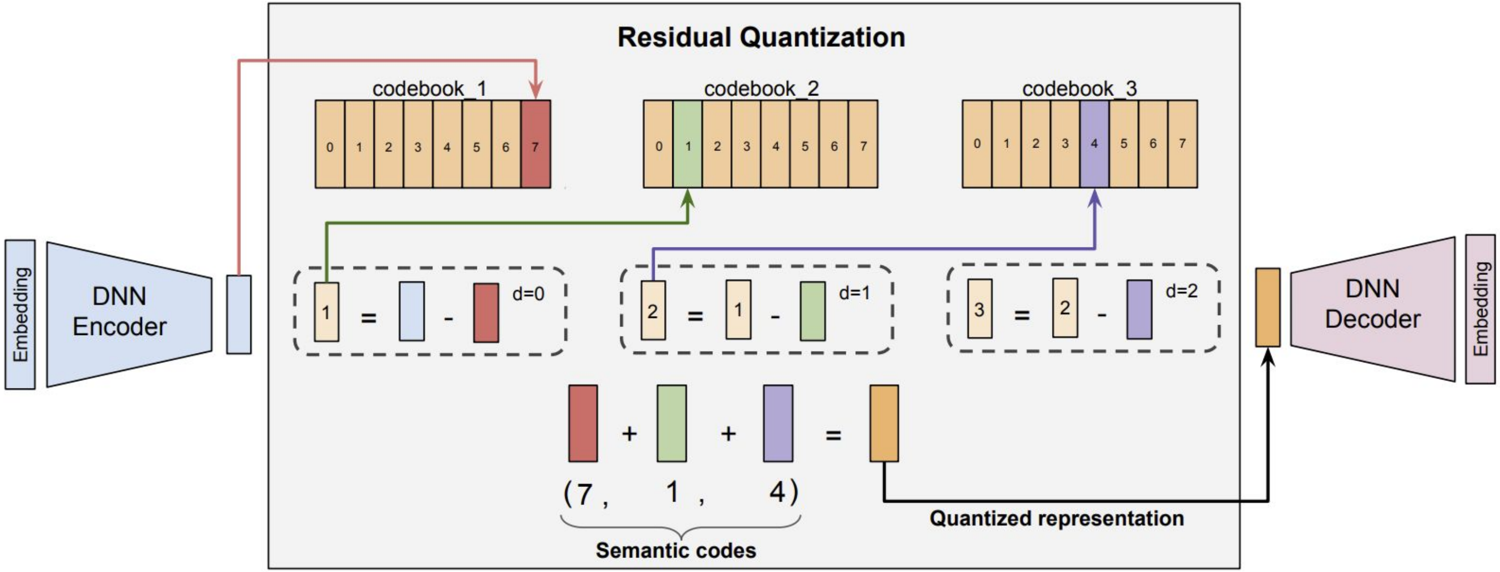


Semantic IDs:

- **Prior knowledge**: semantically similar items have similar tokens;
- **#Parameters**: proportional to #tokens from a compact vocabulary;

Lately: Generative Recommendation

- **E.g. TIGER** (from youtube):



What are Generative Recommender Systems? (Gen-RecSys)

Recommender systems enhanced by Generative AI

Structured List
generate lists, sequences or bundles tailored to user

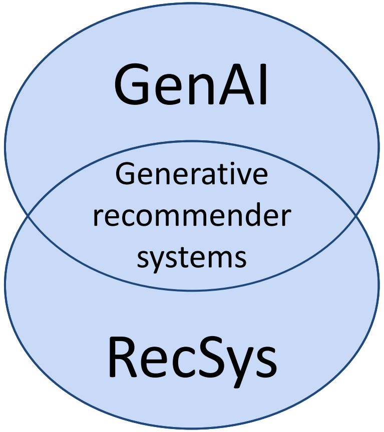
Conversational & textual
Generate natural language responses, recommendations and explanations

Visual & multimodal
Generate images or other media to enrich the recommendation experience

Recommender Systems with Generative Models (Gen-RecSys)

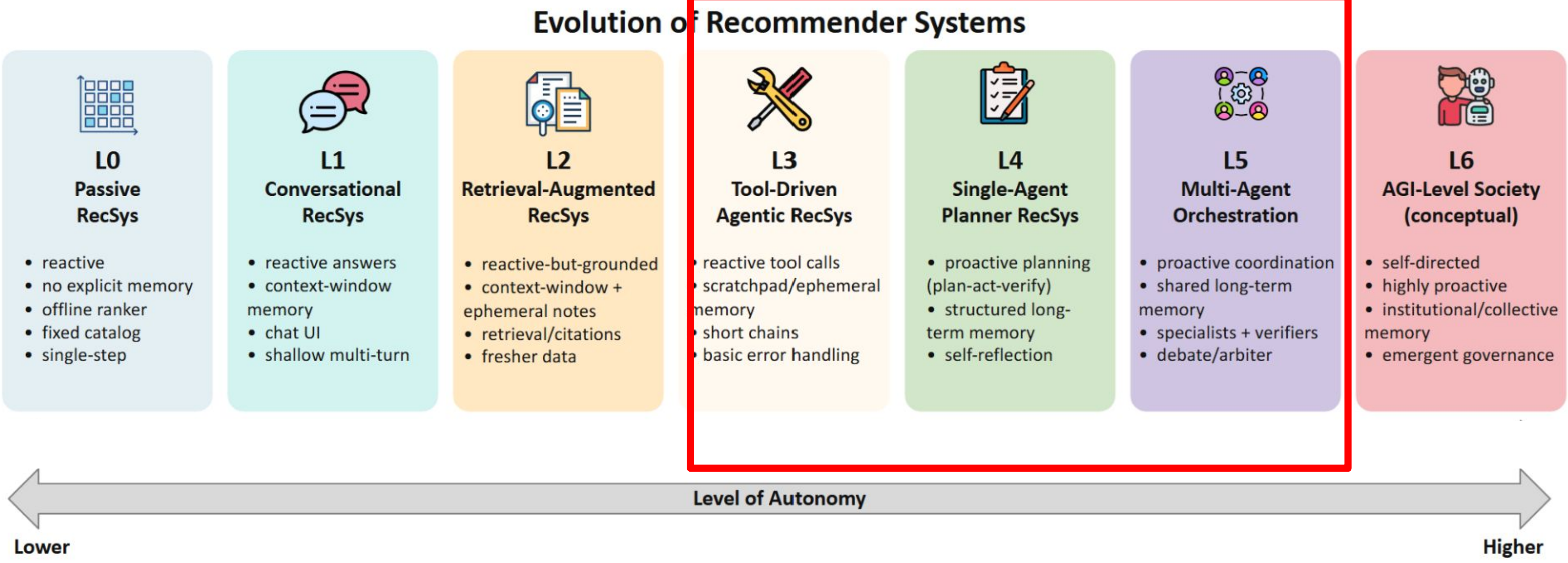
What are generative recommender systems?

- Systems that generate **structured outputs**, like bundles, lists, sets, sequences, etc.
- Systems that generate **text,** including conversational models, abstractive explanations, etc.
- Systems that generate **images,** including virtual try-on, fashion designs, image generation
- Existing generative models that involve **personalization,** e.g. personalized LMs or diffusion models



Essentially: any model that combines ideas from **Generative AI** and **recommender systems**

Agentic-RecSys Zone



Link: https://arxiv.org/abs/2409.15173

Dimension: Personalization

Classical	Generative/LLM-enhanced	Agentic
Pattern matching	**Semantic preference modelling**	**Goal and constraint alignment**
Personalizes from observed interactions and similarity signals.	Uses language to infer richer preference descriptions and item meaning.	Tracks goals, constraints, and trade offs across the task.

Progression in system capabilities

Dimension: Personalization

Classical	Generative/LLM-enhanced	Agentic
Pattern matching	**Semantic preference modelling**	**Goal and constraint alignment**
Personalizes from observed interactions and similarity signals.	Uses language to infer richer preference descriptions and item meaning.	Tracks goals, constraints, and trade offs across the task.

Progression in system capabilities

Personalization becomes agentic when decisions align with the user's goal, constraints, and trade-offs - not only predicted preference.

Dimension: Proactivity

Classical	Generative/LLM-enhanced	Agentic
Triggered Rec.	**Responsive dialogue**	**Goal-directed initiative**
Produces recommendations when the system is queried or refreshed	Answers explicit prompts and can explain or reformulate suggestions	Initiates clarification retrieval, verification, or action when uncertainty or task progress warrants it.

Progression in system capabilities

Dimension: Proactivity

Classical	Generative/LLM-enhanced	Agentic
Triggered Rec.	**Responsive dialogue**	**Goal-directed initiative**
Produces recommendations when the system is queried or refreshed	Answers explicit prompts and can explain or reformulate suggestions	Initiates clarification retrieval, verification, or action when uncertainty or task progress warrants it.

Progression in system capabilities

Proactivity becomes agentic when the system initiates timely clarification or action that advances the user's goal.

Dimension: Context Awareness

Classical	Generative/LLM-enhanced	Agentic
Historical context	**Conversational context**	**Maintained state + environment**
Uses past clicks, ratings, purchases, and sessions.	Uses explicitly provided text and conversation context.	Combines user state, task context, tools, and feedback.

Evolution of Recommendation Models

Dimension: Context Awareness

Classical	Generative/LLM-enhanced	Agentic
Historical context	**Conversational context**	**Maintained state + environment**
Uses past clicks, ratings, purchases, and sessions.	Uses explicitly provided text and conversation context.	Combines user state, task context, tools, and feedback.

Progression in system capabilities

Context becomes agentic when it is maintained as state and updated from the user, environment, tools, and feedback.

Dimension: Task Scope

Classical	Generative/LLM-enhanced	Agentic
Single-step ranking	**Generated rec./answer**	**Multi-step workflow**
Optimizes an item list for a narrow recommendation objective	Produces a list, bundle, explanation, or conversational response.	Plans, checks constraints, uses tools, and updates the answer.

Progression in System Capabilities

Dimension: Task Scope

Classical

Single-step ranking

Optimizes an item list for a narrow recommendation objective

Generative/LLM-enhanced

Generated rec./answer

Produces a list, bundle, explanation, or conversational response.

Agentic

Multi-step workflow

Plans, checks constraints, uses tools, and updates the answer.

Progression in System Capabilities

Task scope becomes agentic when recommendation shifts from producing an output to completing a goal through a multi-step workflow.

Dimension: Adaptivity

Classical	Generative/LLM-enhanced	Agentic
Offline updates	**In-session adaptation**	**Closed-loop strategy updates**
Learns from logs through periodic retraining or refresh cycles.	Adapts within the prompt or current dialogue, often without weight updates.	Updates memory, profile, plan, or tool strategy from new evidence.

Progression in System Capabilities

Dimension: Adaptivity

Classical	Generative/LLM-enhanced	Agentic
Offline updates	**In-session adaptation**	**Closed-loop strategy updates**
Learns from logs through periodic retraining or refresh cycles.	Adapts within the prompt or current dialogue, often without weight updates.	Updates memory, profile, plan, or tool strategy from new evidence.

Progression in System Capabilities

Adaptivity becomes agentic when feedback updates state, memory, plans, or tool strategy during the task - not necessarily model weights.

Dimension: Memory

Classical	Generative/LLM-enhanced	Agentic
Implicit history	**Context window (transient)**	**Persistent managed memory**
User history appears as features, embeddings, or logs.	Short-term dialogue and prompt context are available to the LLM	Working, episodic, and semantic memory can persist across steps

Progression in System Capabilities

Dimension: Memory

Classical	Generative/LLM-enhanced	Agentic
Implicit history	**Context window (transient)**	**Persistent managed memory**
User history appears as features, embeddings, or logs.	Short-term dialogue and prompt context are available to the LLM	Working, episodic, and semantic memory can persist across steps

Progression in System Capabilities

Memory becomes agentic when it is persistent, selectively retrieved, updated, and used to guide future decisions.

Dimension: Tools and Knowledge

Classical	Generative/LLM-enhanced	Agentic
Catalog + model (fixed resources)	**Knowledge + retrieval**	**Dynamic tool orchestration**
Uses item catalogs, logs, indexes, and trained model parameters.	Uses pretrained knowledge and may retrieve external evidence.	Selects rankers, search, filters, APIs, or verifiers during reasoning.

Progression in System Capabilities

Dimension: Tools and Knowledge

Classical

Catalog + model (fixed resources)

Uses item catalogs, logs, indexes, and trained model parameters.

Generative/LLM-enhanced

Knowledge + retrieval

Uses pretrained knowledge and may retrieve external evidence.

Agentic

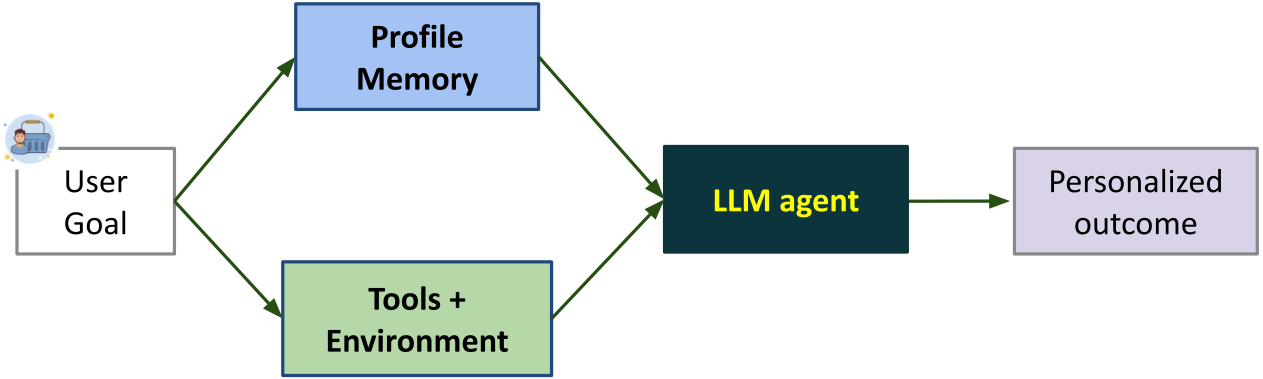
Dynamic tool orchestration

Selects rankers, search, filters, APIs, or verifiers during reasoning.

Progression in System Capabilities

Tool use becomes agentic when the system selects, sequences, and verifies tools within a goal-directed workflow.

What are Agentic Recommender Systems?

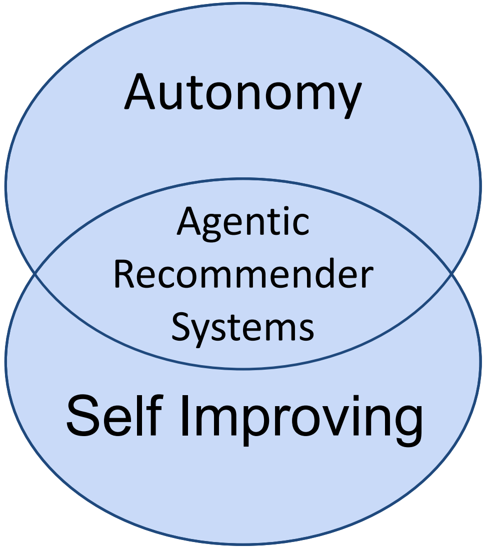


One or more **LLM agents** that plan and act using *tools* and *memory* to produce **personalized outcomes.**

Key Characteristics of Agentic-RecSys

- **Autonomy.** Agents decide actions and adapt based on state and feedback
- **Self-Improvement.** Agents learn from experience, refine models and,

Agentic RS Combine autonomy and self-improvement to proactively gather context, plan, act and reflect.



Agentic Recommender Systems (Agentic-RecSys)

single-turn ranking → **Autonomous goal-driven** assistance

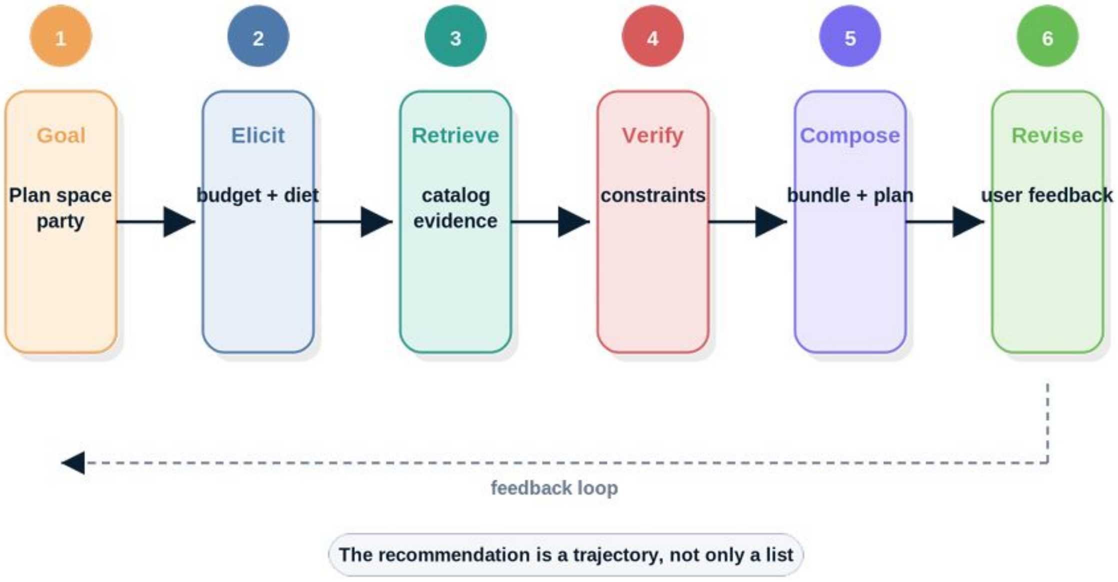
Agentic Recommender Systems (Agentic-RecSys)

single-turn ranking → **Autonomous goal-driven** assistance

“show items” → “solve tasks”

(bundle creation, constraint satisfaction, negotiation, explanation).

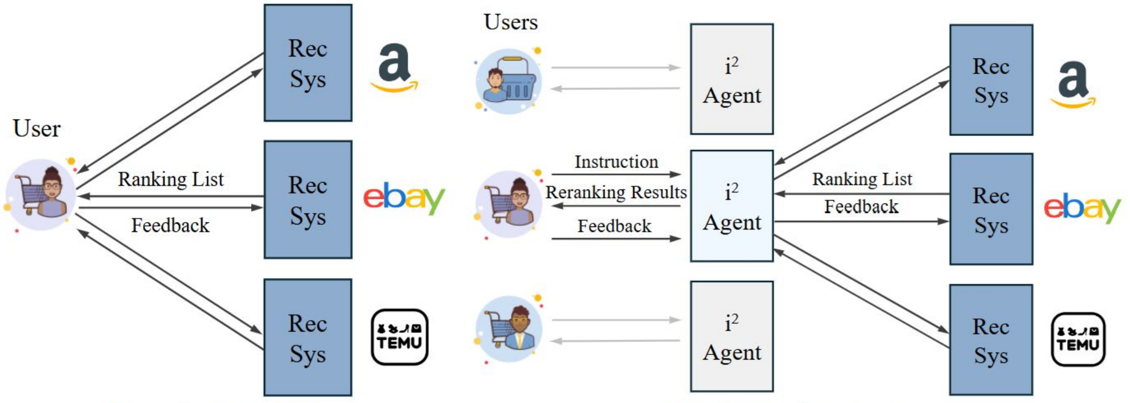
Agentic Recommender Systems (Agentic-RecSys)



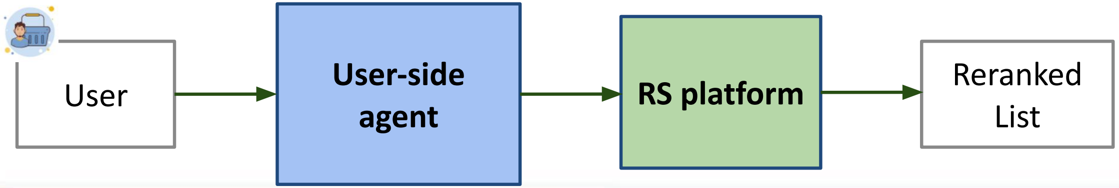
Representative Examples

Representative Examples: InstructAgent

- Agent as **user-side** mediator
- Parses instructions
- Re-ranks platform results
- Maintains user memory



(a) Previous User-Platform Paradigm (b) Ours User-Agent-Platform Paradigm

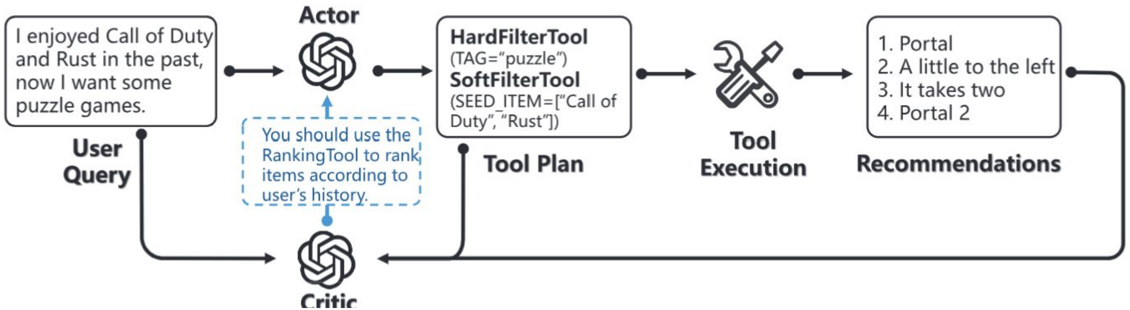


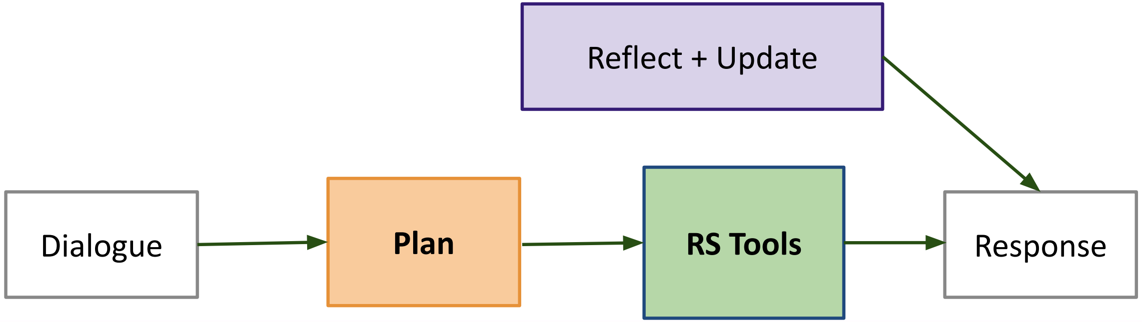
Xu, Wujiang, Yunxiao Shi, Zujie Liang, Xuying Ning, Kai Mei, Kun Wang, Xi Zhu, Min Xu, and Yongfeng Zhang. "Instructagent: Building user controllable recommender via llm agent." *arXiv e-prints* (2025): arXiv-2502.

Representative Examples: InteRecAgent

- Single-agent planner

Decompose ⟶ call RS modules ⟶ refine Agency: plan-act-reflect

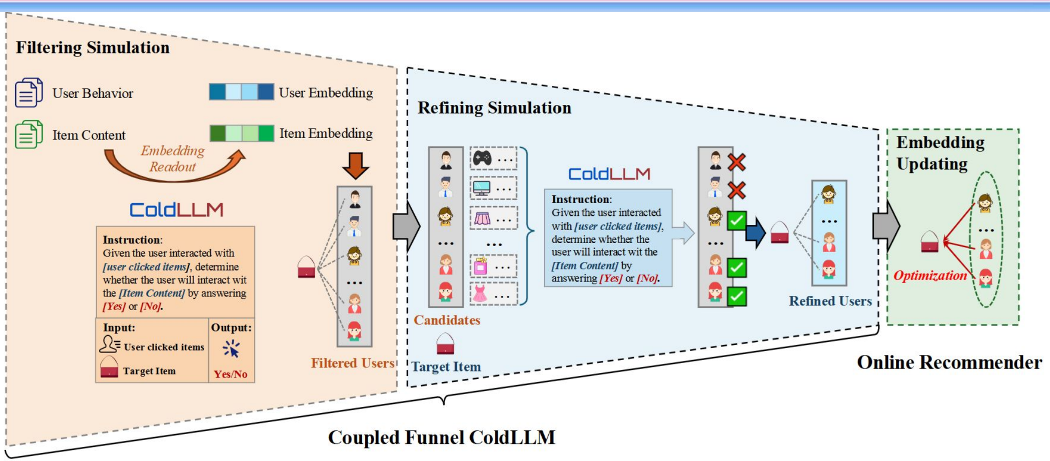


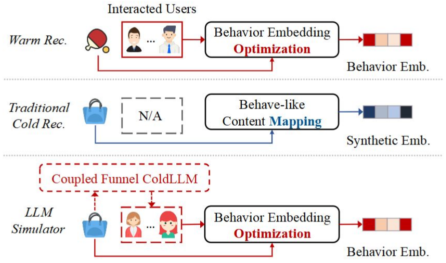


Huang, Xu, Jianxun Lian, Yuxuan Lei, Jing Yao, Defu Lian, and Xing Xie. "Recommender ai agent: Integrating large language models for interactive recommendations." *ACM Transactions on Information Systems* 43, no. 4 (2025): 1-33.

Representative Examples: ColdLLM

- Agent as **user simulator**
- Generate missing feedback
- Warm cold-start embeddings
- Production constraint: speed

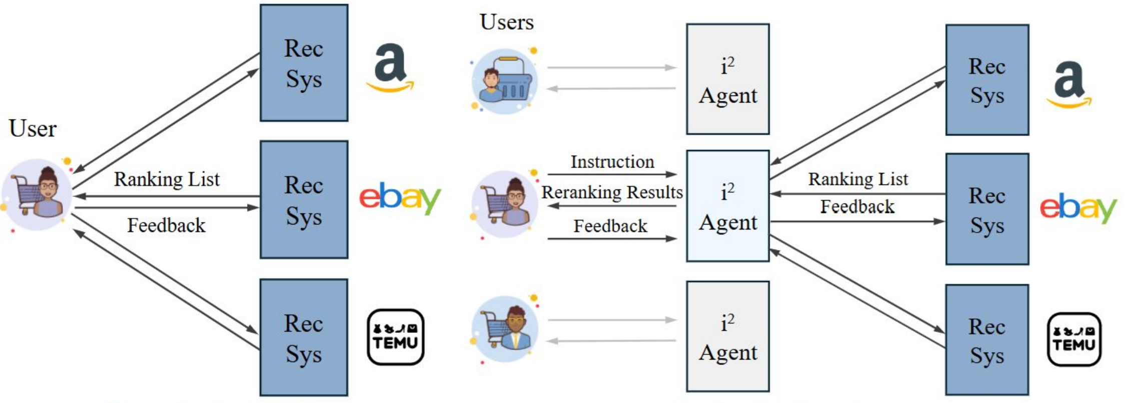




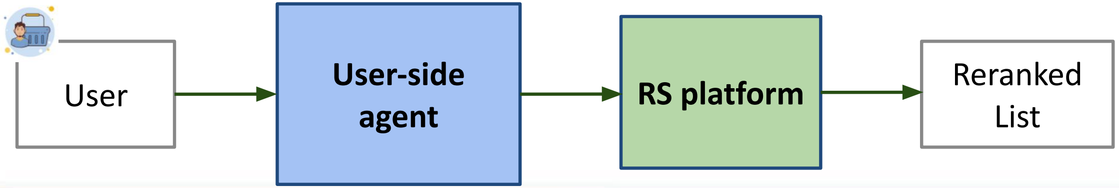
Huang, Feiran, Yuanchen Bei, Zhenghang Yang, Junyi Jiang, Hao Chen, Qijie Shen, Senzhang Wang, Fakhri Karray, and Philip S. Yu. "Large language model simulator for cold-start recommendation." In Proceedings of the Eighteenth ACM International Conference on Web Search and Data Mining, pp. 261-270. 2025.

Representative Examples: InstructAgent

- Agent as **user-side** mediator
- Parses instructions
- Re-ranks platform results
- Maintains user memory



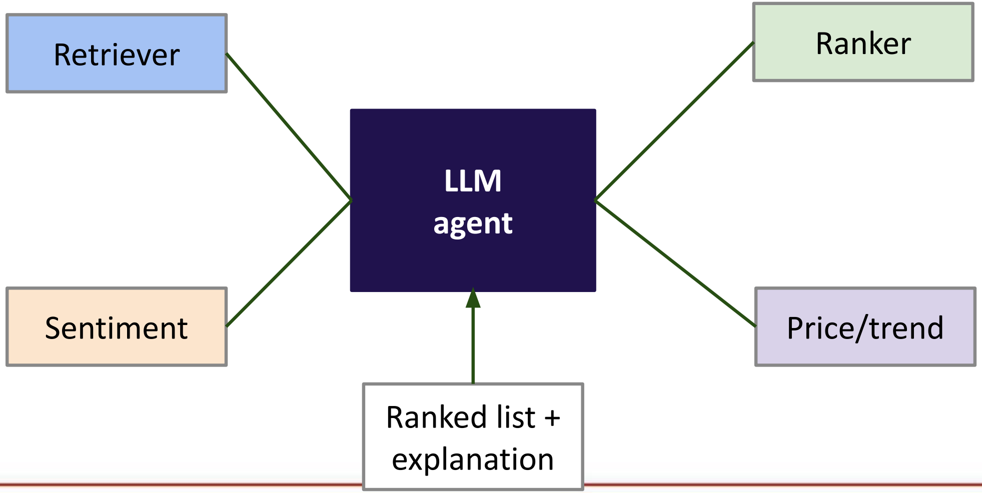
(a) Previous User-Platform Paradigm (b) Ours User-Agent-Platform Paradigm

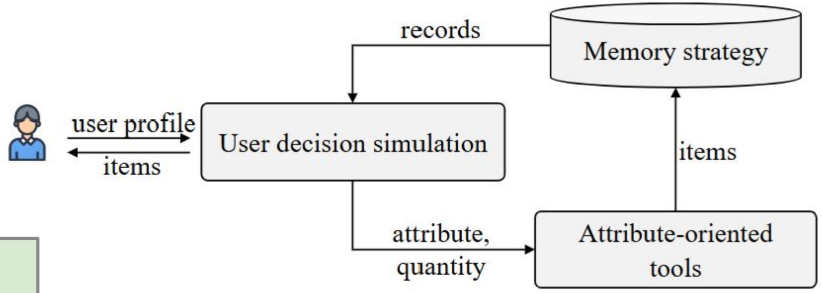


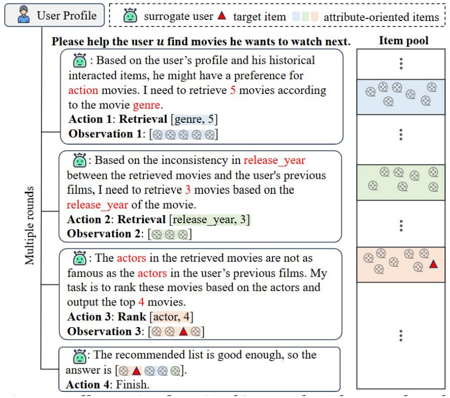
Xu, Wujiang, Yunxiao Shi, Zujie Liang, Xuying Ning, Kai Mei, Kun Wang, Xi Zhu, Min Xu, and Yongfeng Zhang. "Instructagent: Building user controllable recommender via llm agent." *arXiv e-prints* (2025): arXiv-2502.

Representative Examples: ToolRec

- Tool-augmented recommendation
- Retrieve
- Analyze
- Rank and Explain



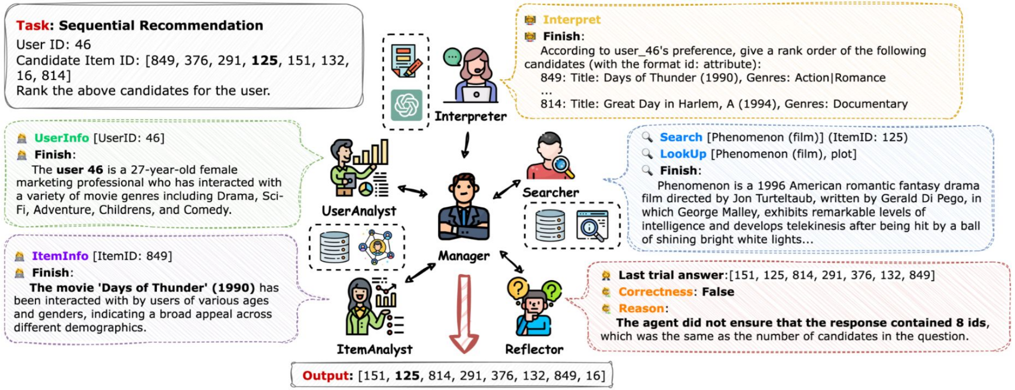


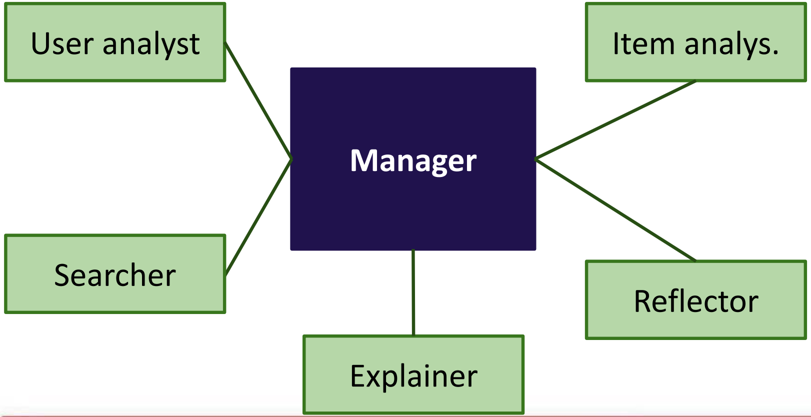


Zhao, Yuyue, Jiancan Wu, Xiang Wang, Wei Tang, Dingxian Wang, and Maarten De Rijke. "Let me do it for you: Towards llm empowered recommendation via tool learning." In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1796-1806. 2024.

Representative Examples: MacRec

- Multi-agent coordination
- Manager + specialists
- Reason - retrieve - critique - explain





Wang, Zhefan, 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. 2024.

Representative Examples: RecoWorld

- Environment for agentic recommendation
- Observe → act → feedback
- Simulator = tool + object of evaluation

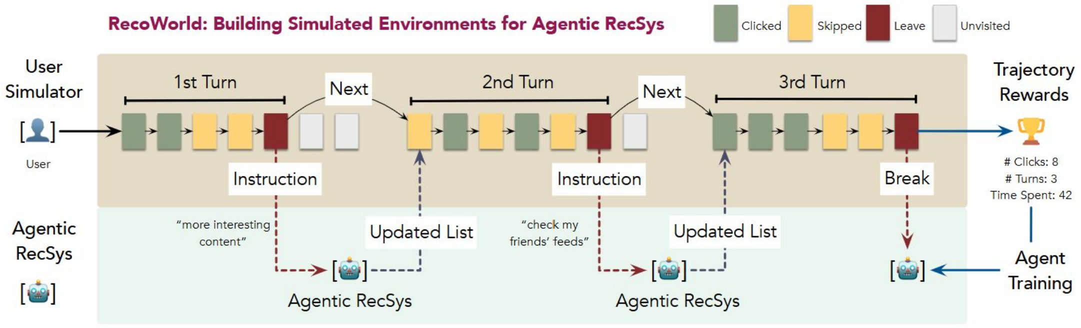
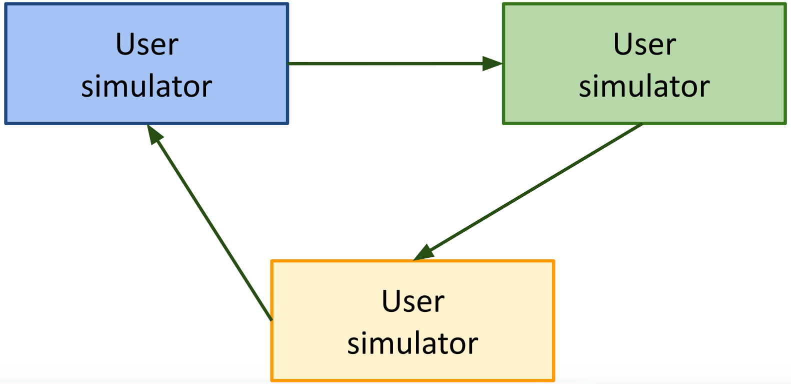


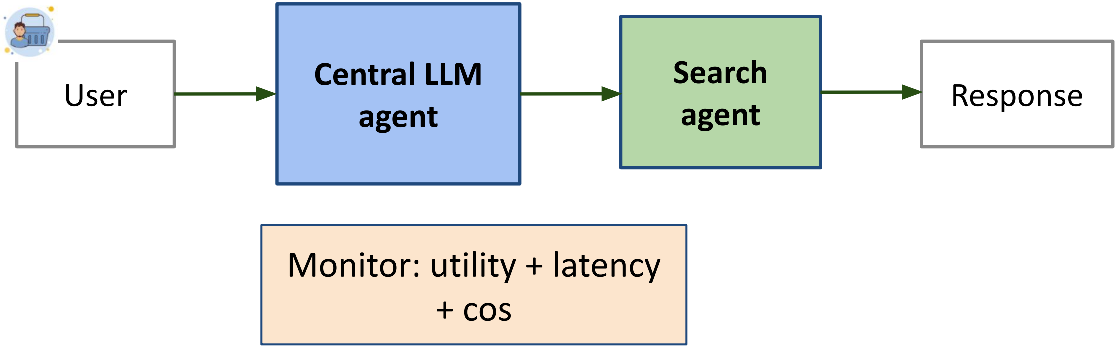
Figure 1: A simulated user interacts with an agentic RecSys over multiple turns within a session.

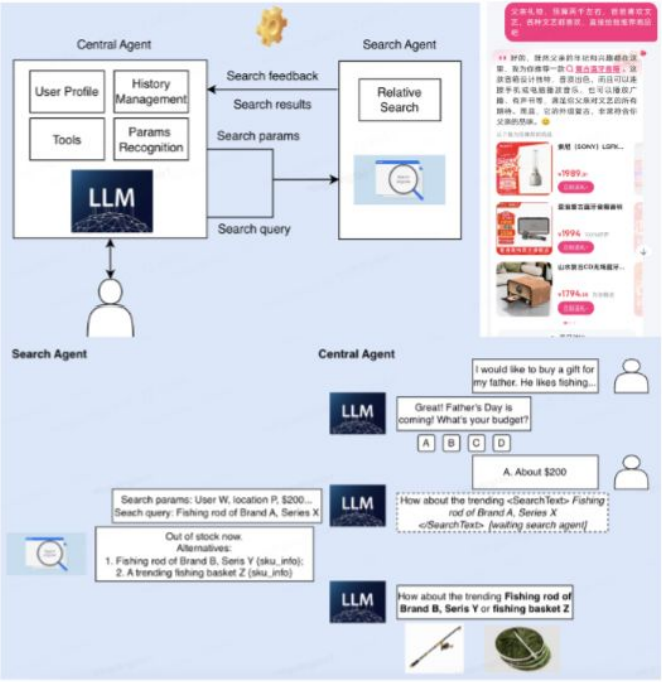


Liu, Fei, Xinyu Lin, Hanchao Yu, Mingyuan Wu, Jianyu Wang, Qiang Zhang, Zhuokai Zhao et al. "Recoworld: Building simulated environments for agentic recommender systems." In *Companion Proceedings of the ACM Web Conference 2026*, pp. 650-659. 2026

Representative Examples: Hybrid-MACRS

- Deployment-oriented agency
- Central LLM agent + search agent
- Measure utility, latency, cost





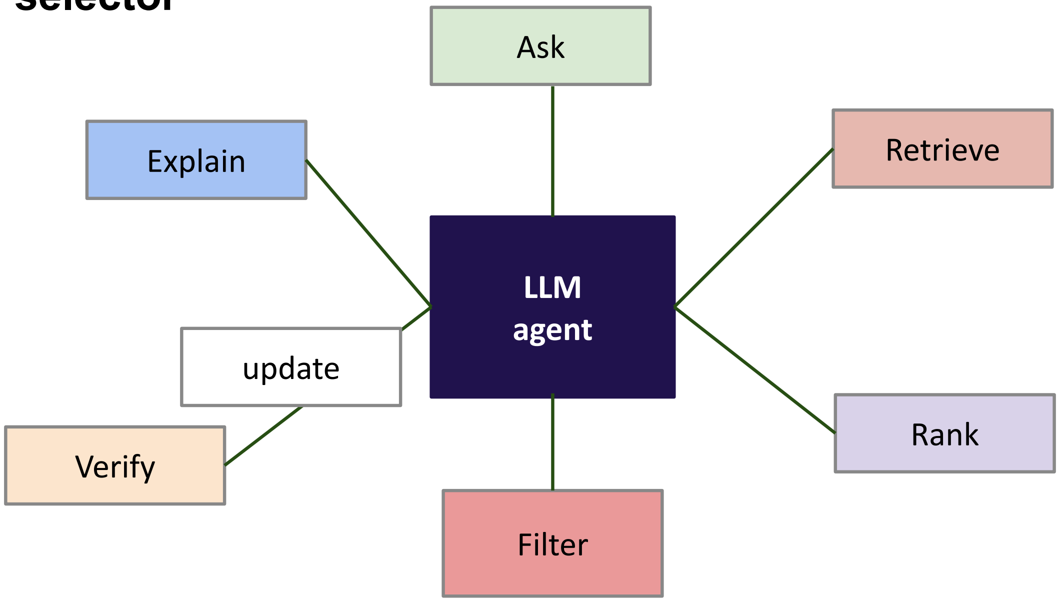
Nie, Guangtao, Rong Zhi, Xiaofan Yan, Yufan Du, Xiangyang Zhang, Jianwei Chen, Mi Zhou et al. "A hybrid multi-agent conversational recommender system with LLM and search engine in e-commerce." In *Proceedings of the 18th ACM Conference on Recommender Systems*, pp. 745-747. 2024.

Representative Examples: Hybrid-MACRS

The recommender becomes an action selector

Not only: recommend item i

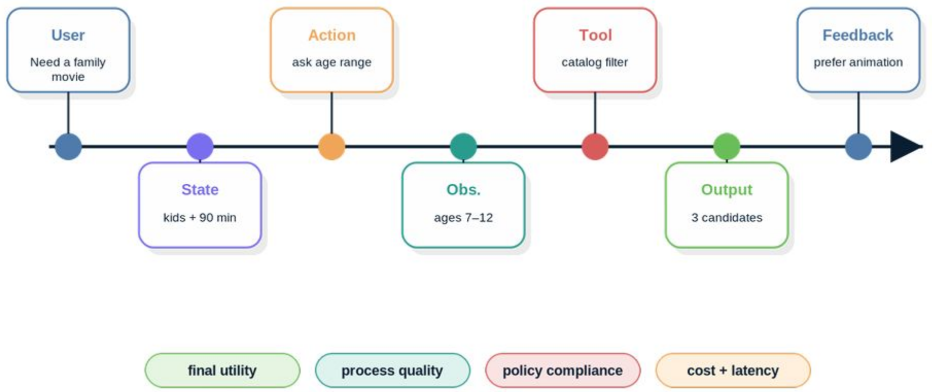
Also: ask, retrieve, rank, filter, verify, explain, update



Evaluation of Agentic-RecSys

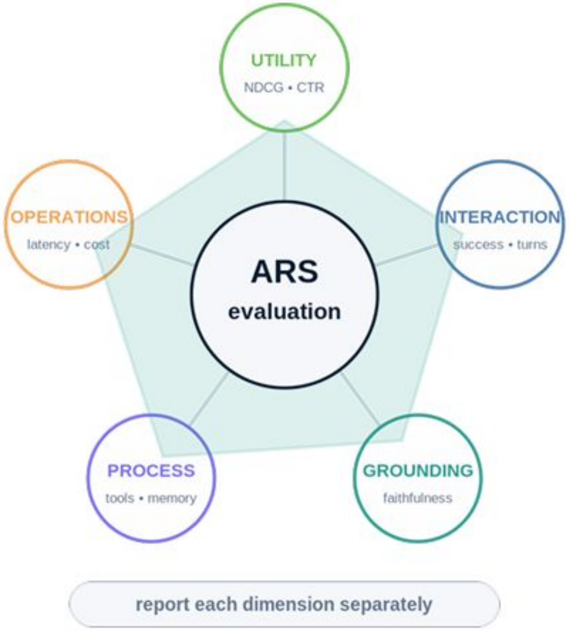
Evaluation: From Lists to Trajectories

- Evaluate trajectories, not only ranked lists
    - User input
    - State Action
    - Observation
    - Output Feedback



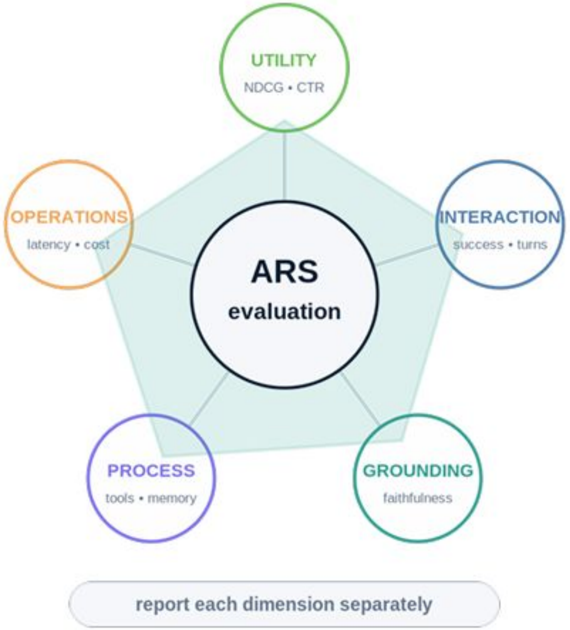
Evaluation Dimenions

- What to measure
  - **Utility:** NDCG, Recall, CTR
  - **Interaction:** success, turns, satisfaction
  - **Grounding:** faithfulness, citations
  - **Process:** tools, memory, planning
  - **Ops:** latency, cost, fallback



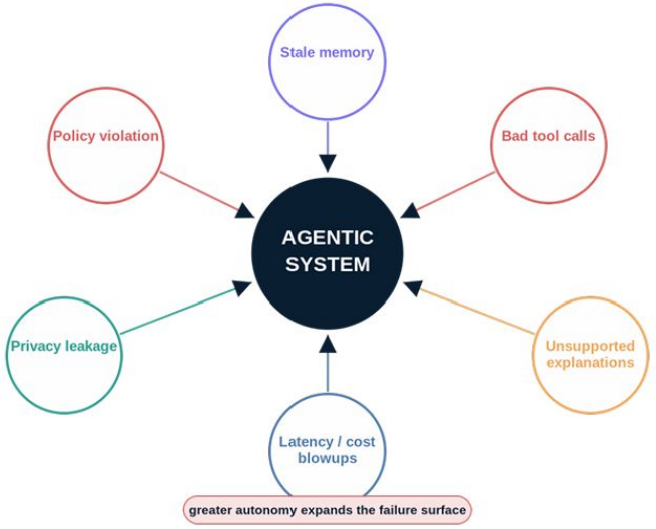
Holistic Evaluation Dimensions

- What to measure
  - **Utility:** NDCG, Recall, CTR
  - **Interaction:** success, turns, satisfaction
  - **Grounding:** faithfulness, citations
  - **Process:** tools, memory, planning
  - **Ops:** latency, cost, fallback



Reliability, safety, controllability

- More autonomy => more failure modes
- Stale memory
- Bad tool calls
- Unsupported explanations
- Privacy leakage
- Latency / cost blowups




Thanks!