RecSys 2025 Prague

Multi-Agentic Recommender Systems: Foundations, Design Patterns, and E-Commerce Applications — An Industrial Tutorial


Yashar Deldjoo[1]


Reza Yousefi Maragheh[2]


Chi Wang[3]


Jason Cho[2]


Derek Zhiyuan Cheng[3]

[1] Polytechnic University of Bari, Italy
[2] Walmart Global Tech, USA
[3] Google DeepMind, USA

ACM Recommender Systems Conference, (RecSys'25)

Prague, Czech Republic, 22-26 Sep 2025

Outline

Introduction (Yashar- 30 min)

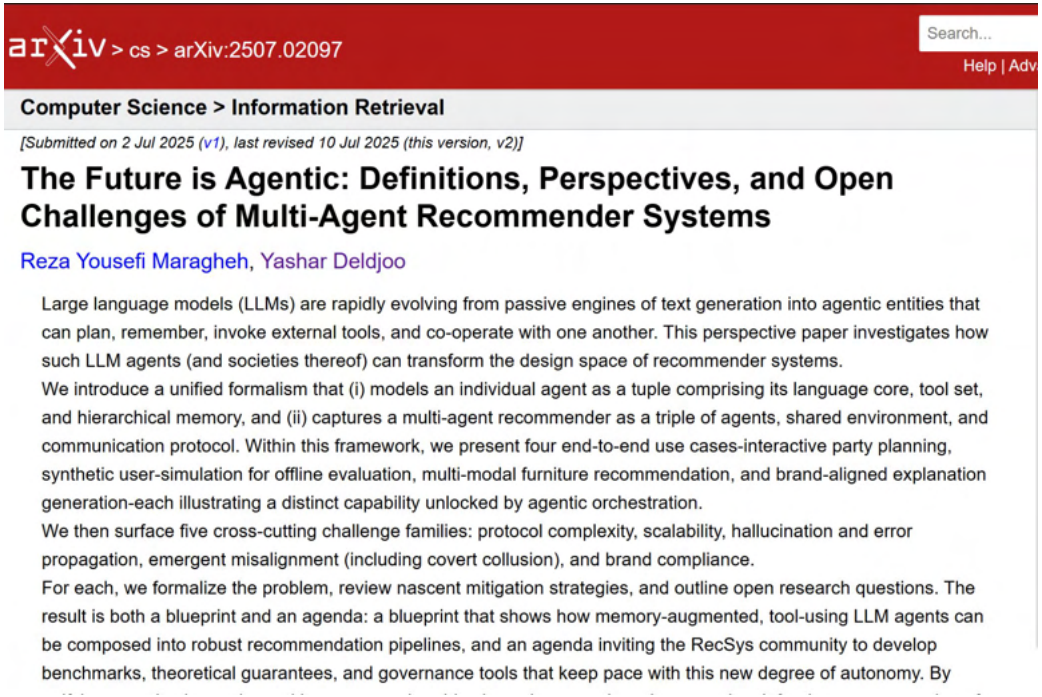
- **Classical recommendation → generative models**
  - Context to traditional (classical) models
  - Definition and types of generative models
  - Applications Area
- **Generative → Agentic models**
  - Evolution
  - Definitions
  - Different Cafoeris of RS
  - Level of autonomy
  - Language agents vs RL agents
  - Use Cases

PART I- Introduction, Foundation, Alphabets

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 ACM TORS 2025 Perspective paper!



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

Check out our ACM TORS 2025 Perspective paper!

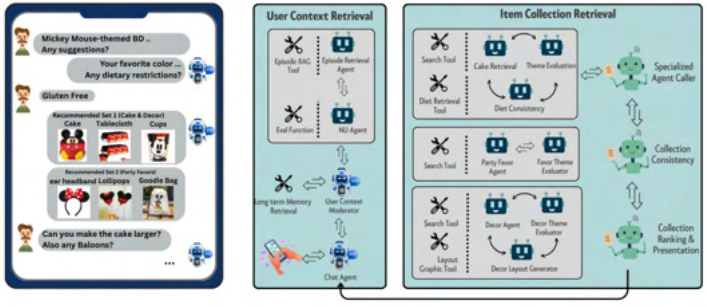
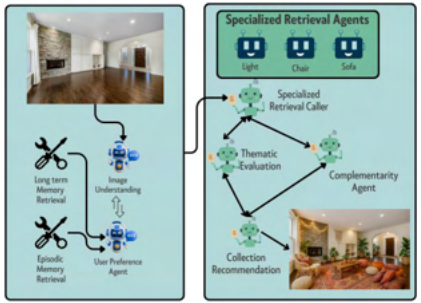
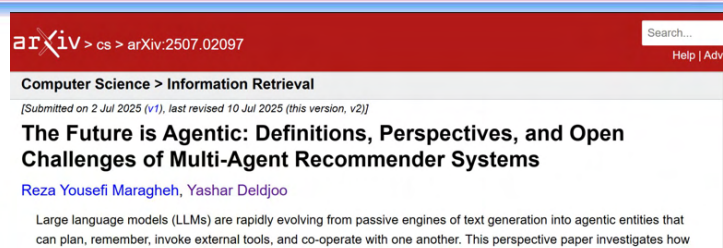


Table 1. Core Capabilities of Agentic Recommender Systems

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

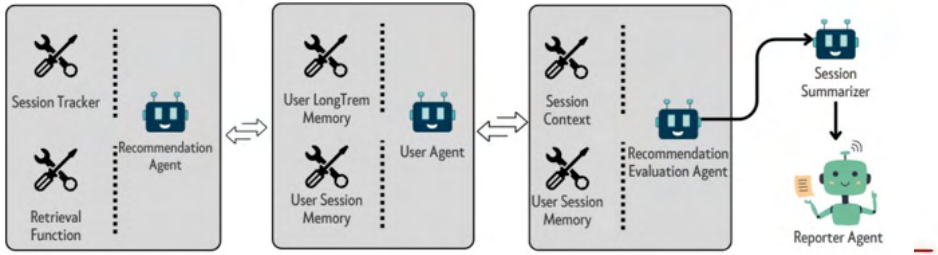
Reza Yousefi Maragheh, Yashar Deldjoo

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.

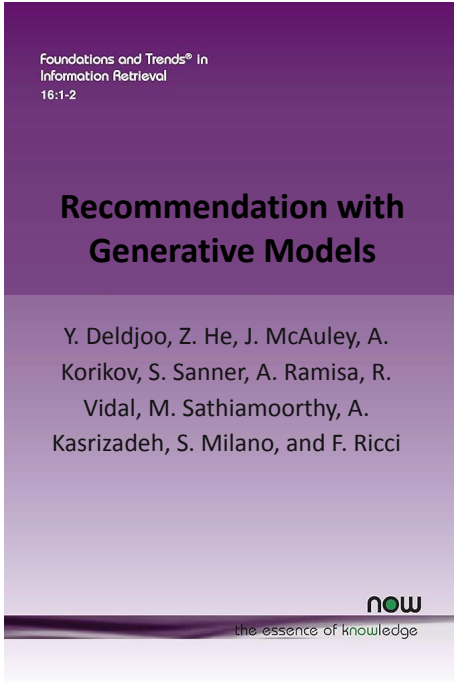
We then surface five cross-cutting challenge families: protocol complexity, scalability, hallucination and error propagation, emergent misalignment (including covert collusion), and brand compliance.

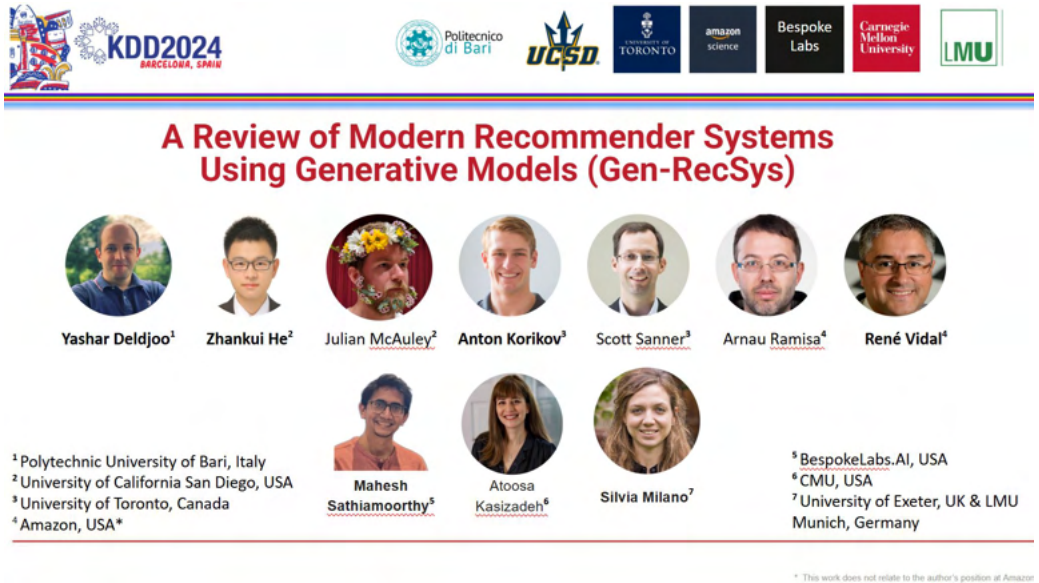
For each, we formalize the problem, review nascent mitigation strategies, and outline open research questions. The result is both a blueprint and an agenda: a blueprint that shows how memory-augmented, tool-using LLM agents can be composed into robust recommendation pipelines, and an agenda inviting the RecSys community to develop benchmarks, theoretical guarantees, and governance tools that keep pace with this new degree of autonomy. By



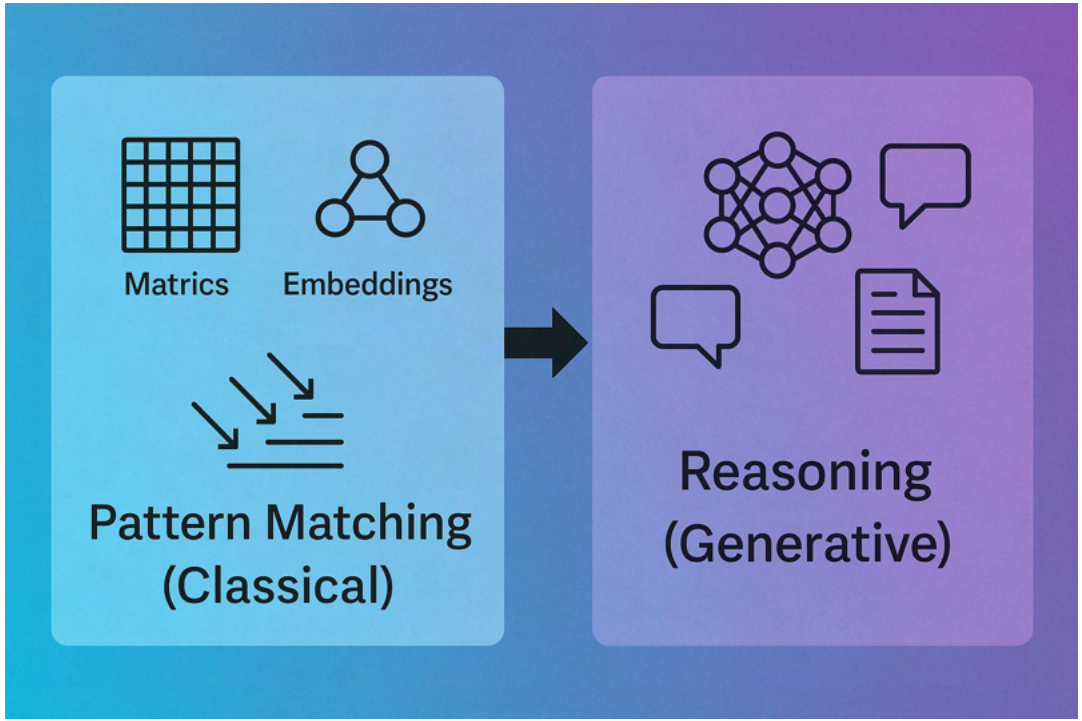
Link: https://arxiv.org/abs/2507.02097

Also Take a look at our Gen-RecSys Work

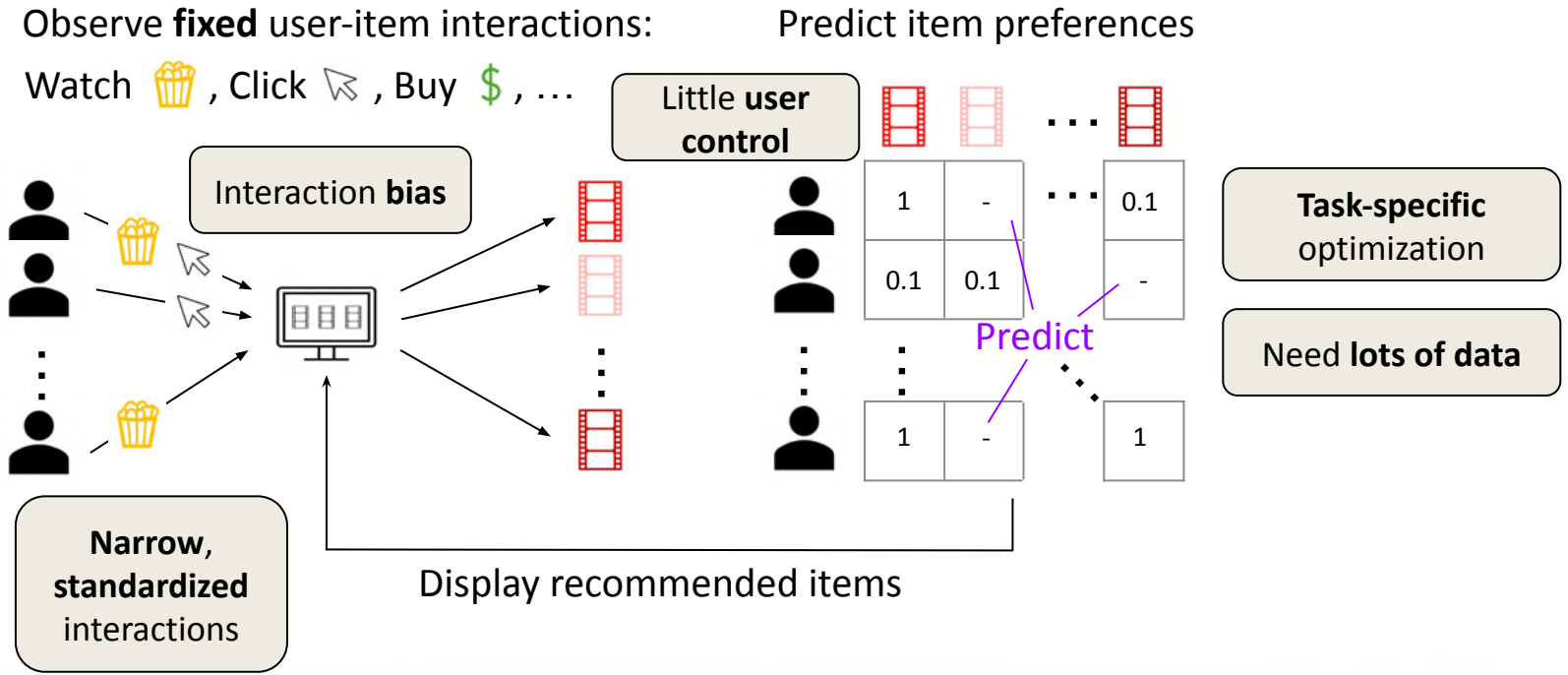




Link: https://arxiv.org/abs/2409.15173


Matrics
Embeddings
Pattern Matching
(Classical)
Reasoning
(Generative)

Traditional Recommendation (Recap)



LLMs Unlock NL in Recommender Systems

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

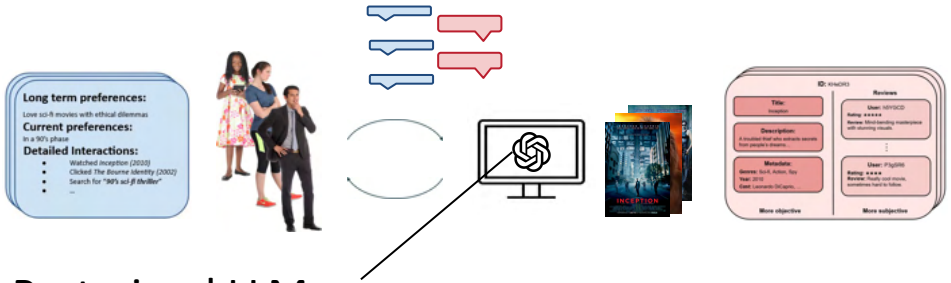
User preferences

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:

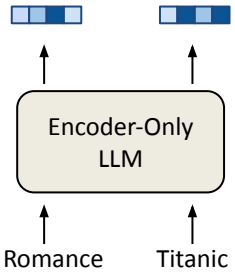
Nuanced **personalization** in **diverse contexts**

Interactive, real-time recommendation

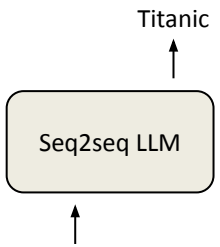
Faster system design and deployment

NL-based Models

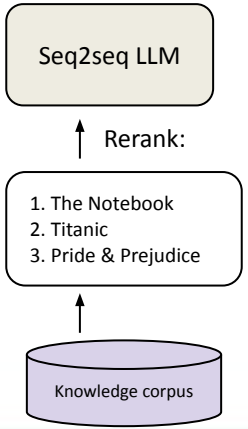
Encoder-Only LLMs



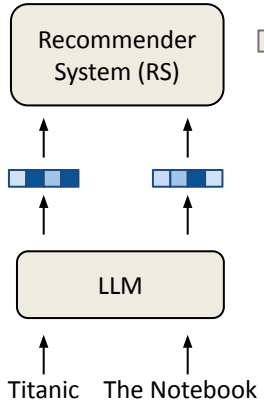
Sequence to Sequence (Seq2seq) LLMs



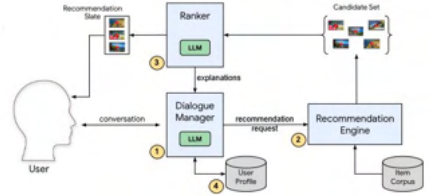
Retrieval-Augmented Generation (RAG)



LLM Representation Generation



Conversational Recommendation



Editable NL User Profiles

User Interactions

90's sci-fi thriller

LLM generated summary

NL Templates

NL User Profile

ID: P3gSR6

Long term preferences: Love sci-fi movies with ~~ethical dilemmas~~ lots of action

Current preferences: In a 90's phase

Detailed Interactions:

- Watched *Inception (2010)*
- Clicked *The Bourne Identity (2002)*
- Search for "***90's sci-fi thriller***"
- …

Inspect & Edit

Control

Privacy

Feedback Incentive

Real-time preferences

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

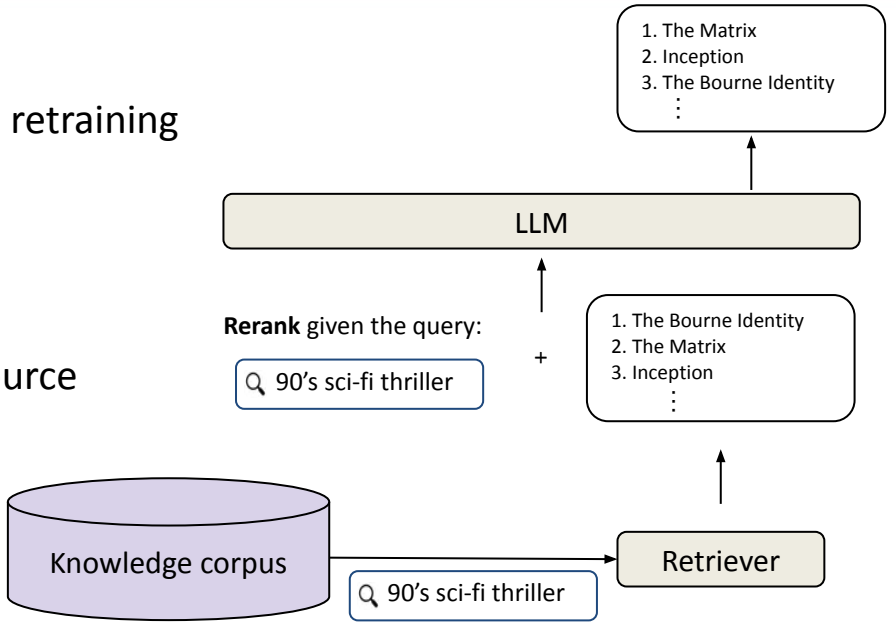
Retrieval Augmented Generation (RAG)

Limitations of LLM internal knowledge:

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

RAG:

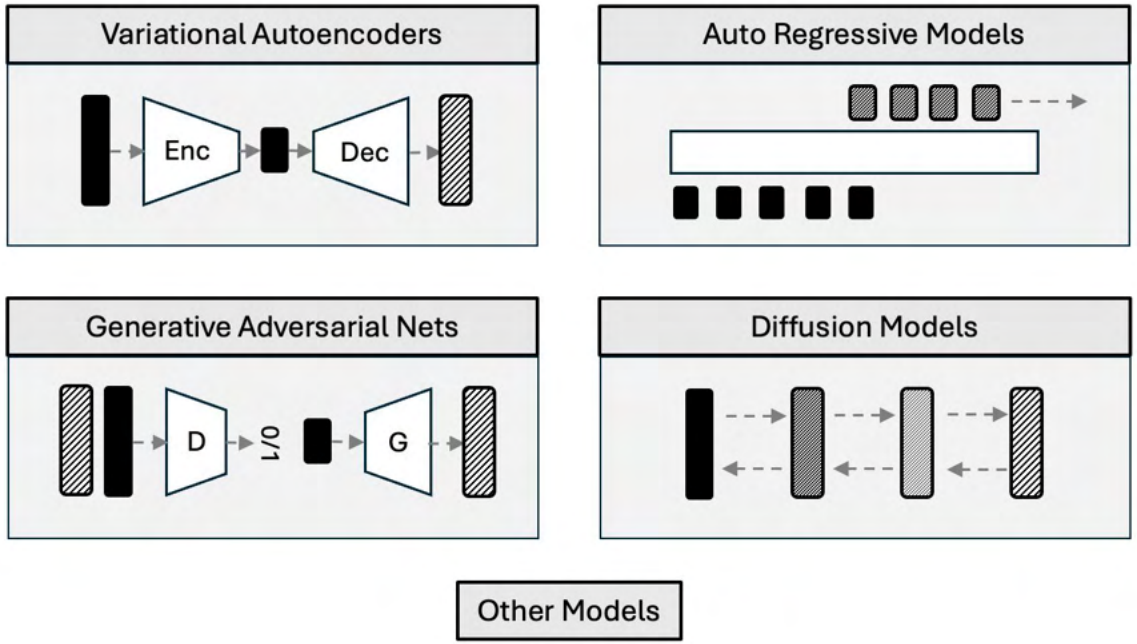
- Retrieve information from an external source
- Prompt LLM with retrieved information
- Advantages:
  - Smaller LLM
  - Opportunity for source attribution



Example: RAG for recommendation reranking

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/

What are Generative Recommender Systems? (Gen-RecSys)

Recommender systems enhanced by Generative AI

Structured List
generate lists, sequences or bundles tailored to user

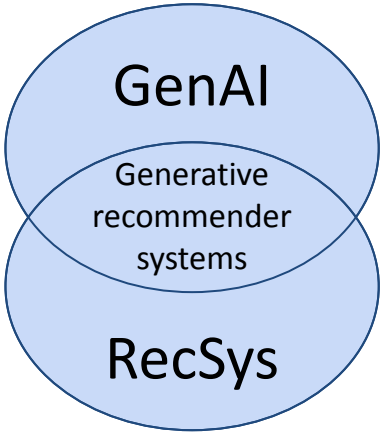
Conversational & textual
Generate natural language responses, recommendations and explanations

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**

Deep Generative Models (DGMs)

- DGMs

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

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

- Attention!

  high-dimensional

  1. **Different** $\mathbf{X}$ in *RecSys w/ UI Data*
  2. **Different** reasons to model $p(\mathbf{x}|\cdot)$ 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** $p_\theta(\mathbf{x}_t|\mathbf{x}_{<t}, \cdot)$
- ……

Some motivating context (from r/MovieSuggestions)

USER: Normally I watch horror and thrillers , but I've been really into comedies lately , specifically from the mid to late 90's (and some 00's) . I watched White Chicks last night and can't remember the last time I laughed so hard that my face hurt! Looking for recommendations on movies that will give me a good laugh , preferably 90's but anything will do !

SYSTEM: 90's era chuckles: ***Friday, Groundhog Day, Kingpin, The Cable Guy, Rush Hour, and Office Space***.

Q: How should a recommender system generate these recommendations?

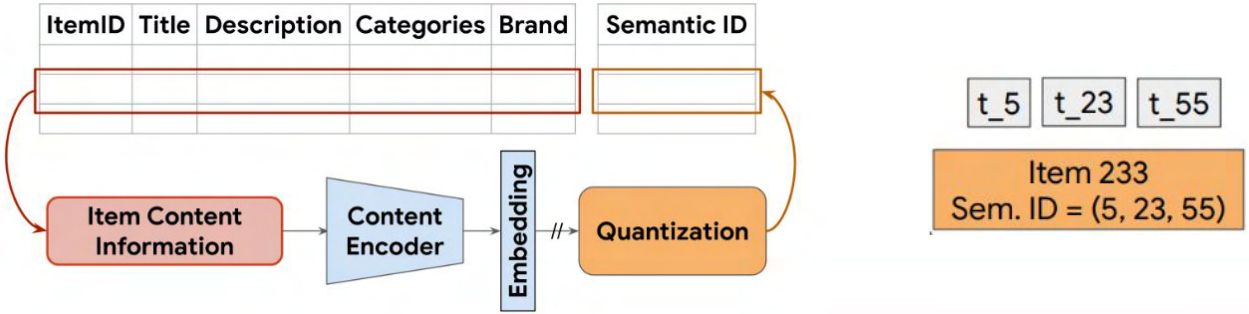
Some motivating context

Q: How should a recommender system generate these recommendations?

- **"Pure" language:** just generate some tokens!
  - Hallucination? Constrained beam search? Controllability? Fairness/bias?
- Add items as **new tokens** in the language model; maybe connect it with a pre-trained recommender
  - Scalability? Our Amazon review dataset has 48M item IDs! Cold-start?
- **Something else?** *Are there options for tokenization between language and items?*

Lately: **Generative recommendation**

- **Semantic IDs**: a few discrete tokens from a shared vocabulary, representing semantics for each item (e.g. quantized sentence embeddings)
- ***Generative Recommendation****:* models that autoregressively generate semantic IDs as recommendations



Lately: **Generative recommendation**

- "Recommendation" becomes a **seq**-to-**seq** generation problem:

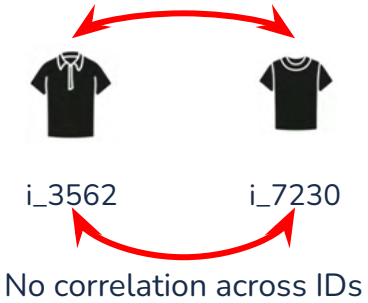
  Input: user interacted items {c_{11}, c_{12}, c_{13}, c_{14}, c_{21}, c_{22}, ...}

  Output: next item {c_{t1}, c_{t2}, c_{t3}, c_{t4}}

Item IDs vs Semantic IDs

Item IDs:

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



Semantic IDs:

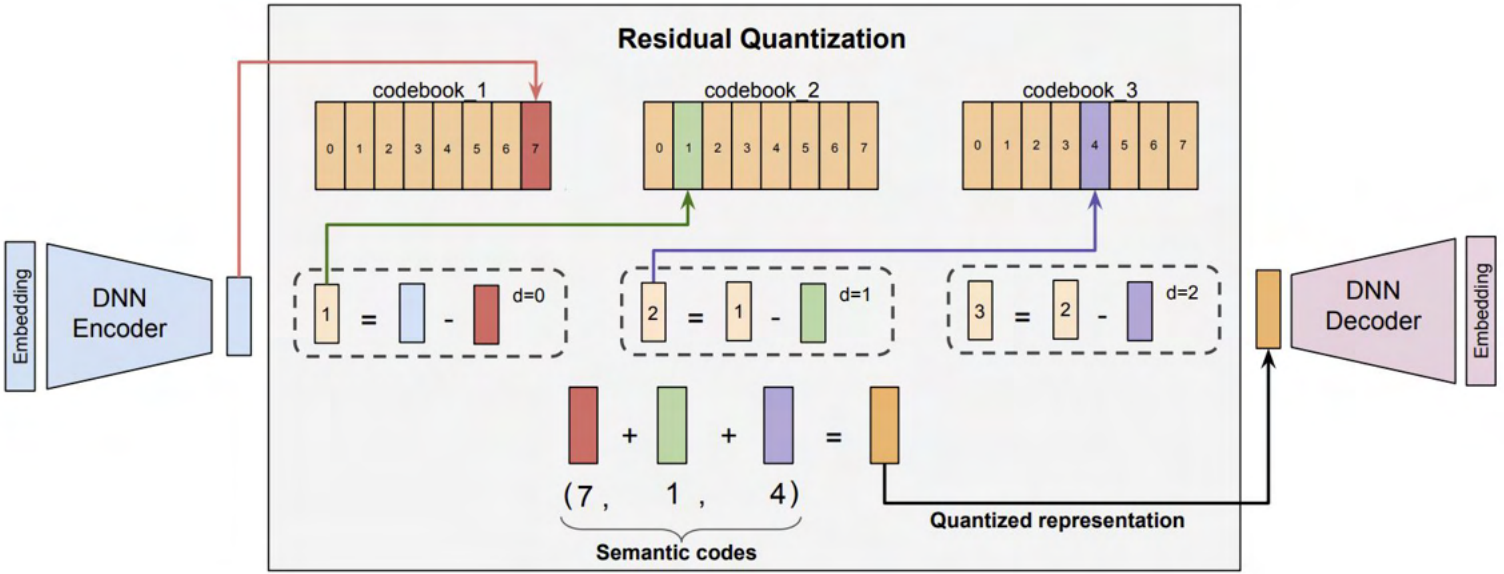
- **Prior knowledge**: semantically similar items have similar tokens;
- **#Parameters**: proportional to #tokens from a compact vocabulary;

[t3, t65, t131, t243]

[t3, t65, t173, t243]

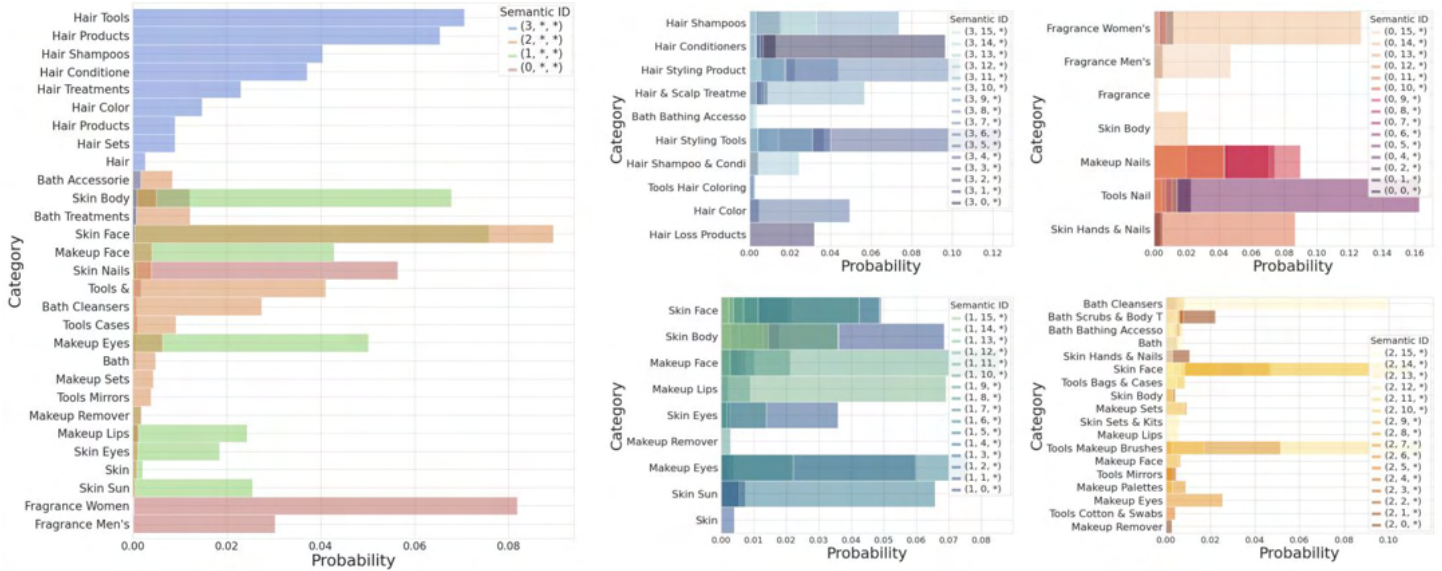
Lately: Generative Recommendation

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



Lately: Generative Recommendation

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




Generative
Agentic
Pursues a
goal-oriented task

Comparison: Personalization

Classical	Generative (pLLMs)	Agentic
Pattern matching	Semantic Reasoning	Goal driven
Surface-level personalization	Contains deeper understanding, infers deeper context	aligns recommendations with user goal

Evolution of Recommendation Models

Comparison: Proactivity

Classical	Generative (pLLMs)	Agentic
Reactive	Reactive answers	Proactive guidance
suggests only when asked	answers specific prompts	initiates suggestions and leads the conversation

Evolution of Recommendation Models

Comparison: Context Awareness

Classical	Generative (LLMs)	Agentic
Past behavior	Prompted context	Situational, Multimodal Context

can also uses context explicitly given in the query/prompt

Evolution of Recommendation Models

Comparison: Task Scope

Classical	Generative (pLLMs)	Agentic
Single step recommendation	Single-turn answer	Multi-Step Planning

plans and executes complex goals

Evolution of Recommendation Models

Comparison: Adaptivity

Classical	Generative (pLLMs)	Agentic
Offline training	Frozen weights at inference	Lifelong adaptation
Offline training and batch updates (rarely refreshed)	Pretrained model weights: static at inference	plans and executes complex goals

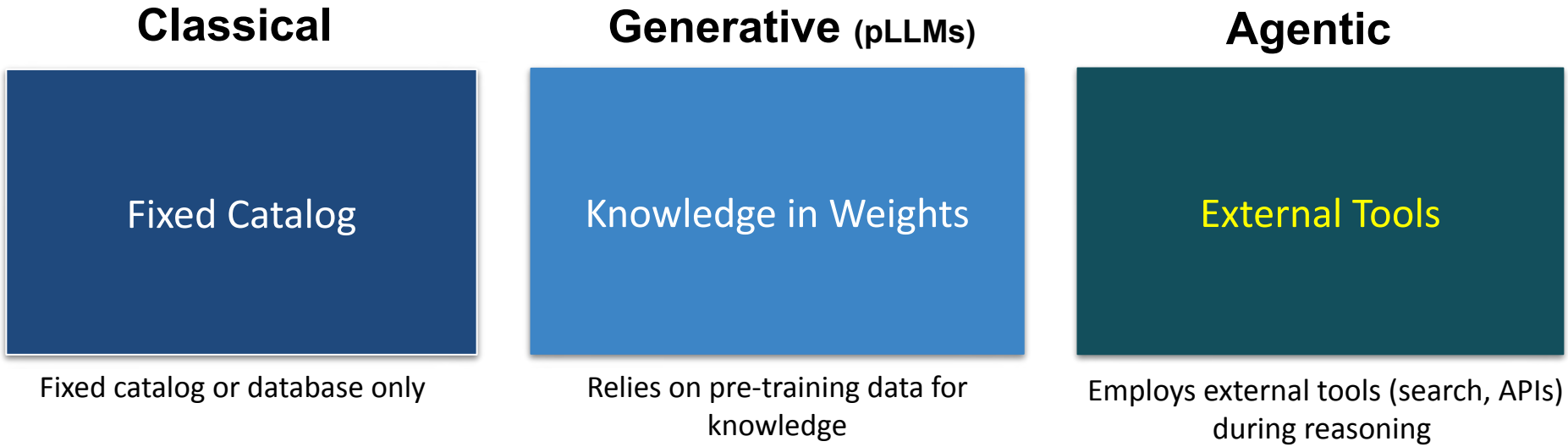
Evolution of Recommendation Models

Comparison: Memory

Classical	Generative (pLLMs)	Agentic
No explicit working memory	Context window	Structured long-term memory
No explicit short-term memory	Session context limited by the LLM's context window	Structured long-term memory for persistent user history

Evolution of Recommendation Models

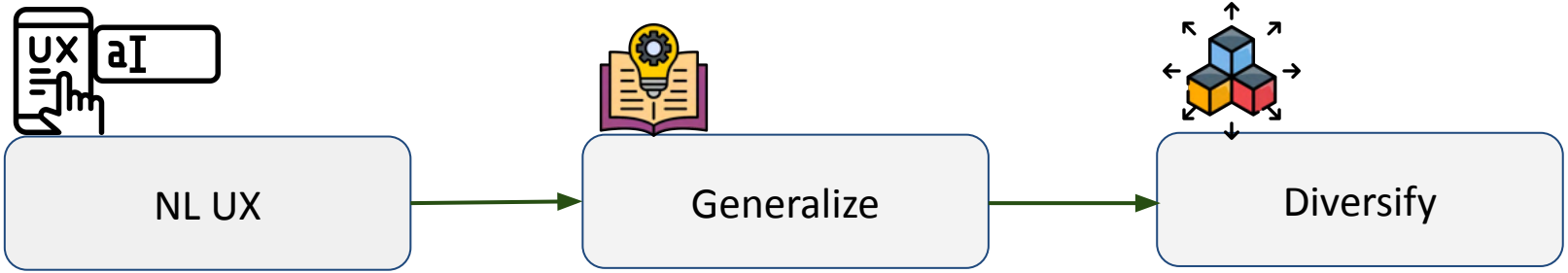
Comparison: Tools and Knowledge



Gen-RecSys Strength

Strength at a glance:

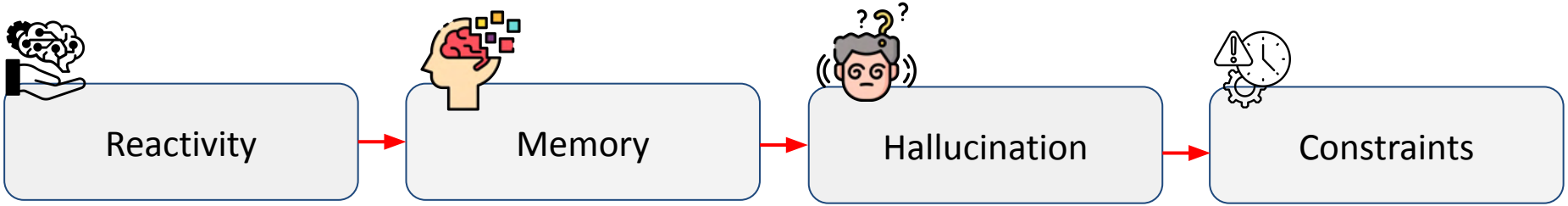
- Natural language interface accelerates preference elicitation.
- Pretrained knowledge improves zero-/few-shot generalization.
- Richer UX via explanations and diversified suggestions



Gen-RecSys Limitations

Risks hotspots:

- Mostly reactive → it must be prompted with a **specific query** before generating a response
- Limited long-term memory.
- Constraint satisfaction can fail without explicit checkers.
- Hallucination risk without grounding + verification.



What is a Deep Research AI Agent

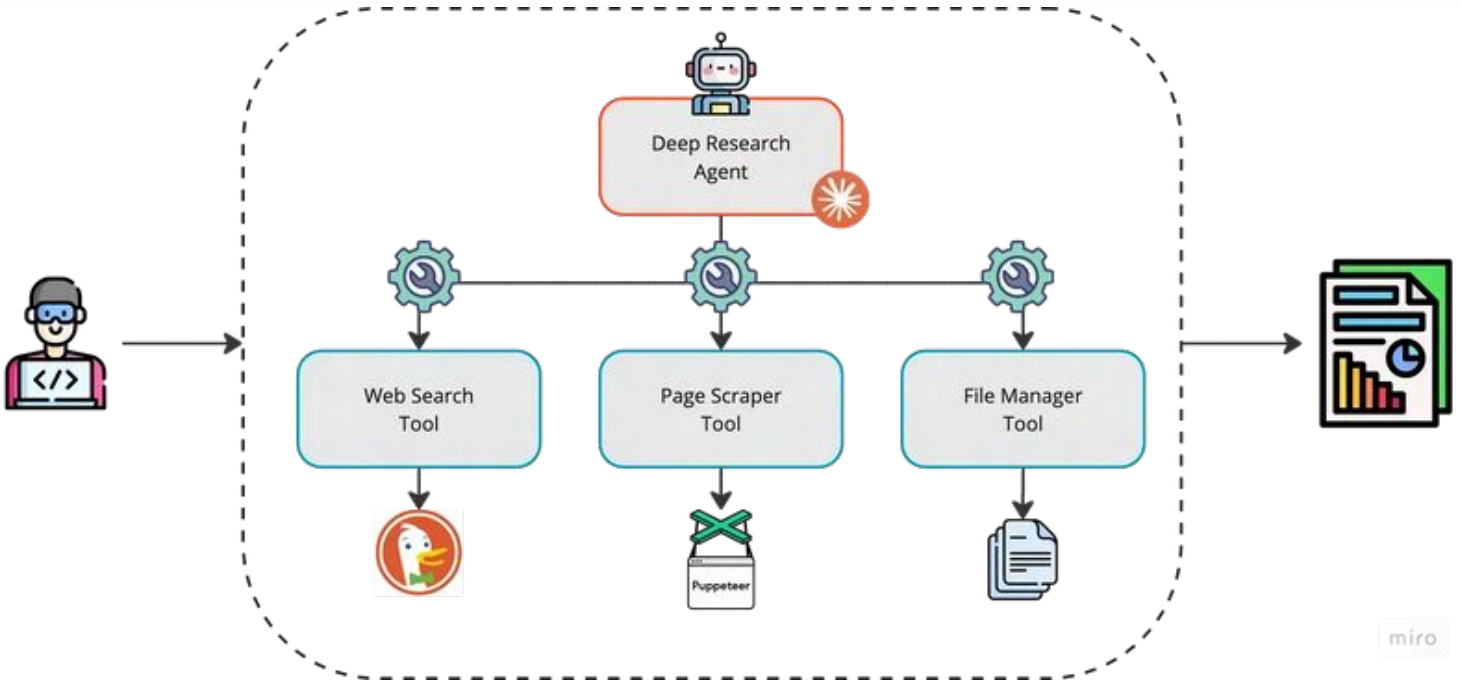
Definition: Autonomous system leveraging large language models (LLMs) to perform complex research tasks with minimal human intervention.

Purpose: Automate tasks such as literature reviews, competitive analysis, and technical documentation.

Key Features:

- Multi-step reasoning
- Tool integration
- Memory management
- Autonomous decision-making

Simple Architecture



https://dev.to/subeshb1/building-your-very-own-deep-research-agent-4k77

Why Agentic Recommender Systems?

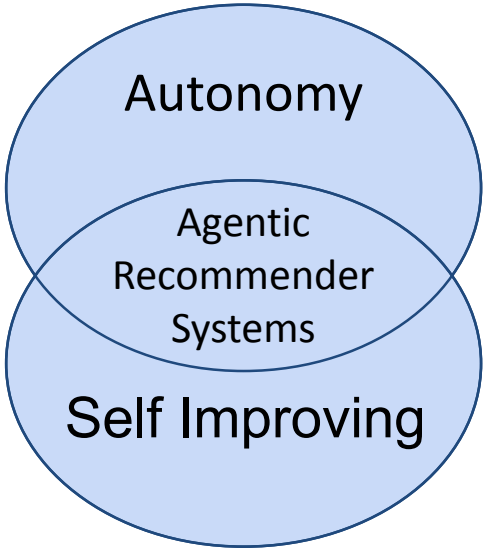
Recommenders are evolving from **static predictors** to **adaptive agents** capable of reasoning, planning and interaction.

- Large language models (LLMs) enable natural language understanding and flexible tool usage.
- Agentic systems incorporate **memory**, **reflection** and **feedback** to personalize and align recommendations.

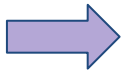
Agentic Recommender Systems (Agentic-RecSys)

- **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

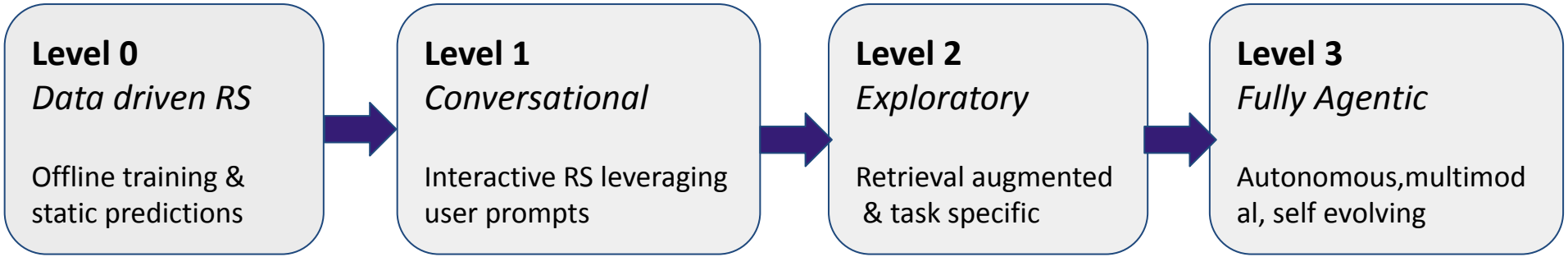
Agentic Recommender Systems (Agentic-RecSys)

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

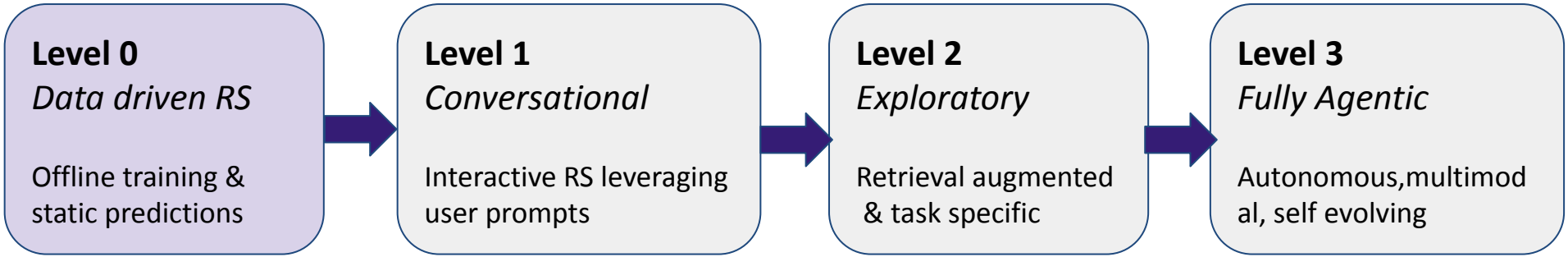
“show items” → “solve tasks”

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

Four Categories of Recommender Systems

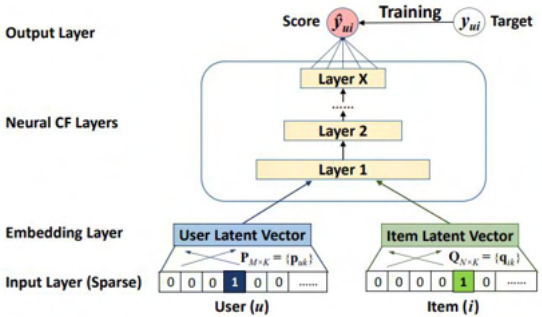


Four Categories of Recommender Systems

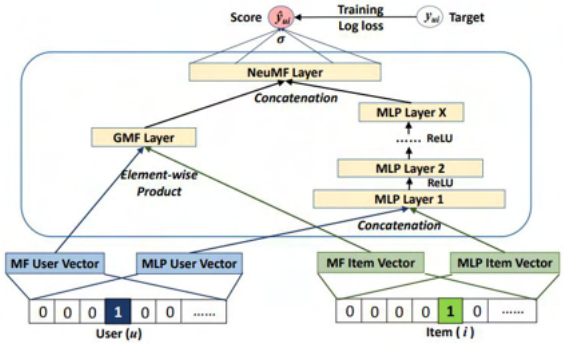


Level 0 - Data Driven RS

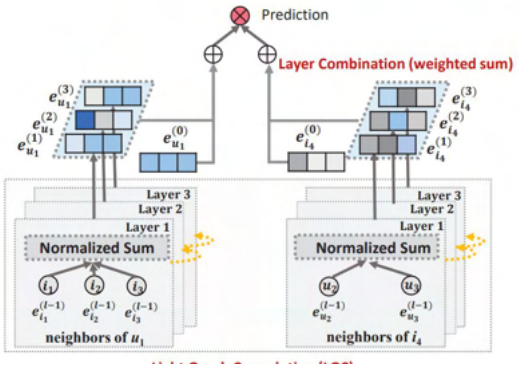
. Latent-space Modeling in Mainstream CF Models



Neural collaborative filtering framework (NCF)



Neural matrix factorization model (NeuMF)



Light Graph Convolution (LGC)

[1] He, X., Liao, L., Zhang, H., Nie, L., Hu, X., & Chua, T. S. (2017, April). Neural collaborative filtering. In Proceedings of the 26th international conference on world wide web (pp. 173-182).

[2] He, X., Deng, K., Wang, X., Li, Y., Zhang, Y., & Wang, M. (2020, July). Lightgcn: Simplifying and powering graph convolution network for recommendation. In Proceedings of

Four Categories of Recommender Systems

Characteristics

- Offline training and static predictions
- Learn from historical interactions; no dialog or reasoning

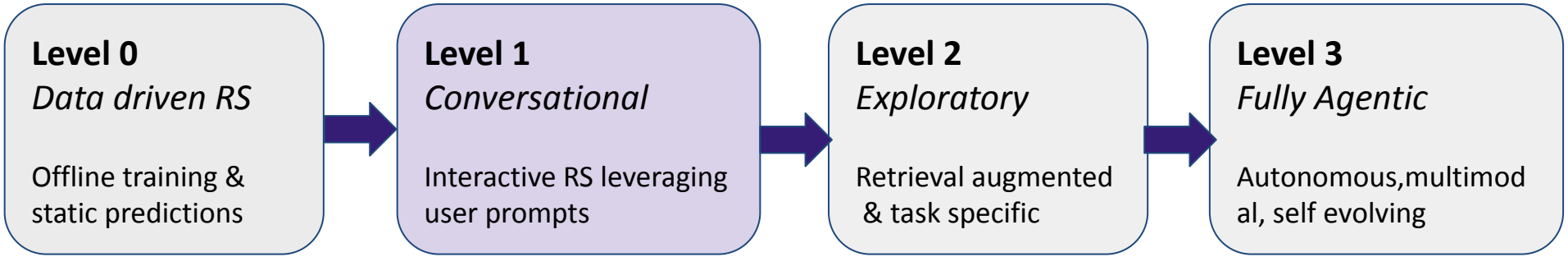
Representative models

- Collaborative Filtering (k NN, matrix factorisation: SVD, SVD++, Funk SVD);
- Neural models: Neural CF, Wide & Deep, BERT4Rec, SASRec, GNN
- Hybrid: content based + collaborative (e.g., Fab system)

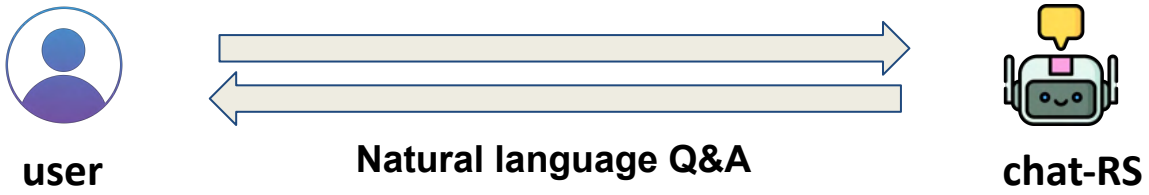
Level 0
Data driven RS

Offline training & static predictions

Four Categories of Recommender Systems



Level 1 - Conversational



- Interactive dialogue between user and system
- LLM interprets user intent and asks clarifying questions

Example: "Recommend a romantic movie like Titanic" → suggestions and follow up questions

Level 1 - Conversational

Characteristics

- Interactive dialogue with users via natural language
- Collect preferences on the fly and adjust recommendations

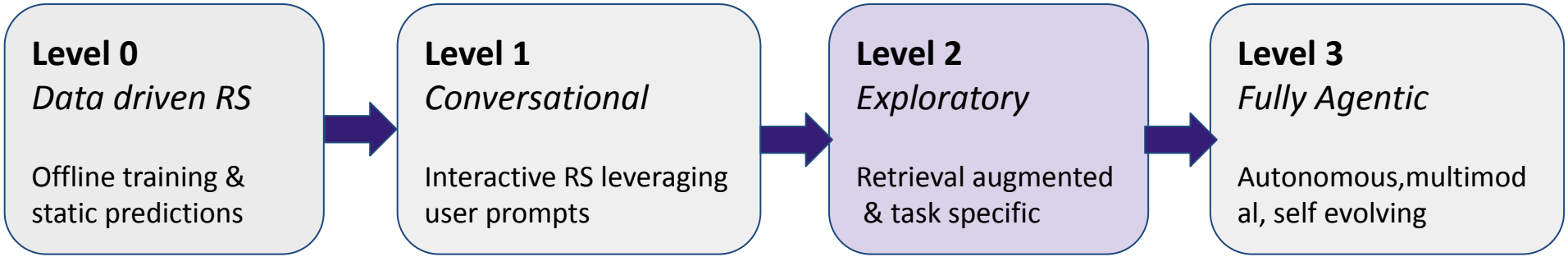
Representative models

- Early CRSs: ReDial (2018), KBRD, CR-MLS, etc.
- ChatRec and LLaMA Rec: LLM-based conversational recommenders
- MACRec & RecMind: multi agent and self inspiring architectures enabling conversation

Level 1
Conversational

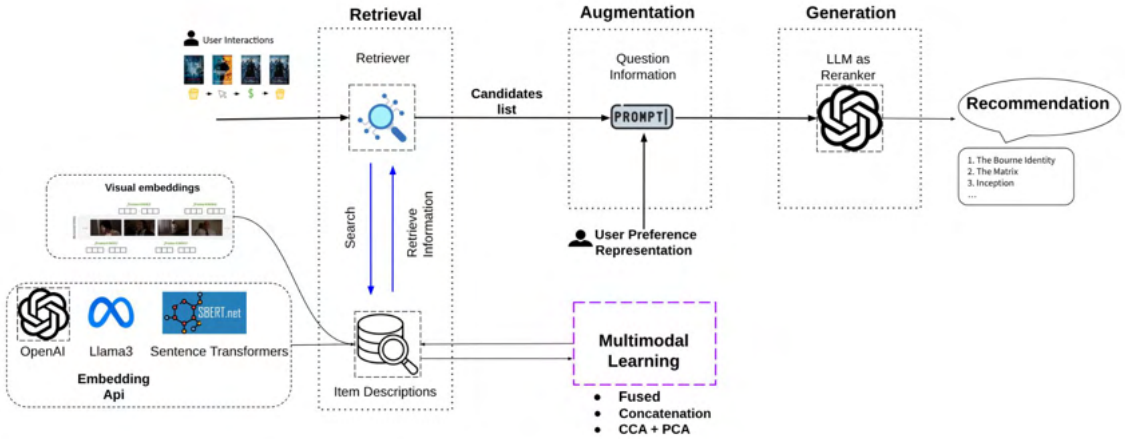
Interactive RS leveraging user prompts

Four Categories of Recommender Systems



Level 2 - Exploratory (RAG)

- Augments LLM knowledge with external data
- Retrieves relevant documents/items on demand
- Enables source attribution and up to date answers



Example: "90s sci fi thriller" → retrieve candidates and let LLM re rank/explain

Four Categories of Recommender Systems

Characteristics

- Retrieval augmented reasoning: mix LLMs with search and database tools;
- Task specific modules for rating prediction, sequential and direct recommendations

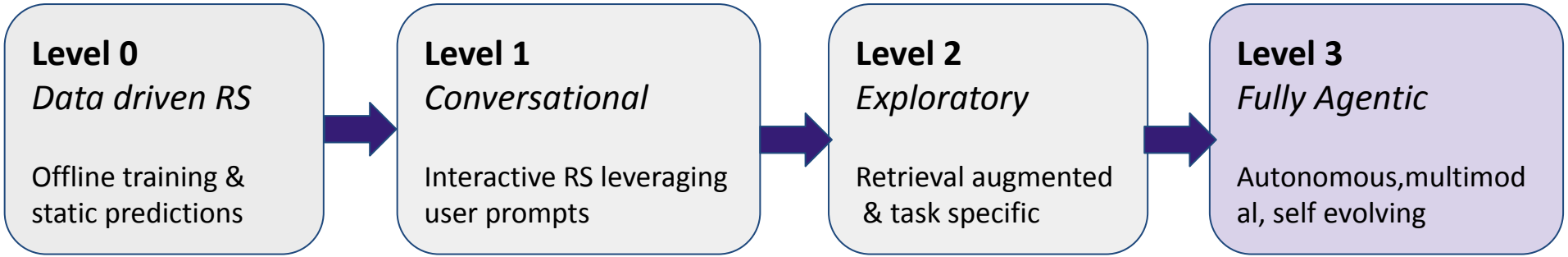
Level 2
Exploratory

Retrieval augmented & task specific

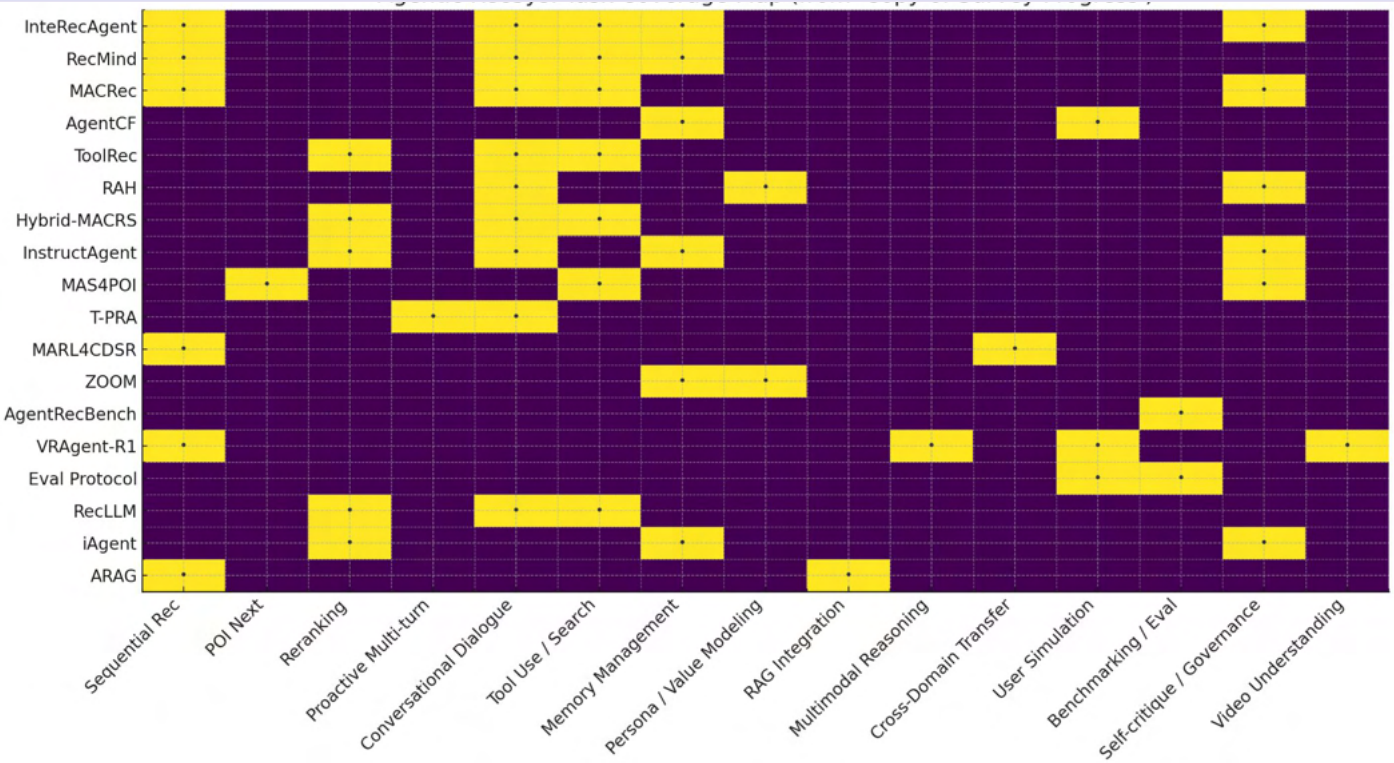
Representative models

- RecMind: Self Inspiring planning with database and search tools
- MACRec: searcher retrieves external information for explanation;
- Self Ask + Search and RAG style agents: decompose tasks and query external sources

Four Categories of Recommender Systems



Notable Examples



Example 1: InteRecAgent

A: recommender
B: single agent
C: plan-then-execute, reflection

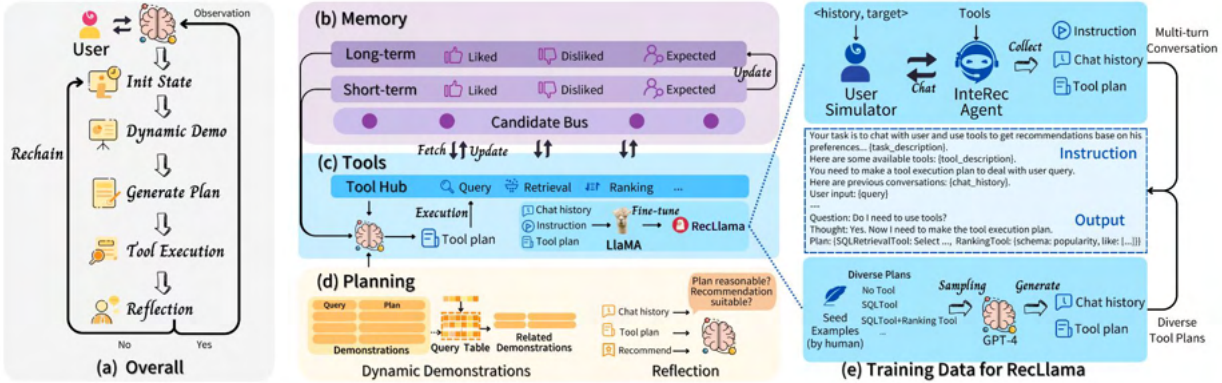


Figure 1: InteRecAgent Framework. (a) The overall pipeline of InteRecAgent; (b) The memory module, consisting of a candidate memory bus, a long-term and a short-term user profile; (c) Tool module, consisting of various tools, the plan-first execution strategy and the fine-tuning of RecLlama; (d) Planning module, involving the dynamic demonstrations and the reflection strategy; (e) Sources of fine-tuning data for RecLlama.

Example 2: MACRec

A: recommender
B: multi
C: interleaved plan and act

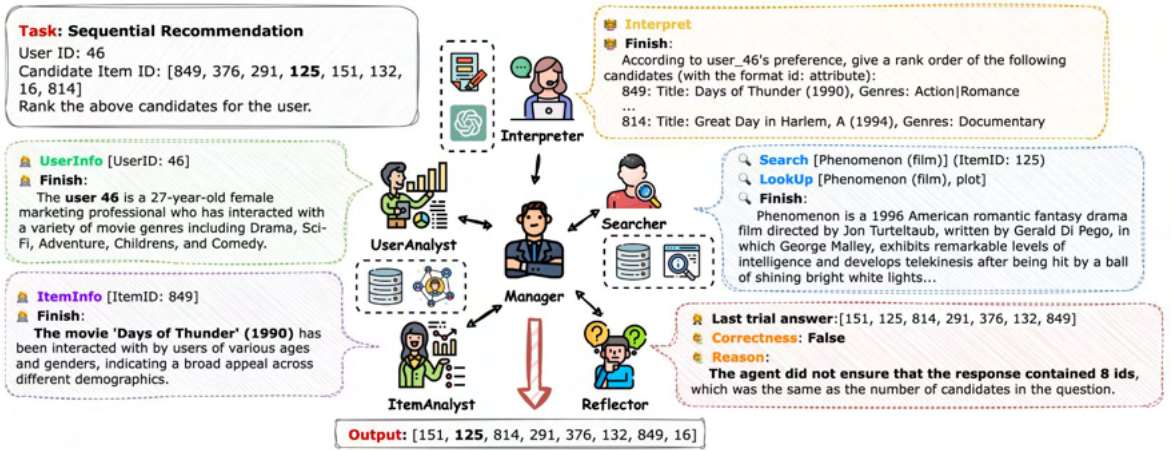


Figure 1: The Framework of MACRec. We take a sequential recommendation task as an example to show how these agents work collaboratively.

Example:3 AgentCF

A: recommender

B: multi agent

C: no plan

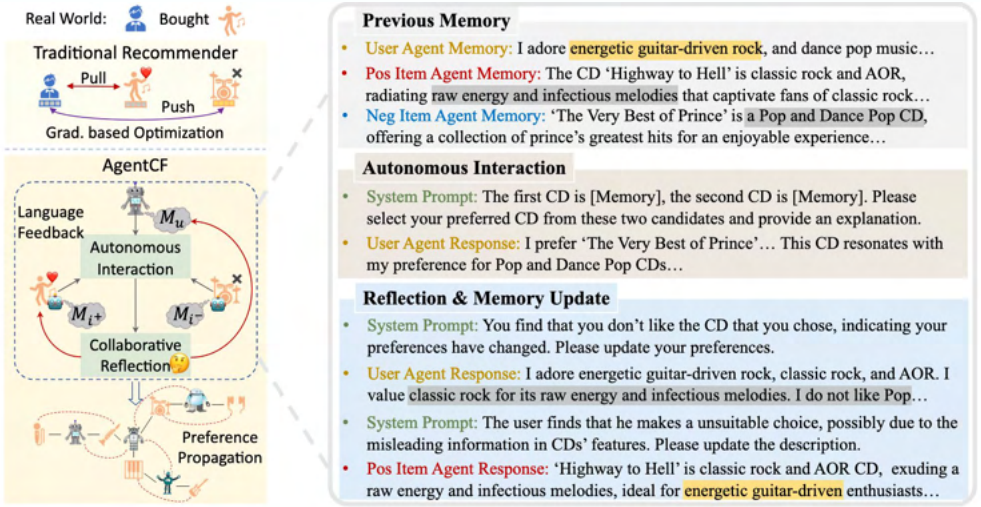


Figure 1: The overall framework of AgentCF and a case about the optimization process of agents: (1) The user and item agents are first prompted to autonomously interact. (2) These agents adjust the misconceptions in their memory, by reflecting on the disparities between their decisions and real-world interactions. In this process, the simulated preferences of user and item agents aggregate (as indicated by the highlighted content) and can propagate to other agents in subsequent interactions.

Multi-Agentic Recommender Systems: Foundations, Design Patterns, and E-Commerce Applications

An Industrial Tutorial

Reza Yousefi Maragheh[1] Yashar Deldjoo[2] Chi Wang[3] Jason Cho[1] Derek Cheng[3]

[1]Walmart Global Tech

[2]Polytechnic University of Bari

[3]Google DeepMind

RecSys 2025

Thought Process Evolution

Explain briefly Not Long!

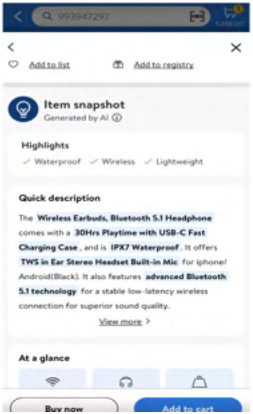
TLDR

Multi-stage modeling framework for **highlight extraction**, **Theme-Aware Keyword Extraction model** (LLM-TAKE) utilizing Large Language Models (LLMs), focused on scalability and integrated evaluation

Objective of LLM-TAKE framework

- Summarizing products through highlights that are context and theme aware, and are diverse
- **Helps customers** in their decision journey through **swiftly understanding product** characteristics
- Overcome LLM-specific **challenges**:
  - **Scalability**
  - **Evaluating the results**
  - **Hallucinations** in LLMs

Highlights



Thought Process Evolution

What's the deal with this item?

Hypothesis

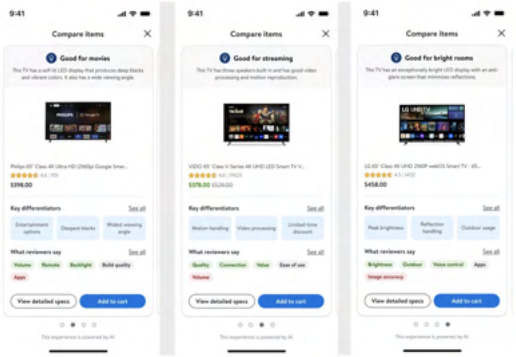
WE KNOW
Customers are looking for guidance and comparison between products they are interested in while browsing

IF WE
Show customers the most relevant use case of an item while they are comparing

THEN
We will make it easier for customers to make a decision

RESULTING IN
In an increase in the quality of ATC and higher conversion to purchases

Planned Solution



Why this is harder?

But this one is harder. Why?
This falls upon largescale AI reasoning

Question:
If I give you item features, can You give me the usecases?
You should reason your way to generation and validation of usecases

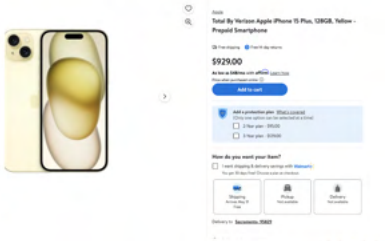
Thought Process Evolution

Myriad of Problems

The Relevance Problem → **The Phrasing Problem** → **The Non-Informativeness Problem**

Good for Professional Photography

Good for Fashion Accessory!



Good for Communication!

The question: How to deal with these edge cases in scale?

Thought Process Evolution

The Problem of Hallucination

- **Trustworthiness** of generated text by Language Models is an important element for large-scale implementations.
- Large Language Models have shown to **hallucinate** (Ji et al., 2023; Rawte et al., 2023)
- They generate responses that are **not consistent with instructions** (Duan et al., 2024).
- Thus, **evaluating the outcome of these models is crucial.**

Do LLMs Know about Hallucination? An Empirical Investigation of LLM's Hidden States

Hanyu Duan Yi Yang Kar Yan Tam
Department of Information Systems, Business Statistics and Operations Management
The Hong Kong University of Science and Technology
hduanac@connect.ust.hk {imyiyang, kytan}@ust.hk

Abstract

Large Language Models (LLMs) can make up answers that are not real, and this is known as hallucination. This research aims to see if, how, and to what extent LLMs are aware of hallucination. More specifically, we check whether and how an LLM reacts differently in its hidden states when it answers a question right versus when it hallucinates. To do this,

Maynez et al., 2020). Recently, this issue has triggered a notable discussion among thousands of AI researchers around the world, resulting in over 30,000 signatures on an open letter [1], calling for a six-month pause on "giant AI experiment" (Pause Giant, 2023).

A growing body of NLP literature has examined hallucinations in LLMs, such as looking into the source of hallucinations from a training

Classic Approaches

- **n − gram based** metrics
- **model-based** such as
  - Rouge Lin (2004),
  - BLEU Papineni et al. (2002),
  - BERTScore Zhang et al. (2019)
  - BARTScore Yuan et al. (2021)
- But have proven to show **weak correlation with human evaluations** Wang et al. (2023).

BERTSCORE: EVALUATING TEXT GENERATION WITH BERT

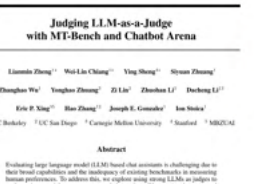
Tianyi Zhang*[1]; Varsha Kishore*[1]; Felix Wu*[1]; Kilian Q. Weinberger[1]; and Yoav Artzi[1]
[1]Department of Computer Science and [1]Cornell Tech, Cornell University
{vk352, fw245, kilian}@cornell.edu {yoav}@cs.cornell.edu
*ASAPP Inc.
tzhang@asapp.com

ABSTRACT

We propose BERTSCORE, an automatic evaluation metric for text generation. Analogously to common metrics, BERTSCORE computes a similarity score for each token in the candidate sentence with each token in the reference sentence. However, instead of exact matches, we compute token similarity using contextual embeddings. We evaluate using the outputs of 363 machine translation and image captioning systems. BERTSCORE correlates better with human judgments and

Neo-Classic Approaches

- Rapid evolution of LLMs, some researchers and practitioners have considered using LLMs for the evaluation purpose
  - Wang et al. (2023);
  - Zheng et al. (2024);
  - Maragheh et al. (2023);
  - Zhou et al. (2024).
- In this framework, due to high capability of **LLMs in mimicking human language capabilities**, they are used as a judge (Zheng et al., 2024)



Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena

Lianmin Zheng[1*] Wei-Lin Chiang[1*] Ying Sheng[2*] Siyuan Zhuang[1]
Zhanghao Wu[1] Yonghao Zhuang[3] Zi Lin[3] Zhuohan Li[1] Dacheng Li[3†]
Eric P. Xing[5†] Hao Zhang[1†] Joseph E. Gonzalez[1] Ion Stoica[1]
[1] UC Berkeley [2] UC San Diego [3] Carnegie Mellon University [4] Stanford [5] MBZUAI

Abstract

Evaluating large language model (LLM) based chat assistants is challenging due to their broad capabilities and the inadequacy of existing benchmarks in measuring human preferences. To address this, we explore using strong LLMs as judges to

Thought Process Evolution

Images Credit: **Walmart.com**

Recommendation Explanation

For reasoning tasks,

- LLM as judge don't quite work really.
- Meaning it does not work.
- Yes, it does not work on complex tasks.

Is the usecase **"professional Photography"** relevant to this item given its features? **What is a professional camera? Really?**

Chat Completion

Metrics	Understandability		Groundedness	
	ρ	κ	ρ	κ
Vanilla	0.336	0.304	0.455	0.455
CoT	0.270	0.215	0.485	0.485
Self-Refine	0.167	0.144	0.412	0.412

Recommendation Explanation

Metrics	Relevance		Phraseness			
	ρ	κ	ρ	κ	ρ_{avg}	κ_{avg}
Vanilla	0.102	0.058	0.018	0.011	0.060	0.035
CoT	0.084	0.062	0.040	0.029	0.062	0.045
Self-Refine	0.075	0.046	0.026	0.012	0.050	0.029

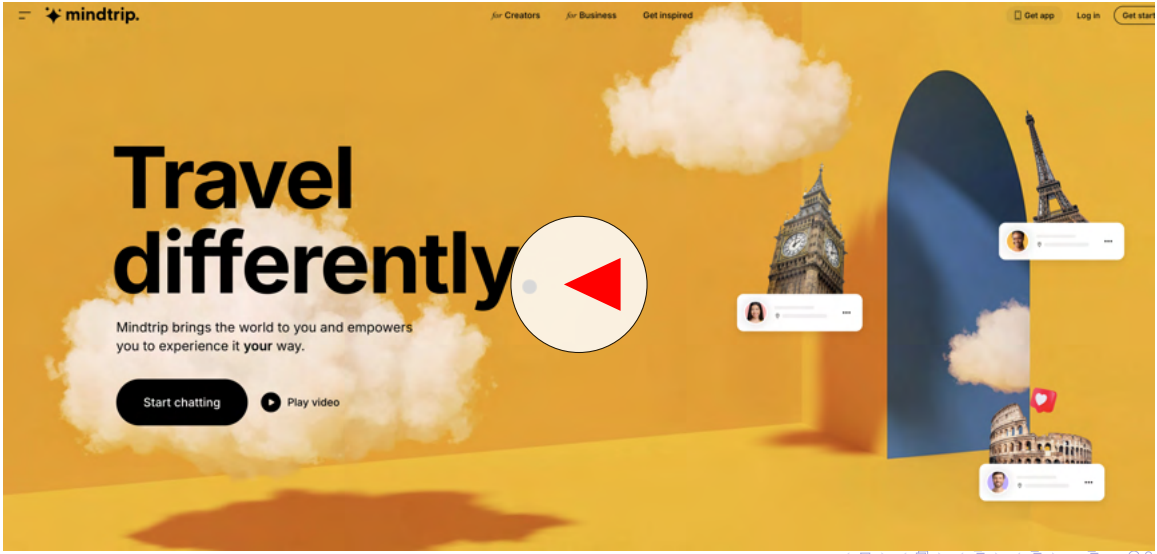
Kappa, rho <0.1 is not good at all!
Single Prompts Don't work!

Startups!

Startups

Image Credit: **dreamstime.com**

MindTrip


mindtrip.
for Creators
for Business
Get inspired
Get app
Log in
Get start
Travel differently
Mindtrip brings the world to you and empowers you to experience it your way.
Start chatting
Play video

MindTrip

Image Credit: **mindtrip.ai**

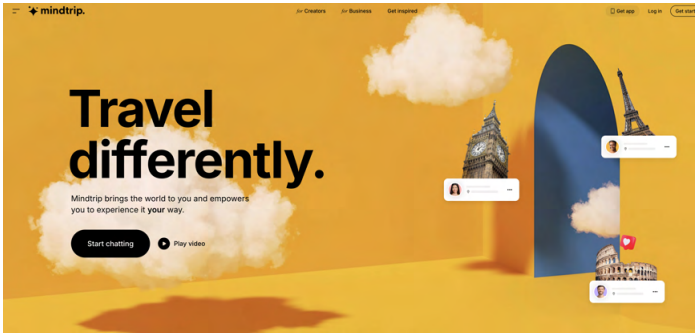
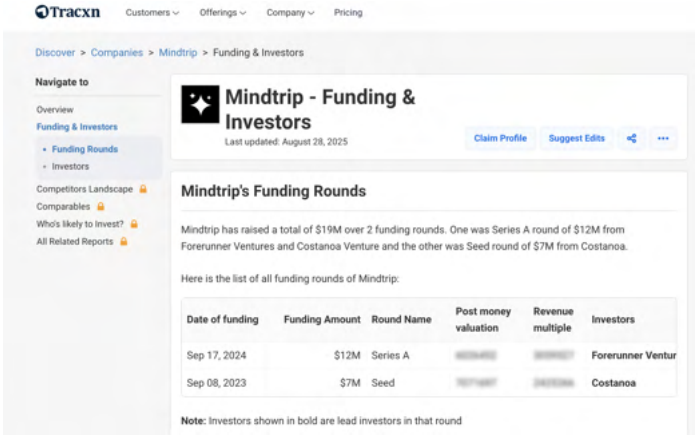


Figure: AI travel agent that recommends/optimizes itineraries, hotels, activities-Series A: Oct 2024

MindTrip

Credit: **tracxn.com**



DayDream

Image Credit: **daydream.ing**

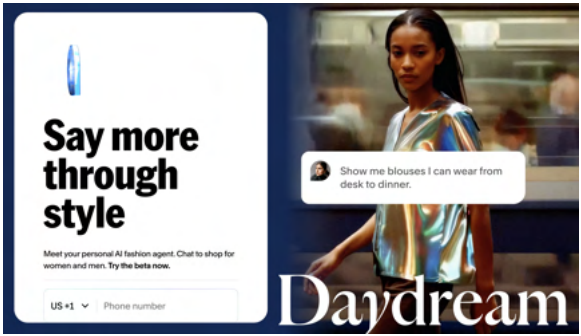
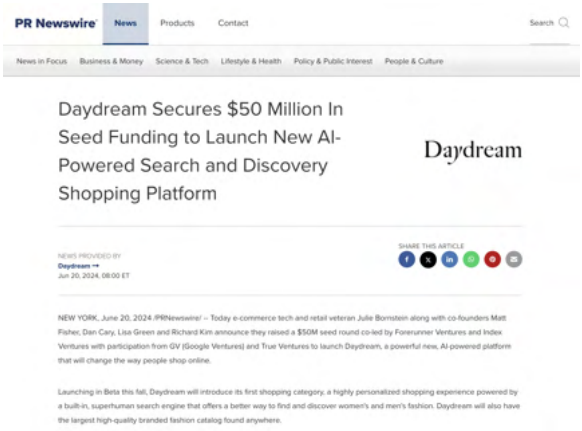


Figure: Chat-based fashion shopping agent for discovery and recommendations across thousands of brands. Seed: June 2024

DayDream

Credit: **prnewswire.com**



PR Newswire News Products Contact Search

News in Focus Business & Money Science & Tech Lifestyle & Health Policy & Public Interest People & Culture

Daydream Secures $50 Million In Seed Funding to Launch New AI-Powered Search and Discovery Shopping Platform

Daydream

NEWS PROVIDED BY
Daydream →
Jun 20, 2024, 08:00 ET

SHARE THIS ARTICLE

NEW YORK, June 20, 2024 /PRNewswire/ -- Today e-commerce tech and retail veteran Julie Bornstein along with co-founders Matt Fisher, Dan Cary, Lisa Green and Richard Kim announce they raised a $50M seed round co-led by Forerunner Ventures and Index Ventures with participation from GV (Google Ventures) and True Ventures to launch Daydream, a powerful new, AI-powered platform that will change the way people shop online.

Launching in Beta this fall, Daydream will introduce its first shopping category, a highly personalized shopping experience powered by a built-in, superhuman search engine that offers a better way to find and discover women's and men's fashion. Daydream will also have the largest high-quality branded fashion catalog found anywhere.

ALTA

Image Credit: **altadaily.com**

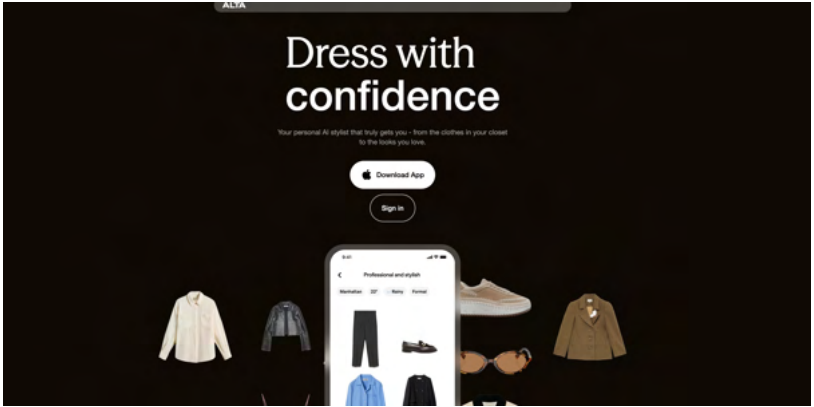
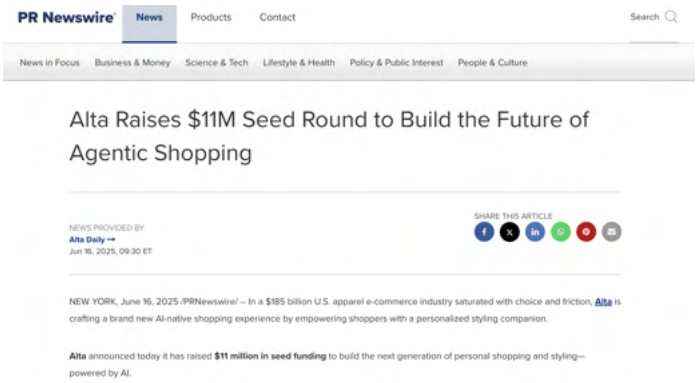


Figure: Agentic AI personal stylist that plans outfits and recommends products. Seed: June 2025

ALTA

Credit: **prnewswire.com**



Aampe

Image Credit: **aampe.com**

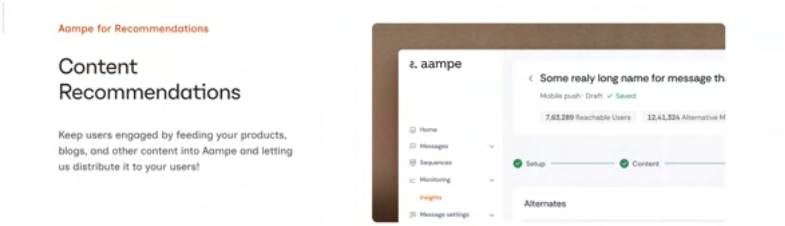
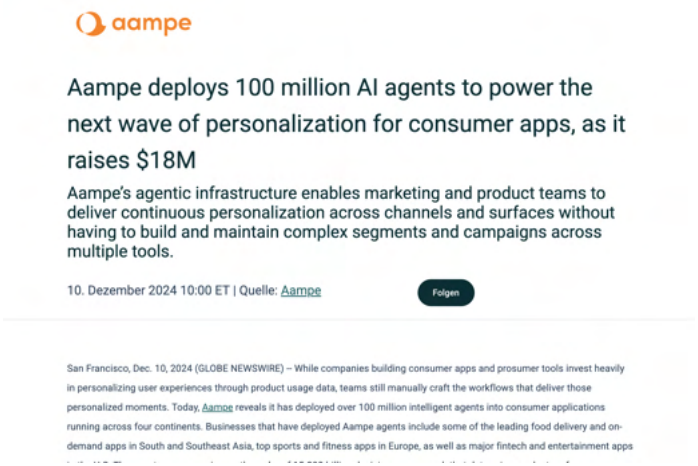


Figure: Aampe's Agentic AI learns what works for each customer. Then it instantly adapts your messaging and delivers at optimal times to drive better engagement, growth and unlock valuable insights. Series A-DEc 2024

Aampe

Credit: **globenewswire.com**–All: 27.3M



aampe

Aampe deploys 100 million AI agents to power the next wave of personalization for consumer apps, as it raises $18M

Aampe's agentic infrastructure enables marketing and product teams to deliver continuous personalization across channels and surfaces without having to build and maintain complex segments and campaigns across multiple tools.

10. Dezember 2024 10:00 ET | Quelle: Aampe

Folgen

San Francisco, Dec. 10, 2024 (GLOBE NEWSWIRE) -- While companies building consumer apps and prosumer tools invest heavily in personalizing user experiences through product usage data, teams still manually craft the workflows that deliver those personalized moments. Today, Aampe reveals it has deployed over 100 million intelligent agents into consumer applications running across four continents. Businesses that have deployed Aampe agents include some of the leading food delivery and on-demand apps in South and Southeast Asia, top sports and fitness apps in Europe, as well as major fintech and entertainment apps

Constructor

Image Credit: **constructor.com**

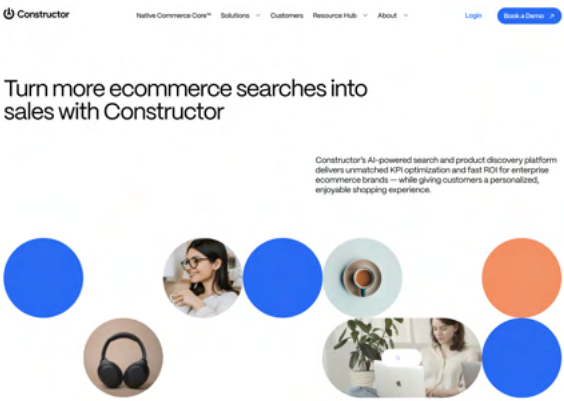
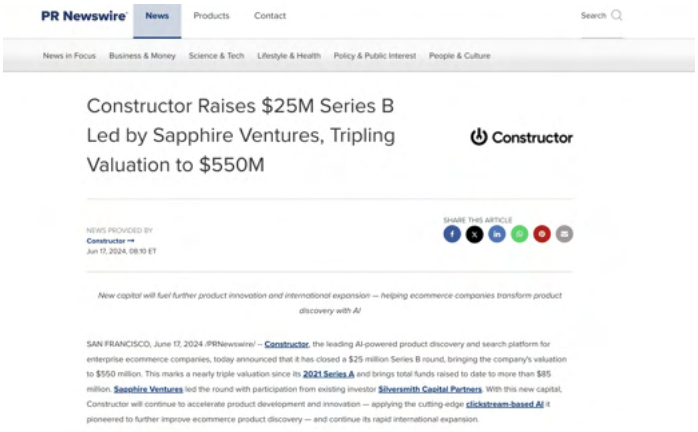


Figure: AI Shopping Agent + search & recommendations for enterprise ecommerce. Series B: June 2024

Constructor

Credit: **prnewswire.com**-all funds: 85M



PR Newswire News Products Contact Search

News in Focus Business & Money Science & Tech Lifestyle & Health Policy & Public Interest People & Culture

Constructor Raises $25M Series B Led by Sapphire Ventures, Tripling Valuation to $550M

Constructor

NEWS PROVIDED BY
Constructor
Jun 17, 2024, 08:10 ET

SHARE THIS ARTICLE

New capital will fuel further product innovation and international expansion — helping ecommerce companies transform product discovery with AI

SAN FRANCISCO, June 17, 2024 /PRNewswire/ -- **Constructor**, the leading AI-powered product discovery and search platform for enterprise ecommerce companies, today announced that it has closed a $25 million Series B round, bringing the company's valuation to $550 million. This marks a nearly triple valuation since its **2021 Series A** and brings total funds raised to date to more than $85 million. **Sapphire Ventures** led the round with participation from existing investor **Silversmith Capital Partners**. With this new capital, Constructor will continue to accelerate product development and innovation — applying the cutting-edge **clickstream-based AI** it pioneered to further improve ecommerce product discovery — and continue its rapid international expansion.

Why so much $$$?

What is an agent anyway?

Image Credit: Quite Like You! Andy Shauf, youtube.com

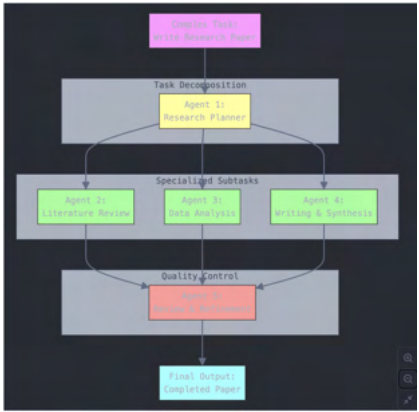
"I'll call "Society of Mind" this scheme in which each mind is made of many smaller processes. These we'll call agents. Each mental agent by itself can only do some simple thing that needs no mind or thought at all. Yet when we join these agents in societies-in certain very special ways-this leads to true intelligence".

From: **The Society of Mind, Marvin Minsky**

Why Multi-Agents?

Task Chunking

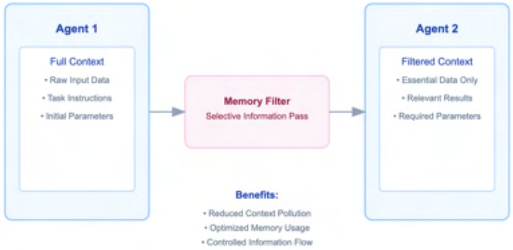
- **Reducing Reasoning Load**:Complex tasks often involve multiple interdependent steps that can overwhelm a single agent's decision-making capacity. Breaking these down into smaller, manageable chunks.
- **Failure Isolation**: Failures can be isolated to specific subtasks rather than requiring a complete restart of the entire process.
- **Specialized Agents**: Different agents can be optimized for different types of subtasks.



Why Multi-Agents?

Memory Moderation

- **Reducing Context Pollution**: Controlled information flow between agents prevents context pollution and reduces the likelihood of conflicting or irrelevant information affecting decision-making processes.
- **Reducing Computational Overhead**: Selective memory passing allows for efficient resource utilization by only sharing essential information.



Why Multi-Agents?

Tool Calling

- **Beyond Language Processing**: Agents can be designed to interface with external tools like databases, APIs, or specialized software, expanding their capabilities beyond pure language processing.
- **Less expensive compute**: Task-specific tools can handle computationally intensive operations (like complex mathematical calculations or data processing) more efficiently than language models alone.

Why Multi-Agents?

We practice what we present:

- Past three slides (an only past three of them) are researched, written, drawn, evaluated, corrected by an agentic pipeline!

Gamma

Formalism

Formal Definition: LLM Agent

Definition

An **LLM Agent** is an intelligent system whose core decision-making and interaction capabilities are powered by one or more large language models.

$$A_{\text{LLM}} = \Big(\mathcal{M}, \mathcal{I}, \mathcal{O}, \mathcal{F}, \Omega\Big)$$

Formal Definition: LLM Agent

- $\mathcal{M}$: base model(s) for *understanding, reasoning, generation* (e.g., instruction-tuned transformer; optionally a mixture-of-experts).
- $\mathcal{I}$: *input space* observable by the agent (user text, images, traces, features, tool responses).
- $\mathcal{O}$: *output space* producible by the agent (natural language, structured JSON, function/tool calls, actions).
- $\mathcal{F} = \{f_k\}_{k=1}^{K}$: *external tools/APIs* the agent may invoke (retrieval, DB lookup, calculators, vision encoders, planners).
- Ω: *state & memory* enabling persistence across turns/sessions (short-term/working, episodic, semantic, procedural).

Multi-Agent System (MAS): Formal Definition

Definition

A **Multi-Agent System** is an ordered triple

$$\mathrm{MAS} = \Big(\mathcal{A}, \mathcal{E}, \Pi\Big),$$

where:

- $\mathcal{A} = \{A_1, \ldots, A_n\}$: finite set of agents; each A_i may be an LLM agent or another modular service.
- $\mathcal{E}$: shared *environment* exposing percepts/resources (APIs, UIs, simulators). Formally, a partially observable state space from which each agent receives observations and executes actions.
- $\Pi = (\mathbf{C}, \Gamma)$: *interaction protocol* constraining who may talk to whom and how.

Execution view: a run of MAS is a sequence of environment states and inter-agent messages that respect Π.

Interaction Protocol $\Pi = (\mathbf{C}, \Gamma)$

Communication matrix.

$$\mathbf{C} \in \{0,1\}^{n \times n}, \quad \mathbf{C}_{ij} = 1 \iff \text{messages of type } \gamma \in \Gamma \text{ are permitted from } A_i \to A_j.$$

- **Preset (static) routing:** $\mathbf{C}$ fixed at design time.
- **Autonomous (dynamic) routing:** channels toggled by guard functions $g_{ij} : \text{state} \to \{0,1\}$, e.g. $\mathbf{C}_{ij}(t) = g_{ij}(\mathcal{E}_t, \Omega_t^i, \Omega_t^j)$.

Message schemata Γ.

- *Performatives* (semantics): `INFORM`, `REQUEST`, `ACCEPT`, `REJECT`, ...
- *Serialization rules* (syntax): JSON schema, field names/types for machine readability.
- *Timing constraints*: ordering, deadlines, rate limits, retry/timeout policies.

Each $\gamma \in \Gamma$ defines both **syntax** and **semantics** so a receiver can parse and act on messages deterministically.

Minimal Example (Rec $\leftrightarrow$ Eval)

Agents: $\mathcal{A} = \{A_{\text{rec}}, A_{\text{eval}}\}$.

Environment:

$$\mathcal{E} = \big(\text{UserProfileDB},\ \text{ProductCatalogue},\ \text{BusinessRulesKB}\big).$$

Protocol:

$$\Pi = (\mathbf{C}, \Gamma), \quad \mathbf{C} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \Gamma = \{\texttt{candidate\_list},\ \texttt{compliance\_report}\}.$$

Message flow.

- $A_{\text{rec}} \to A_{\text{eval}}$: `candidate_list` (ranked JSON array $\{(s_1, \text{score}_1), \ldots, (s_L, \text{score}_L)\}$).
- $A_{\text{eval}} \to A_{\text{rec}}$: `compliance_report` (violations w.r.t. brand/policy $\in$ BusinessRulesKB).

Off-diagonal ones in $\mathbf{C}$ permit bidirectional messaging; zeros on the diagonal disallow self-messaging.

Dynamic Channels & Termination Conditions

Autonomous moderation of links.

$$\mathbf{C}_{\text{eval,rec}}(t) = \begin{cases} 1, & \text{NOT Halt}(t) \\ 0, & \text{Halt}(t) \end{cases} \quad \text{with} \quad \text{Halt}(t) \iff \big(\text{violations} = \varnothing\big) \vee (t{-}t_0 > \text{timeout}).$$

Protocol-level guards in Γ:

- *Stop condition*: end dialogue when all items pass compliance.
- *Retry/backoff*: bounded retries on transient failures.
- *Escalation*: on persistent violation, route to A_{policy} or human-in-the-loop.

Takeaway. Π can be preset yet *autonomously reconfigured* according to system state which balances accuracy with efficiency.

Let's Recap

Industrial Agentic RecSys-Usecases

Industrial Agentic RecSys-Usecases

Planning → Simulation → Multi-modal → Explanation

Scenarios we will walk through

- **Interactive conversational Recommendation**
  - sub-agents coordinate (planner, retrieval, constraints) → curated "Mickey-Mouse party" bundles.
- **User-simulation-Reco Eval**:
  - synthetic users + logging/summarization → stress-test policies pre-deployment.
- **Contextual & multi-modal Recommendation**:
  - vision+text (room photo → Boho set) → complementarity & aesthetic coherence.
- **Recommendation Explanation**:
  - brand-consistent narratives that expose latent ranking logic ("inspired by your Disney purchases").

Interactive Recommendation

Interactive Recommendation

Agentic conversational planning for context-aware bundles

- Multi-turn conversation ⇒ refine constraints & intent
- Sub-agents for retrieval, validation, ranking, layout
- Memory-aware (STM/SEM/EPI/PROC) & tool-using

Interactive Recommendation

Task at a Glance

Definition (informal)

Interactive Recommendation engages in a **multi-turn** dialogue to identify, refine, and present suitable items, adaptively incorporating user feedback, contextual constraints, and memory/tool signals.

Why interactive?

- Real-world user intents are **open-ended** and **multi-step**
- Preferences emerge through **clarification** and **trade-offs**

Running example

"Mickey-Mouse themed birthday": decorations, gluten-free cake, favors; one conversation, many sub-goals.

Let $\mathcal{D}$ be conversation turns and $\mathcal{S}$ the item universe (SKUs). d_i and a_i are user and agent's utterances. A session at step t has transcript

$$\mathcal{C}_{1:t} = (d_1, a_1, \ldots, d_{t-1}, a_{t-1}, d_t), \quad d_i, a_i \in \mathcal{D}.$$

Define the interactive recommendation mapping

$$\Phi : \big(\mathcal{C}_{1:t},\, \mathcal{E}\big) \to \langle s_{(1)}, \ldots, s_{(L)} \rangle, \quad s_{(j)} \in \mathcal{S},$$

where, $\mathcal{E}$ is the shared space among agents. Φ outputs a ranked list of length L.

Objective with implicit constraints

$$\langle s_{(1)}, \ldots, s_{(L)} \rangle = \arg \max_{\langle s_1, \ldots, s_L \rangle} \sum_{j=1}^{L} \mathrm{Rel}\big(s_j \mid \mathcal{C}_{1:t}\big)$$

subject to constraints (theme, dietary, budget) encoded in $\mathcal{C}_{1:t}$ and available tools/environment $\mathcal{E}$.

- $\mathrm{Rel}(\cdot)$ may incorporate user history (SEM/EPI) and session cues (STM)
- Feasibility checked via tools (e.g., inventory, price, layout fit)

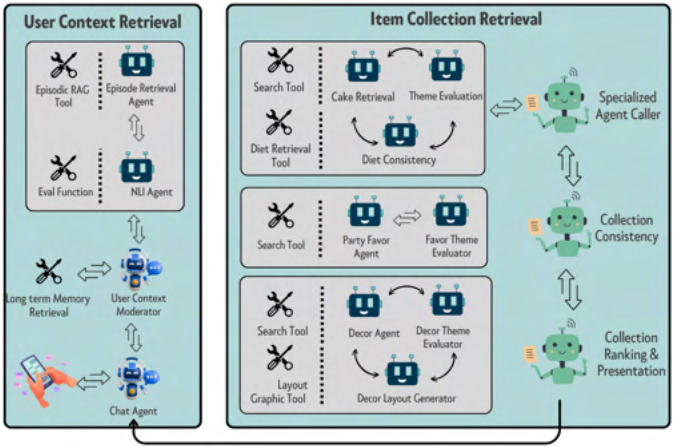
Objective

Dynamically adapt the Retrival/Ranking function Φ across turns to minimise user effort while satisfying latent sub-goals (e.g., *gluten-free chocolate cake*, *palette-consistent decor*).

- Spawn specialised sub-agents on demand (category retrievers, validators)
- Maintain **coherence** across turns via short/long-term memory
- Present **holistic** bundles (ranked collection + layout preview)

Interactive Recommendation

The Visual



Agent set $\mathcal{A}$

A_{chat} (chat), A_{epi} (episodic retrieval $\mathcal{Q}$), A_{nli} (episode vetting), A_{SAC} (specialised-agent caller),
A_{cake}, A_{decor}, A_{favor} (category retrieval), $A_{\text{col\_check}}$ (collection consistency), A_{rank} (ranking/presentation).

Environment & Protocol

$\mathcal{E} = \big(\text{ProductCatalogue}, \text{UserProfileDB}, \text{VectorDB}\big)$;
$\Pi = (\mathbf{C}, \Gamma)$ with $\mathbf{C}_{ij}{=}1$ iff A_j is spawned by A_i or $A_i{=}A_{\text{chat}}$;
$\Gamma{=}\{\texttt{query}, \texttt{episode\_list}, \texttt{tool\_call}, \texttt{item\_set}, \texttt{ranked\_list}\}$.

Interactive Recommendation

Message Flow & Orchestration

$$A_{\text{chat}} \xrightarrow{\texttt{query}} A_{\text{epi}} \xrightarrow{\texttt{episode\_list}} A_{\text{nli}} \xrightarrow{\texttt{validated\_episodes}} A_{\text{SAC}} \xrightarrow{\texttt{spawn}} \{A_{\text{cake}}, A_{\text{decor}}, A_{\text{favor}}\}$$
$$\xrightarrow{\texttt{item\_set}} A_{\text{col\_check}} \xrightarrow{\texttt{validated\_set}} A_{\text{rank}} \xrightarrow{\texttt{ranked\_list}} A_{\text{chat}}.$$

- A_{SAC} spawns micro-MAS blocks per category
- A_{nli} and $A_{\text{col\_check}}$ act as *gates* (reduce hallucinations, enforce theme/diet)
- A_{rank} personalises final presentation; `LayoutTool` renders a board

Combination of workflow and autonomy: A_{SAC} determines which specialized agents to call.

Stores

- **STM** (working): A_{chat} keeps last L turns of $\mathcal{C}_{1:t}$
- **EPI**: A_{epi} manages episode store via $\mathcal{U}/\mathcal{Q}$
- **SEM**: stable traits (colors, dietary rules, budget)
  - This may have its update pipeline.
- **PROC**: reusable prompts/templates, DB schema for autonomous tool use and DB read.
  - What are the internal tables and when should the agent call them?

Retrieve Retrieval Function $\mathcal{Q}$ for sub-goal τ of cake retrieval:

$$\begin{aligned}\widehat{\mathcal{C}} &= \mathcal{Q}(\Omega_t, \tau) \\ &= \{\text{Flavor\_pref: chocolate,} \\ &\quad \text{query: “Mickey-Mouse Cake”}\}.\end{aligned}$$

Feeds `SearchCakeAPI` with gluten-free, chocolate filter.

Update Update function for memory $\mathcal{U}$:
User: “gluten-free restriction.”

- Episode Distillation:

$$\widetilde{\mathcal{C}}_t = \text{“}\{\text{allergy: gluten}\}\text{”}$$

- Episodic Memory Update:

$$\Omega_{t+1}^{\text{EPI}} = \Omega_t^{\text{EPI}} \cup \{\text{allergy: gluten}\}.$$

Interactive Recommendation

Tooling

- Tools $\mathcal{F}$: `SpecializedSearchAPI`, (graphic board of decor set).
- Sparse $\mathbf{C}$ limits communication.
- Observability: per-agent logs for `query`/`tool_call`/`ranked_list`
- Timeouts & fallbacks:
  - Control on hit rate,
  - Failure back up

Interactive Recommendation

Discussion: Benefits in Production

- **Modularity**: category blocks are micro-MAS units, reusable across themes
- **Memory-aware personalisation**: injects user- and session-specific constraints , SEM/EPI yield relevant recommendation.
- **Error containment**: NLI & collection checks act as fact gates
- **Autonomy**: specialised-agent caller spawns workers on demand.
- **Richer User Experience**: interactive, using tools like `LayoutTool` offers better experience than a static recommendation.

Agentic Recommendation Evaluation

Task Definition

Simulation-based evaluation (informal)

Generate **synthetic user behavior** to stress-test recommendation policies *before* live deployment.

Core mappings

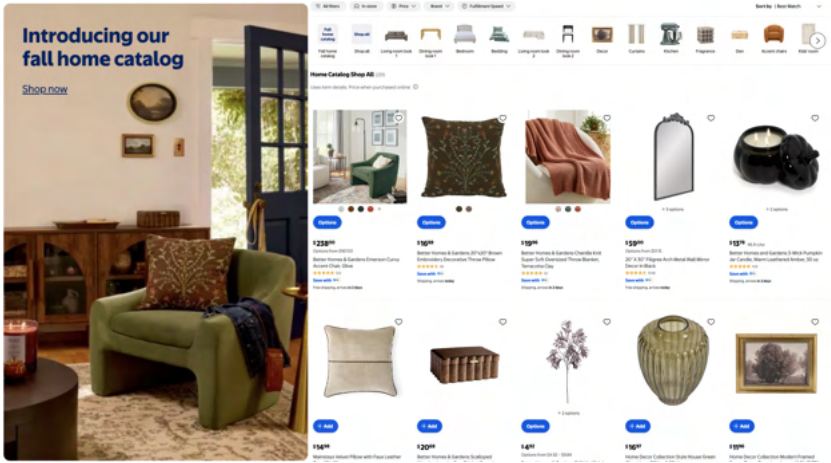
$$R_\phi : \mathcal{X} \to \mathcal{S}^L \quad \text{(recommender maps state } x \text{ to } L \text{ items)}$$

$$\mathcal{M}_{\theta,\omega} : \mathcal{X} \times \mathcal{S}^L \to \mathcal{A}_{\text{user}} \quad \text{(user simulator maps } (x, \text{list}) \text{ to an action)}$$

$\mathcal{A}_{\text{user}} \in \{\text{Select}, \text{Not Select}\}$ or richer (e.g. $\{$Click,Pass,Purchase$\}$).

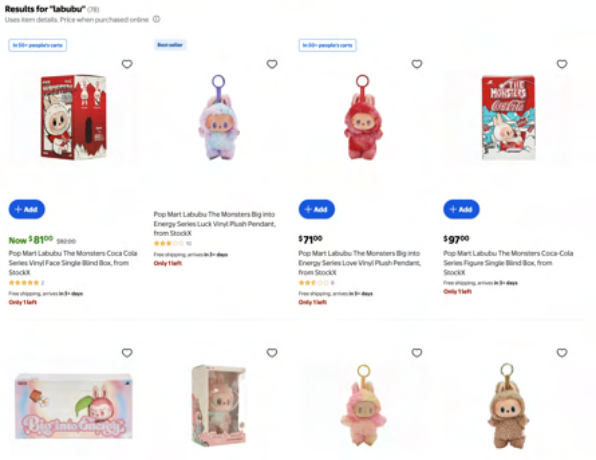
Agentic Recommendation Evaluation

Fall Seasonal Recommendation



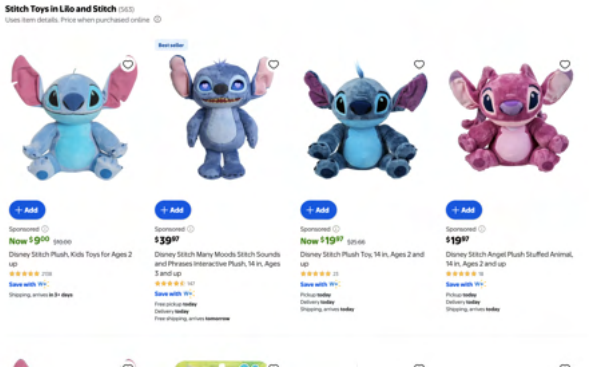
User Simulation

What is a new trend?



User Simulation

What is a new trend?



High-Level Goals (I)

- **De-risk A/B:** approximate user responses when real traffic is scarce or expensive
  - Sometimes, duration of the recommendation being live is short-lived (no time for AB),
  - Example: Seasonal Recommendation.
- **Fast iteration:** offline loops to evaluate design choices, ranking features, and guardrails
  - AB tests by design are slower (specially if we wait for stat sig results on niche general merchandise items)
- **Coverage:** explore edge cases (long-tail preferences, rare journeys) at scale
  - Hard to perform AB test on user cohorts,
  - Example: all new users with account opening in the last year.
- **Safety:** detect regressions, policy violations, and mode collapse before serving
  - Better control on the edge cases because we can simulated edge cases at scale.

High-Level Goals (II)

- **Measurement:** CTR/CVR, diversity, novelty, calibration, fairness, robustness
  - Should optimize on the above metrics,
  - Hard to track for customer cohorts.
- **Session granularity:** short (*quick browse*) vs. long (*deliberative*)
  - In each AB test we want to analyse short term effect and long term effect of recommendation,
  - Short term: purchase within session,
  - Longer term: product return patterns,
  - Longerer term: User loyalty.
- **Population realism:** sample (θ, ω) to reflect heterogeneous users
  - θ: can represent user profiles.
  - ω: we can add controlled noise to better simulate user variability withing cohorts.
- **Explainability:** log rationales and session summaries for root-cause analysis

Formal Definition (I)

In Recommendation Evaluation and User simulation task

- We define a user/platform state $\mathcal{X}$,
- Recommend $\mathcal{S}^L$,
- Observe user behavior $\mathcal{A}_{\text{user}}$.

$$R_\phi : \mathcal{X} \to \mathcal{S}^L, \qquad x_t \mapsto s_t^{(L)}$$
$$\mathcal{M}_{\theta,\omega} : \mathcal{X} \times \mathcal{S}^L \to \mathcal{A}_{\text{user}}, \qquad (x_t, s_t^{(L)}) \mapsto a_t^{\text{user}}$$

Iterated for $t = 1, \ldots, T$ to generate trajectories $\{x_t, s_t^{(L)}, a_t^{\text{user}}\}_{t=1}^T$.

Formal Definition (II)

Task objective

Task objective is to estimate any relevant metrics for user interaction,

$$\Psi(R_\phi, \mathcal{M}_{\theta,\omega}) = \mathbb{E}\Big[\sum_{t=1}^{T} g\big(x_t, s_t^{(L)}, a_t^{\text{user}}\big)\Big],$$

where g can encode Select/Purchase reward, diversity/penalty, fairness, etc.

- Estimate Ψ under controlled distributions of (θ, ω)
- Simulate and estimate results via Monte Carlo simulation.

Formal Definition (III): Population & Estimation

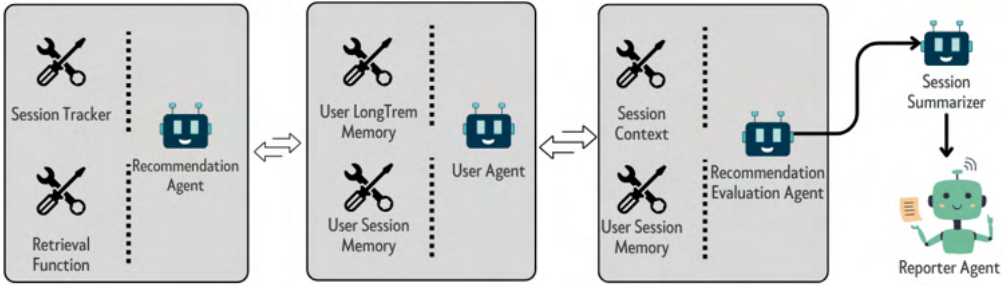
- **Population:** $(\theta, \omega) \sim P$ (PreferenceSampler)
- **Monte Carlo estimator:**

$$\widehat{\Psi} = \frac{1}{N} \sum_{n=1}^{N} \sum_{t=1}^{T} g\big(x_t^{(n)}, s_t^{(L,n)}, a_t^{\mathsf{user}(n)}\big)$$

- **Risk-aware variants:** Mearure metrics like
  - CVaR@ k,
  - worst-case across cohorts,
  - per-segment fairness gaps.

Architecture

The Visual



System Architecture: Agents

$\mathrm{MAS}_{\mathsf{sim}} = (\mathcal{A}, \mathcal{E}, \Pi)$

$$\mathcal{A} = \{A_{\mathsf{rec}}, A_{\mathsf{user}}, A_{\mathsf{note}}, A_{\mathsf{eval}}, A_{\mathsf{summ}}, A_{\mathsf{report}}\}$$

- A_{rec}: wraps R_ϕ; calls `SearchAPI`, `SessionTracker`
- A_{user}: implements $\mathcal{M}_{\theta,\omega}$ with structured memory
- A_{note}: durable logging of $(x_t, s_t^{(L)}, a_t^{\mathsf{user}})$
- A_{eval}: computes g_t; emits `eval_event`
- A_{summ}: *compresses* sessions via $\mathcal{R}_{\mathsf{sum}}$
- A_{report}: aggregates, tests, produces PDF/CSV dashboards

Environment & Protocol

Environment

$\mathcal{E} = (\text{ProductDB},\ \text{PreferenceSampler},\ \text{LogStore})$

Protocol

$$A_{\text{rec}} \rightleftarrows A_{\text{user}} \to A_{\text{eval}} \to A_{\text{summ}} \to A_{\text{report}},$$

$$\Gamma = \{\texttt{rec\_list}, \texttt{user\_action}, \texttt{log\_entry}, \texttt{eval\_event}, \texttt{session\_summary}, \texttt{final\_report}\}.$$

$\mathbf{C}$ permits minimal paths to reduce chatter; logs are append-only.

End-to-End Flow

1. A_{rec}: produce $s_t^{(L)}$ for state x_t
2. A_{user}: act $a_t^{\text{user}} \sim \mathcal{M}_{\theta,\omega}(x_t, s_t^{(L)})$
3. A_{note}: log $(x_t, s_t^{(L)}, a_t^{\text{user}})$
4. A_{eval}: compute g_t and `eval_event`
5. A_{summ}: periodic $\mathcal{R}_{\text{sum}}$ summaries
6. A_{report}: aggregate $\Rightarrow$ KPIs, cohort analyses

Memory Requirements (I)

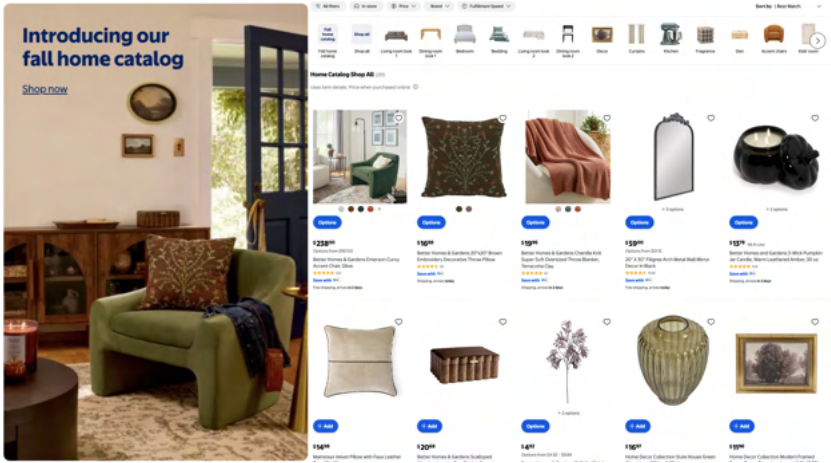
- A_{user}:
  - Ω^{SEM} (latent taste vector, price sensitivity, stable user features)
  - Ω^{STM} (current trajectory)
  - Ω^{EPI} (prior sessions / exposure effects)
  - Ω^{PROC} (navigation/click heuristics)
    - Encodes how the user proceeds in a session
- A_{eval}:
  - Raw text log + vector store for fast $\mathcal{Q}$ by item/action
  - Sliding-window stats (CTR, entropy) in STM

Tools & Data Access (II)

- `SearchAPI`, `ProductDB.query` for candidate context
- `PreferenceSampler` for (θ, ω) populations
- `LogStore.append`/`LogStore.scan` for $\mathcal{U}/\mathcal{Q}$
- Offline analytics: aggregation, stratified metrics, significance tests

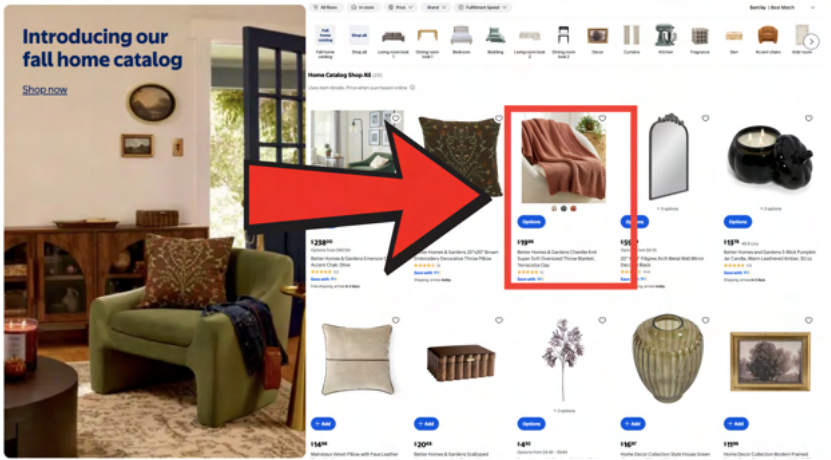
User Simulation

Fall Seasonal Recommendation



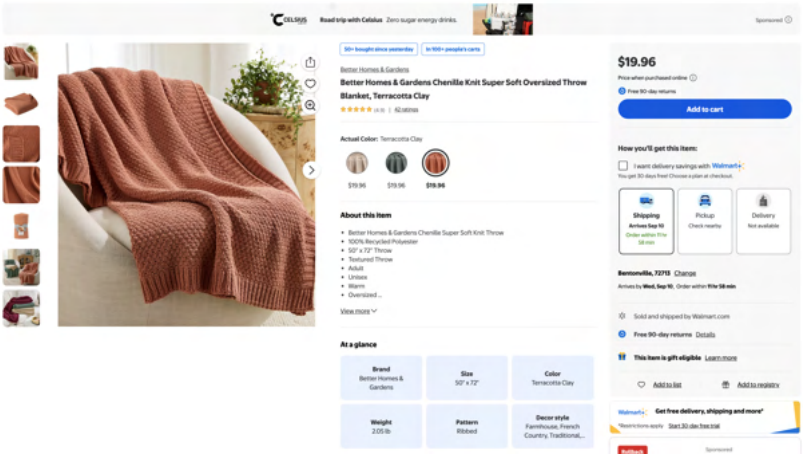
User Simulation

Fall Seasonal Recommendation



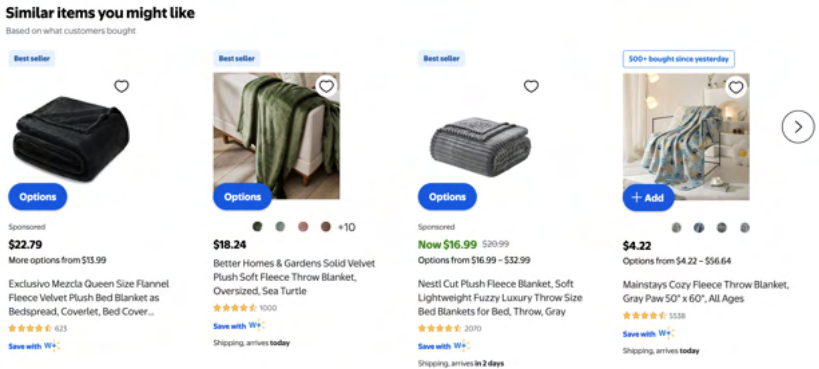
User Simulation

Fall Seasonal Recommendation



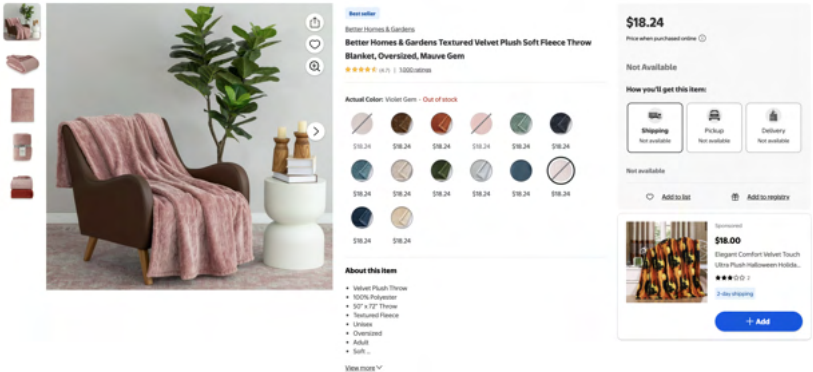
User Simulation

Scroll Down-Click on second item thrid color



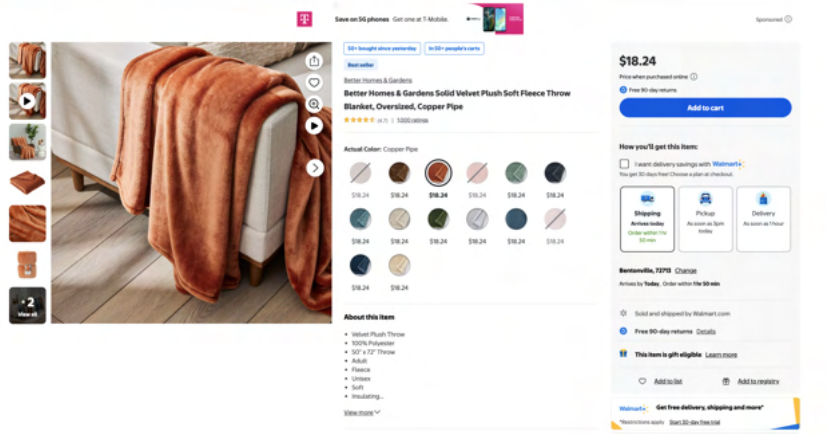
User Simulation

Color Out of Stock!



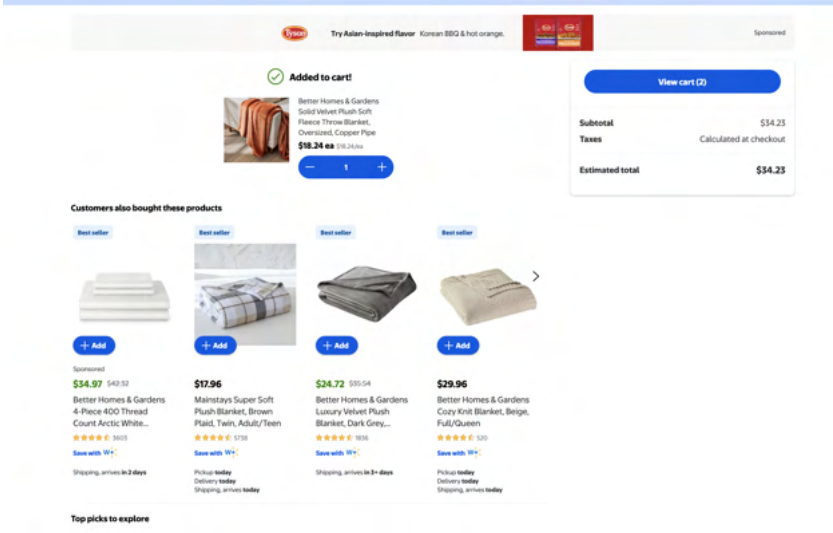
User Simulation

Add to correct the color you like



User Simulation

Added to cart-Rest of the Journey

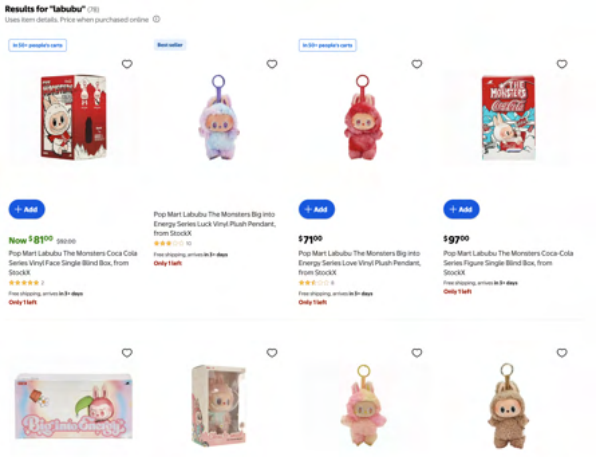


Benefits (I)

- **Cost-effective experimentation:** run $N \gg 1$ users in parallel; compare ϕ variants safely
  - Can study cohort groups better,
  - more granuality.
- **Cold-start mitigation:** sample sparse/unseen θ to probe failure modes early
  - new trends,
  - new items,
  - new users
- **Repeatability:** deterministic seeds $\Rightarrow$ reproducible KPI deltas
  - With different language models,
  - Same LLM to analyze variations in user journey trajectories

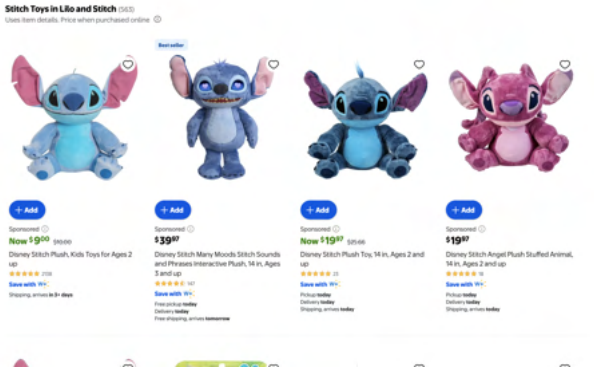
User Simulation

What is a new trend?



User Simulation

What is a new trend?



Benefits (II)

- **Bias diagnosis:** aggregate trends (price sensitivity, popularity bias, demographic skew)
  - Informed edge case handling,
  - Informed recommendation algorithm design.
- **Memory-driven realism:** SEM/EPI/PROC capture carry-over and bounded user attention
  - What is the effect on other recommendation algorithm?
  - what is the effect on long term user loyalty?
  - what is the effect of previous purchases in current user session behavior?
- **Modular extensibility:** add fairness/robustness evaluators without retraining R_ϕ
  - Custom metric evaluation.

Contextual & Multi-Modal Recommendation

fill the image with Boho style furniture

Task Definition

Contextual & multi-modal recommendation (Informal)

Recommend concept-oriented, while adapting to different modalities.

Contextual & multi-modal recommendation (formal)

Let $\mathbf{x} \in \mathcal{T}$ be textual query (e.g. "Bohemian, earthy colors"), $\mathbf{v} \in \mathcal{V}^K$ a set of visuals, and $\mathbf{u} \in \mathcal{N}$ a long-term user profile. We model

$$\mathcal{R}_\phi : \ (\mathbf{x}, \mathbf{v}, \mathbf{u}) \ \longrightarrow \ \widehat{\mathbf{y}} = \langle s^1, s^2, \dots \rangle, \quad s^i \in \mathcal{S},$$

yielding a ranked set subject to aesthetic coherence & complementarity.

High-Level Goals (I)

- **Holistic curation:** move beyond single-item retrieval to a *cohesive* set
  - Focus can be on curation of non-similar bundles
  - Topic-oriented Recommendation,
  - Aesthetic Complementarity.
- **Context grounding:** leverage spatial cues (layout, lighting) and palette from images. Ground Concepts:
  - What does “Boho” Style exactly mean?
  - What does “Industrial” house furniture exactly mean?
- **Personalisation:** incorporate long-term preferences & budget constraints
  - Be aware of user long term features,
  - User’s behavior in similar sessions,
  - Other similar users behavior
- **Effort reduction:** “one-shot designer” experience vs. piecemeal searching
  - Reduce the time and effort of collection building by user.

High-Level Goals (II)

- **Brand/style awareness:** enforce on-brand style guides and user's declared themes
  - The model should be aware of high level queries
  - The model should be aware of brand to style relations
- **Consistency at scale:** ensure cross-item compatibility (materials, colors, sizes)
  - This is hard,
  - constructing cohesive bundles: a lot of edge cases,
- **Interactive refinement:** accept quick corrections
  - Design feature of this task
- **Ready-to-buy boards:** present ranked set with room mock-ups for fast decisions
  - Make life easier for the user.

How to implememt?

Sample pseudo-algo

Goal: turn image + text + history into a coherent, feasible, on-budget bundle.

Key primitives in this module:

1. **Palette vector** $\mathbf{p}$: compact color/material signature of the image.
2. **Layout affordance** ℓ: spatial constraints (free space, obstacles, anchors).
3. **Compatibility kernel** $\kappa(\cdot, \cdot; \mathbf{p})$: pairwise harmony in a bundle.
4. **Total score** S_{tot}: personalized relevance + coherence − cost/penalties.

What is a *palette vector* $\mathbf{p}$?

Definition

A **palette vector** $\mathbf{p} \in \mathbb{R}^d$ is a compact, normalized descriptor of the dominant *colors & materials* present in the scene images $\mathbf{v} \in \mathcal{V}^K$:

$$\mathbf{p} = \big[p_1, \ldots, p_d\big], \quad p_i \geq 0, \quad \sum_{i=1}^{d} p_i = 1.$$

Instantiations (common in practice):

- **Hybrid** (color + material): concatenate color bins with material tags (e.g., rattan, linen, leather) and train a classifier.

Palette Vector: Role in the Pipeline

Usage: $\mathbf{p}$ conditions retrieval/ranking and the *compatibility function* $\kappa(\cdot, \cdot; \mathbf{p})$ so that selected items harmonize with the scene palette.

How to train: Maybe on co-purchase data! Or distill a large VLM to classifier.

Where it enters:

- *Category targeting*: prefer categories whose canonical palettes align with $\mathbf{p}$.
- *Per-item filtering*: discard items with palettes far from $\mathbf{p}$.
- *Bundle scoring*: $\kappa(s^i, s^j; \mathbf{p})$ rewards pairwise harmony under the observed scene palette.

What is a layout affordance ℓ?

Definition

Layout affordance ℓ encodes *spatial constraints* derived from the image including

- geometry: extract the floor plan
- free space dimensions,
- anchor elements in image like window, door, etc

to determine the test feasibility of placing a set $\widehat{\mathbf{y}}$ of items.

Feasibility predicate:

$\mathrm{Layout}(\widehat{\mathbf{y}}; \ell) = 1 \iff \exists$ non-overlapping placement satisfying extracted geometry, free space di

CAL-RAG

an example of optimal layout generation

CAL-RAG: Retrieval-Augmented Multi-Agent Generation for Content-Aware Layout Design

Najmeh Forouzandehmehr, Reza Yousefi Maragheh, Sriram Kollipara, Kai Zhao, Topojoy Biswas, Evren Korpeoglu, Kannan Achan

Walmart Global Tech, Sunnyvale, California, USA

{najmeh.forouzandehmehr, reza.yousefimaragheh, sriram.kollipara, kai.zhao, topojoy.biswas, evren.korpeoglu, kannan.achan}@walmart.com

ABSTRACT

Automated content-aware layout generation—the task of arranging visual elements such as text, logos, and underlays on a background canvas—remains a fundamental yet underexplored problem in intelligent design systems. While recent advances in deep generative models and large language models (LLMs) have shown promise in structured content generation, most existing approaches lack grounding in contextual design exemplars and fall short in handling semantic alignment and visual coherence. In this work, we introduce CAL-RAG, a Retrieval-Augmented, Agentic framework for content-aware layout generation that integrates multimodal retrieval, large language models, and collaborative agentic reasoning. Our system retrieves relevant layout examples from a structured knowledge base and invokes an LLM-based layout recommender to propose structured element placements. A vision-language grader agent evaluates the layout based on visual metrics, and a feedback agent provides targeted refinements, enabling iterative improvement. We implement our framework using LangGraph and evaluate on the PKU PosterLayout dataset, a benchmark rich in semantic and structural variability. CAL-RAG achieves state-of-the-art performance across multiple layout metrics—including underlay effectiveness, element alignment, and overlap—substantially outperforming strong baselines such as LayoutPrompter. Our results demonstrate that combining retrieval augmentation with agentic multi-step reasoning provides a scalable, interpretable, and high-fidelity solution for automated layout generation.

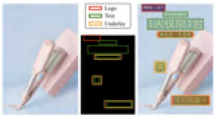


Figure 1: Example for content-aware layout generation: (a) Input canvas with background and content elements; (b) Generated layout based on visual and textual content awareness; (c) Final rendered presentation using the layout from (b).

offer limited generalization, particularly in diverse or semantically rich design contexts. The advent of deep generative models—such as GANs, VAEs, and transformers—has enabled more expressive layout synthesis by learning from annotated layout corpora [1, 2, 7, 9, 10]. However, these models are typically data-hungry, brittle to out-of-distribution inputs, and often struggle to incorporate visual-semantic alignment at inference time.

More recently, Large Language Models (LLMs) and Large Vision-Language Models (LVLMs) have demonstrated remarkable zero-

CAL-RAG

check out this paper

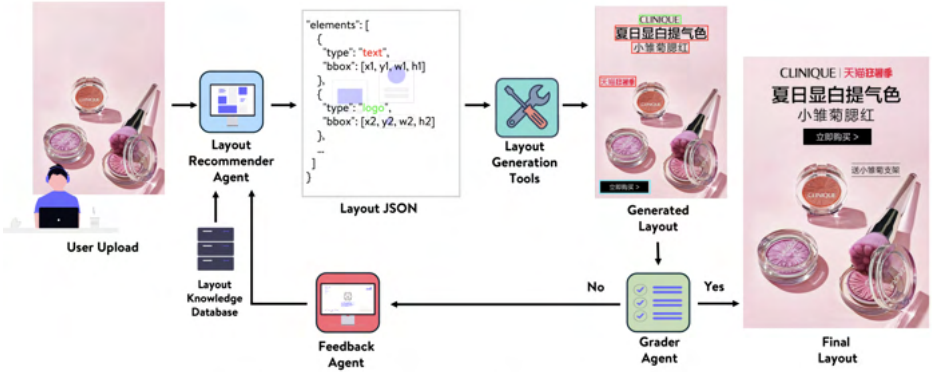


Figure 2: System Architecture Diagram of CAL-RAG.

Compatibility Kernel $\kappa(\cdot, \cdot; \mathbf{p})$ (Examples)

$$\kappa(s^i, s^j; \mathbf{p}) = \underbrace{\mathrm{sim}\big(\mathrm{pal}(s^i), \mathrm{pal}(s^j) \mid \mathbf{p}\big)}_{\text{color harmony}} + \rho \quad \underbrace{\mathrm{sim}_{\text{material}}(s^i, s^j)}_{\text{e.g., GloVe/CLIP on material tags}} \quad + \tau \quad \underbrace{\mathrm{sim}_{\text{style}}(s^i, s^j)}_{\text{style embedding cosine}} .$$

Bundle coherence:

$$S_{\text{comp}}(\widehat{\mathbf{y}}) = \frac{2}{L(L-1)} \sum_{i<j} \kappa(s^i, s^j; \mathbf{p}).$$

Notes:

- Color harmony condition by $\mathbf{p}$ to favor harmony around scene-dominant hues.
- Learned sim heads can be trained to match designer "goes-well-with" labels.

Explaining the *Total Score* S_{tot}

Decomposition

$$S_{\text{tot}}(\widehat{\mathbf{y}}) = \underbrace{\sum_{i=1}^{L} \alpha \, \mathrm{Rel}(s^i \,|\, \mathbf{x}, \mathbf{u})}_{\text{personal relevance}} + \underbrace{\beta \, S_{\text{comp}}(\widehat{\mathbf{y}})}_{\text{coherence/complementarity}} - \underbrace{\gamma \, \mathrm{Cost}(\widehat{\mathbf{y}})}_{\text{budget}} - \underbrace{\lambda \, \mathcal{P}_{\text{brand}}(\widehat{\mathbf{y}})}_{\text{style/brand penalty}} .$$

Hard vs. soft constraints:

$$\text{Hard: } \mathrm{Compat}(\widehat{\mathbf{y}}; \mathbf{p}, \mathcal{G}) = 1; \quad \text{Soft: add penalties in } S_{\text{tot}}.$$

Choosing weights: $\alpha, \beta, \gamma, \lambda$ via grid/Bayes search or *learned* from designer judgments (pairwise ranking).

Optimizing S_{tot} under Constraints

Problem:

$$\widehat{\mathbf{y}}^{\star} = \arg\max_{\widehat{\mathbf{y}}} \; S_{\text{tot}}(\widehat{\mathbf{y}}) \quad \text{s.t. Budget} \le B, \; \text{Layout}(\widehat{\mathbf{y}}; \ell) = 1.$$

Solvers (latency/quality trade-offs):

- *Greedy w/ lookahead*: fast, good for serving.
- *ILP/MIP relaxations*: exact/near-exact, best offline.

Caching: precompute per-item relevance and per-pair κ for top categories to accelerate online assembly.

Another Formal Definition

$$\mathbf{x} \in \mathcal{T}, \quad \mathbf{v} \in \mathcal{V}^K, \quad \mathbf{u} \in \mathcal{N}, \quad \widehat{\mathbf{y}} = \langle s^1, \ldots, s^L \rangle, \; s^i \in \mathcal{S}.$$

$$\text{Objective:} \quad \widehat{\mathbf{y}}^\star = \arg\max_{\widehat{\mathbf{y}}} \sum_{i=1}^{L} \mathrm{Rel}\big(s^i \mid \mathbf{x}, \mathbf{v}, \mathbf{u}\big)$$

Subject to design constraints:

$$\mathrm{Compatability}(\widehat{\mathbf{y}};\, \mathbf{p}, \mathcal{G}) = 1, \qquad \mathrm{Budget}(\widehat{\mathbf{y}}) \le B, \qquad \mathrm{Layout}(\widehat{\mathbf{y}};\, \ell) = \text{feasible},$$

where $\mathbf{p}$ is a palette vector from vision, $\mathcal{G}$ style rules, and ℓ layout affordances.

Another Formal Definition

History retrieval: $\widehat{\mathcal{C}}_{\mathsf{SEM}} = \mathcal{Q}\big(\Omega^{\mathsf{SEM}},\, \tau(\mathbf{x})\big)$

Category selection: $C = \mathrm{Cat}(\mathbf{x}, \mathbf{p}, \widehat{\mathcal{C}}_{\mathsf{SEM}})$

Per-category retrieval: $\forall c \in C,\ \mathcal{F}_{\mathsf{search}}(c, \mathbf{p}, \mathbf{u}) \to \mathcal{S}_c$

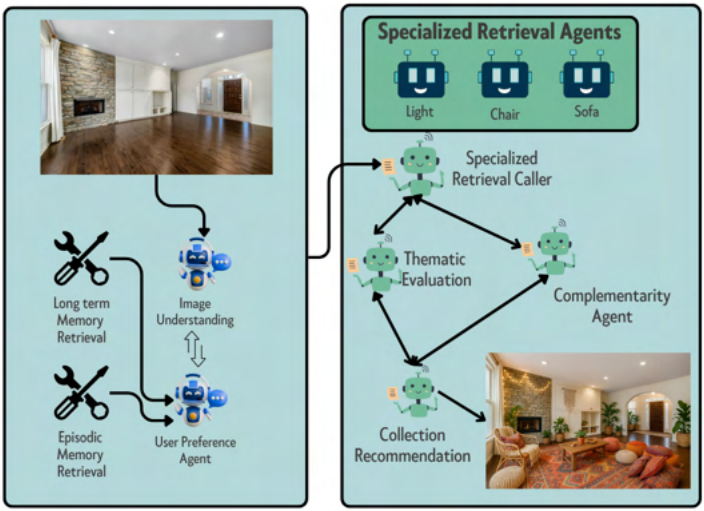
Bundle assembly: $\widehat{\mathbf{y}} = \bigcup_{c \in C} \mathrm{TopK}(\mathcal{S}_c)$

Semantic & complementarity gates: $\mathrm{SemChk}(\widehat{\mathbf{y}}) = 1,\ \mathrm{CompChk}(\widehat{\mathbf{y}}) = 1.$

End-to-End Algorithm (Pseudo-Code)

1. $(\mathbf{x}, \mathbf{v}) \leftarrow$ user input; $\widehat{\mathcal{C}}_{\text{SEM}} \leftarrow \mathcal{Q}(\Omega^{\text{SEM}}, \tau(\mathbf{x}))$.
2. $\mathbf{p} \leftarrow f_{\text{img}}(\mathbf{v})$; $\ell \leftarrow$ affordance extractor$(\mathbf{v})$.
3. $C \leftarrow \text{Cat}(\mathbf{x}, \mathbf{p}, \widehat{\mathcal{C}}_{\text{SEM}})$.
4. For $c \in C$: $\mathcal{S}_c \leftarrow \mathcal{F}_{\text{search}}(c, \mathbf{p}, \mathbf{u})$; keep TopK per category.
5. Assemble $\widehat{\mathbf{y}}$ by maximizing S_{tot} under Budget, Layout.
6. Run SemChk, CompChk; if fail, revise C/constraints and re-solve.
7. Render mock-up; return bundle and visualization to A_{chat}.

Architecture: The Visual



System Architecture: Agents

$\mathrm{MAS}_{\mathsf{mm}} = (\mathcal{A}, \mathcal{E}, \Pi)$

$$\mathcal{A} = \{A_{\mathsf{chat}}, A_{\mathsf{image}}, A_{\mathsf{history}}, A_{\mathsf{cat}}, A_{\mathsf{caller}}, A_{\mathsf{semChk}}, A_{\mathsf{compChk}}, A_{\mathsf{collect}}\}$$

- A_{image}: palette & layout from images $(f_{\mathsf{img}}, \mathcal{F}_{\mathsf{layout}})$
- A_{history}: $\mathcal{Q}$ over Ω^{SEM} for stable tastes/budget
- A_{caller}: combines signals; calls specialized micro-MAS,
- $A_{\mathsf{semChk}}/A_{\mathsf{compChk}}$: rule compliance & harmony gates
- A_{collect}: rank & render; return set to A_{chat}

Message Flow & Protocol

$$
\begin{aligned}
A_{\text{chat}} &\to A_{\text{image}}\ (\mathbf{v}) \ \ \& \ \ A_{\text{history}}\ (\mathbf{x}) \\
&\to A_{\text{caller}}\ (\mathbf{x}, \mathbf{p}, \widehat{\mathcal{C}}_{\text{SEM}}) \\
&\to \{\text{micro-MAS}_c\}_{c \in C} \to A_{\text{semChk}} \to A_{\text{compChk}} \to A_{\text{collect}} \to A_{\text{chat}}
\end{aligned}
$$

Autonomy can happen based in Agent calling to adjust the retrieval process per query.

Memory Requirements (I)

- **Short-term (STM):** live prompt, extracted palette $\mathbf{p}$, current candidate sets
- **Semantic (SEM):** durable preferences (colors, materials, budget caps), style affinities
- **Episodic (EPI):** past styling sessions (accepted/rejected bundles, returns)
- **Procedural (PROC):** routing templates, per-category query schemes, brand/style rules

Tools & Data Access (II)

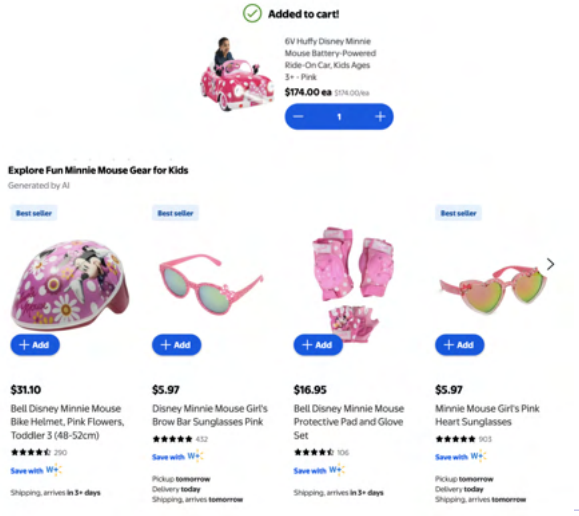
- `VisionEncoder` f_{img}, `LayoutRenderer` $\mathcal{F}_{\text{render}}$
- `VectorSearch` over `ProductDB` for semantic retrieval
- `UniversalSearch`/`PriceAPI` for stock/price validation
- `CompatEngine` for pairwise/multi-item harmony checks

Benefits

- **Designer-quality curation:** coherent, theme-consistent sets in one step
  - Now, near designer quality!
- **Lower cognitive load:** fewer iterations vs. item-by-item browsing
  - Easy user experience
- **Personal, grounded:** vision-aware + history-aware = higher acceptance
  - This was not possible before!
  - **Error containment:** semantic/complementarity gates reduce mismatch & drift
- **Modular extensibility:** add/replace micro-MAS for new styles or categories
  - Easier implementation.

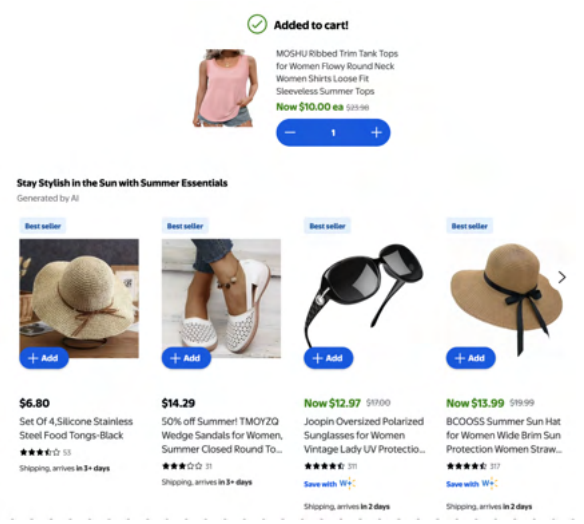
Example

Minnie Mouse



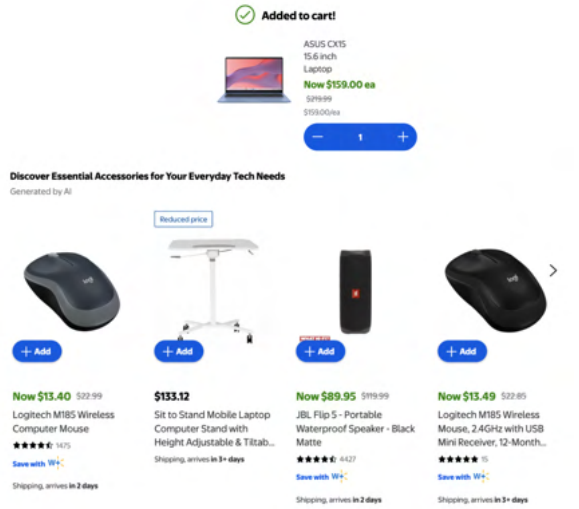
Example

Summer Dress



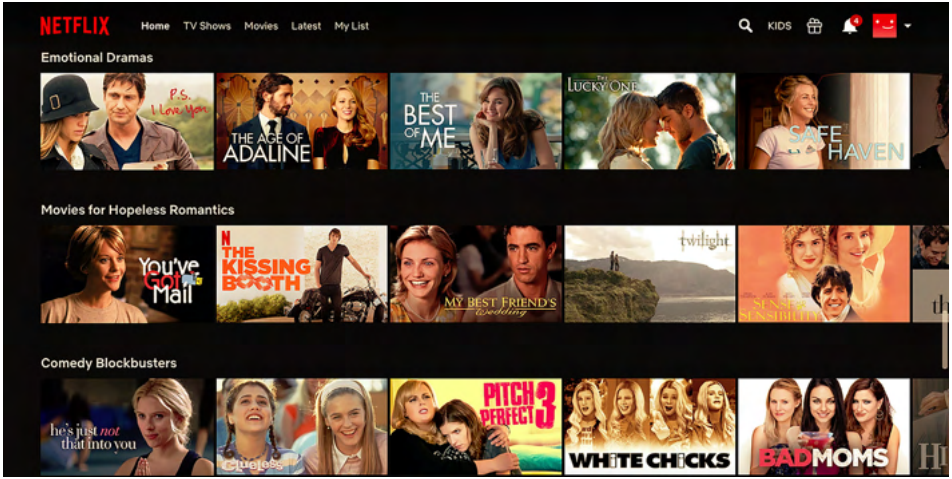
Example

Laptop for Everyday Use



Recommendation Explanation

Why are you recommending this?



Task Definition

Informal Definition

Recommendation Explanation is the process of generating intelligible and contextually relevant justifications for why particular items are recommended to a user.

Recommendation-Explanation maps items and user context to text:

$$\mathbf{r} \in \mathcal{S}^L, \quad \mathbf{u}_t = \left(\widehat{\mathcal{C}}_t^{\mathsf{SEM}}, \widehat{\mathcal{C}}_t^{\mathsf{EPI}}\right), \quad e_t = \Phi_\psi(\mathbf{r}, \mathbf{u}_t) \in \mathcal{E}_{\mathsf{text}}.$$

Constraints (consistency predicate)

$$\mathrm{Consistency}(e_t) = \underbrace{\mathrm{Factual}(e_t : \mathbf{r}, \mathbf{u}_t)}_{\text{no hallucinations}} \wedge \underbrace{\mathcal{C}_{\mathcal{P}}(e_t) = 1}_{\text{brand/style compliance}}$$

Notes. $\widehat{\mathcal{C}}_t^{\mathsf{SEM}}, \widehat{\mathcal{C}}_t^{\mathsf{EPI}}$ obtained via memory retrieval $\mathcal{Q}$; $\mathcal{C}_{\mathcal{P}}$ is a policy checker (tone, vocabulary, legal).

High-Level Goal

- **Transparency:** expose the rationale behind recommended items in one concise narrative.
  - Explaining can help bridging the gap with what user wants and the recommendation,
  - makes the experience easier.
- **Trust & engagement:** increase user confidence and CTR with factual, user-aware explanations.
  - If the explanasion explains what user is looking for –¿ then the decision process is faster and conversion happens with more probability.
  - we have tested this and have evidance for it.
- **Brand consistency:** ensure on-tone language and regulatory alignment across languages & markets.
  - This is one of the most important issues, in large launches.
- **Low latency:** integrate into serving path with bounded overhead and revise-on-fail loop.
  - Cost benefit analysis, if doing so has a positive net impact!

System Architecture (Agents)

$$\mathrm{MAS}_{\mathsf{expl}} = (\mathcal{A}, \mathcal{E}, \Pi), \quad \mathcal{A} = \{A_{\mathsf{journey}}, A_{\mathsf{session}}, A_{\mathsf{rec}}, A_{\mathsf{expl}}, A_{\mathsf{eval}}\}.$$

- $A_{\textbf{session}}$ (User Session Summary): $\mathcal{Q}$ over $\Omega^{\mathsf{SEM}} \cup \Omega^{\mathsf{EPI}}$ to form $\mathbf{u}_t$.
- $A_{\textbf{journey}}$ (Journey Detector): tags live intent from clickstream (e.g., "holiday décor upgrade").
- $A_{\textbf{rec}}$ (Recommender): returns $\mathbf{r} \in \mathcal{S}^L$ given $(\mathbf{u}_t, \text{intent})$.
- $A_{\textbf{expl}}$ (Explanation Writer): $e_t = \Phi_\psi(\mathbf{r}, \mathbf{u}_t)$.
- $A_{\textbf{eval}}$ (Explanation Evaluator): verifies factuality & brand; issues `revise` if needed.

Environment $\mathcal{E}$: ProductDB, UserProfileDB, BrandPolicy, LogStore.

Communication & Flow

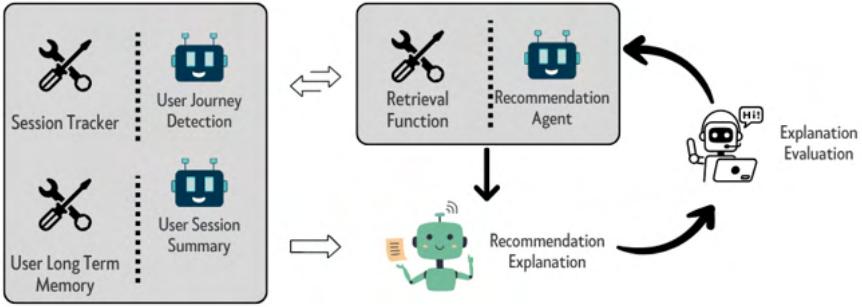
Allowed edges $(\mathbf{C}_{ij} = 1)$:

$$A_{\text{session}} \to A_{\text{rec}}, \quad A_{\text{journey}} \to A_{\text{rec}}, \quad A_{\text{rec}} \to A_{\text{expl}} \to A_{\text{eval}} \to A_{\text{rec}}.$$

Interaction loop:

1. A_{session}: $\mathbf{u}_t \leftarrow \mathcal{Q}(\Omega^{\text{SEM}} \cup \Omega^{\text{EPI}}, \tau)$
2. A_{journey}: infer intent from clicks.
3. A_{rec}: produce $\mathbf{r}$ conditioned on $(\mathbf{u}_t, \text{intent})$.
4. A_{expl}: generate e_t.
5. A_{eval}: check Factual & $\mathcal{C}_{\mathcal{P}}$; if fail $\Rightarrow$ `revise` $\rightarrow$ (3–4).

Architecture



Formal Consistency Checks

Factuality (NLI/grounding):

$$\mathrm{Factual}(e_t : \mathbf{r}, \mathbf{u}_t) = \mathbf{1}\big[\forall\, \texttt{fact} \in e_t :\ \texttt{fact} \in \mathrm{Facts}(\mathbf{r}) \cup \mathrm{Facts}(\mathbf{u}_t)\big]$$

Brand/style compliance:

$$\mathcal{C}_{\mathcal{P}}(e_t) = \begin{cases} 1, & \text{all rules in policy } \mathcal{P} \text{ satisfied} \\ 0, & \text{otherwise} \end{cases}$$

Joint predicate: $\mathrm{Consistency}(e_t) = \mathrm{Factual}(e_t) \wedge \mathcal{C}_{\mathcal{P}}(e_t)$.

Optional score:

$$J(e_t) = \lambda_1 \,\mathrm{Readability}(e_t) + \lambda_2 \,\mathrm{Specificity}(e_t) - \lambda_3 \,\mathrm{Redundancy}(e_t),$$

maximized subject to $\mathrm{Consistency}(e_t) = 1$.

Memory & Tools

Memory (by agent):

- A_{session}: Ω^{SEM} (stable prefs), Ω^{EPI} (recent sessions), Ω^{STM} (current turns).
- A_{expl}: Ω^{PROC} (brand tone exemplars, banned phrases, templates).
- A_{eval}: cached fact table for $\mathbf{r}$; policy store $\mathcal{P}$.

Tools $\mathcal{F}$:

- `UserData.fetch`, `ProductDB.lookup` (explanation and item grounding).
- `NLI.verify` (trace facts).
- `PolicyCheck` (brand/legal), `StyleTransfer` (tone repair).

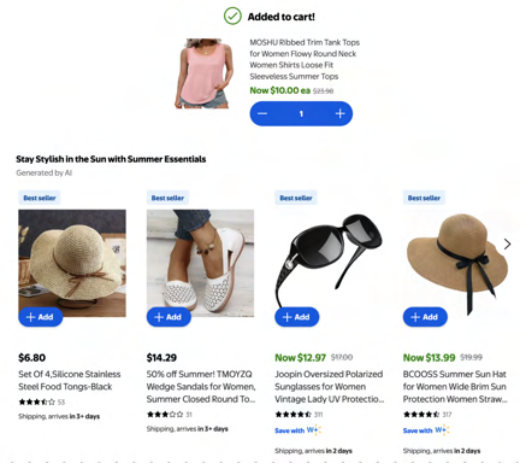
Benefits & Discussion

- **Trust & CTR uplift:** clear, user-aware rationale improves acceptance.
- **Brand safety at scale:** evaluator veto enforces $\mathcal{P}$ with low integration cost.
- **Personal continuity:** explanations reuse the same $\Omega^{\text{SEM}}/\Omega^{\text{EPI}}$ driving ranking.
- **Modularity:** new style guides or fairness rules added via Ω^{PROC} and A_{eval} prompts.

Implementations

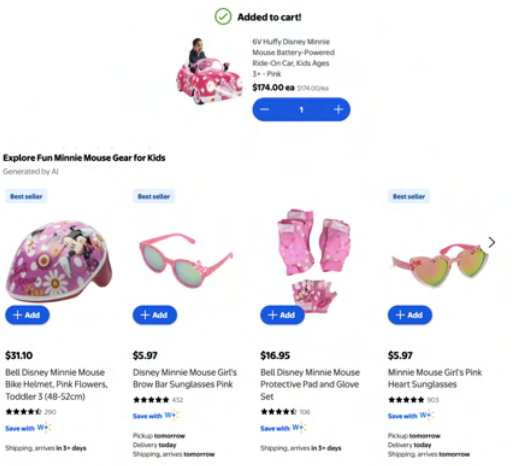
Intent Base Recommendation Model-post add to cart explore page

Image Credit: **Walmart.com**



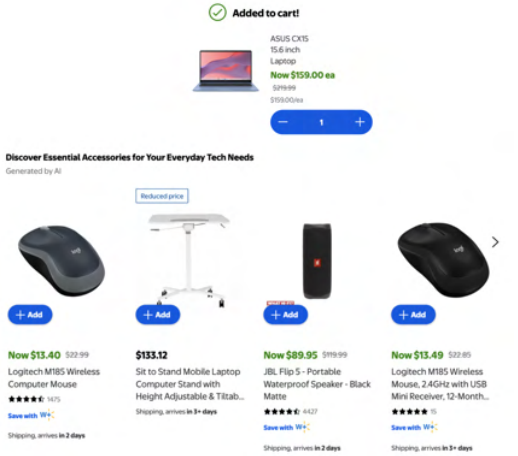
Intent Base Recommendation Model-post add to cart explore page

Image Credit: **Walmart.com**

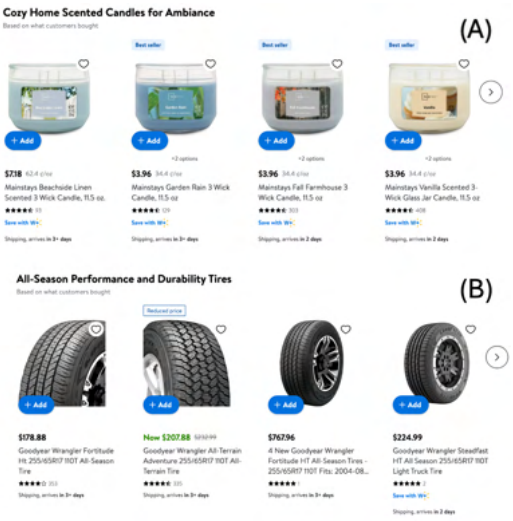


Intent Base Recommendation Model-post add to cart explore page

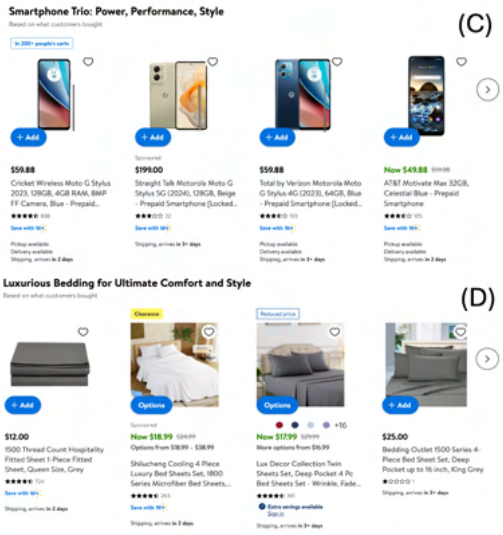
Image Credit: **Walmart.com**



Explain It to Me-item page



Explain It to Me-item page

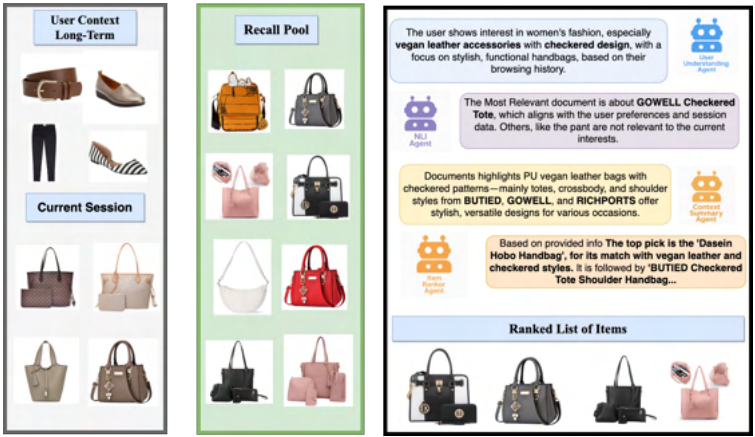


Explain It to Me-item page

Table: A/B testing evaluation results. The lift results are stat sig.

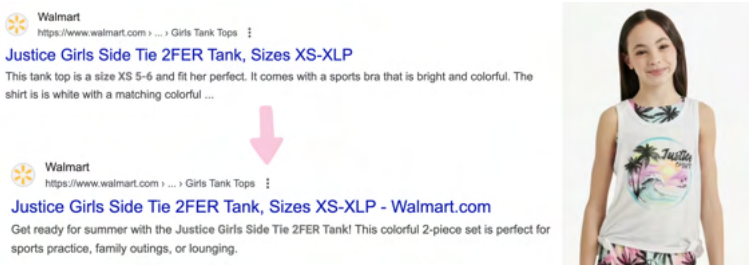
	CTR	GMV
Lift	>50pbs	>100pbs

Agentic RAG for Personalized Recommendation



MetaSynth for SEO

Credit: **Google.com**



MetaSynth for SEO

Table: A/B testing evaluation results. The lift results are stat sig.

	CTR	Traffic
Lift	+1026 bps	+7510 pbs

Search Compare Eval-MAREval

Credit: **Walmart.com**

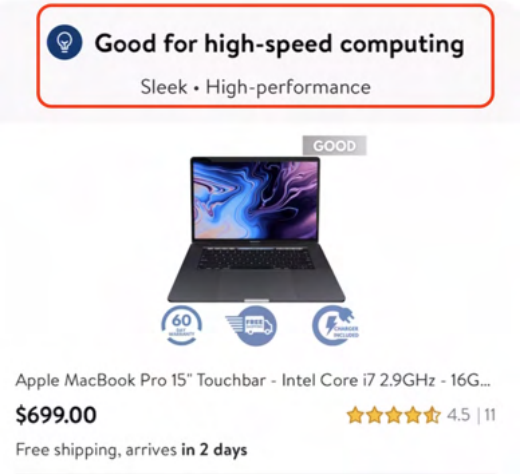
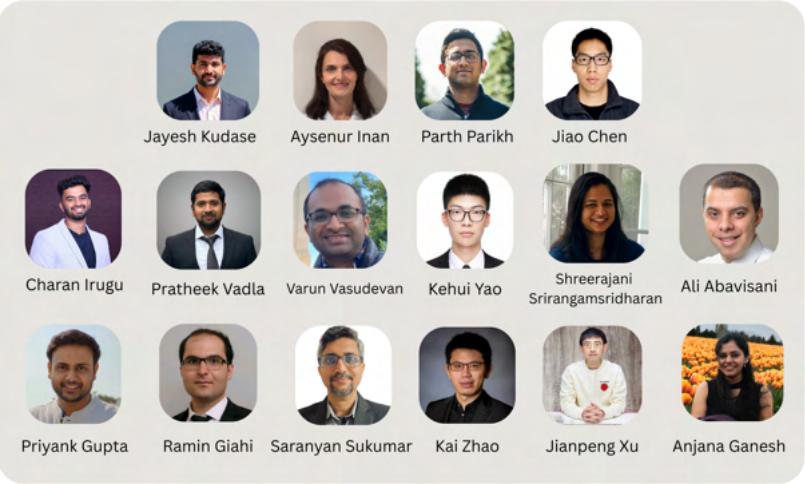


Table: online A/B test results

Metrics	Improvement	P-Value
ATC	118 bps	0.05
GMV	136 bps	< 0.01

Personalization Team

Acknowledgment



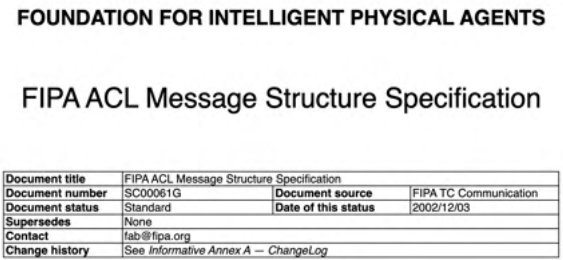
MCP in Multi-Agent RecSys: Why It Matters

- Agents must exchange **user context**, **item signals**, and **intermediate results** reliably.
- A robust **Model Context Protocol (MCP)** ensures message clarity, task negotiation, and state sync.
- Weak/adhoc protocols $\Rightarrow$ *bottlenecks*, *misinterpretations*, and *integration debt*.
- RecSys adds pressure:
  - **low latency**,
  - **high update rate**,
  - **clear semantics-in explanasion usecases**.

Protocol Standardization Problem & Goals

Snapshot: FIPA.org

- Goal: **plug-and-play** agents across teams/vendors via a shared syntax & semantics.A standard communication protocol.
- Legacy lesson (e.g., FIPA ACL): semantics help, but **complexity can hinder adoption**.



FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS

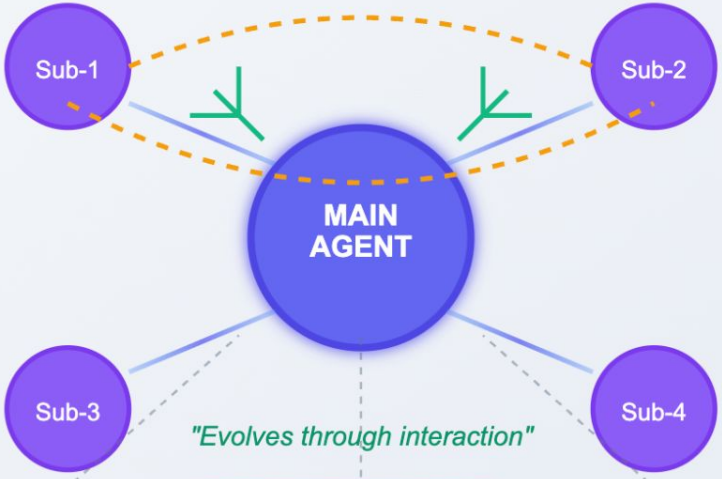
FIPA ACL Message Structure Specification

Document title	FIPA ACL Message Structure Specification		
Document number	SC00061G	**Document source**	FIPA TC Communication
Document status	Standard	**Date of this status**	2002/12/03
Supersedes	None		
Contact	fab@fipa.org		
Change history	See *Informative Annex A — ChangeLog*		

Frontiers of Agentic AI

RecSys'25 Tutorial
Agentic Recommendation Systems

𝕏 @Chi_Wang_



"Evolves through interaction"

2023: AutoGen (AG2)

Foundation laid

2025: Industry Progress

What remains?

North Star: AI Agent That Grows With You

NATURAL INTERFACE

Sight • Sound • Context

STRONG CAPABILITY

Code • Action • Learning

SCALABLE ARCHITECTURE

Multi-Agent • Parallel

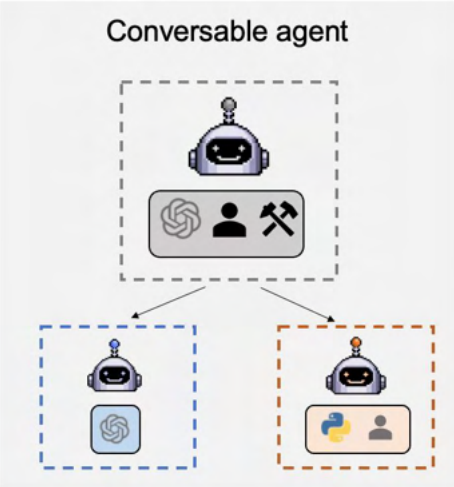
AG2: Open-Source AgentOS

- Simplicity: Simplify developers' thought process
  - **Intuitive unified agentic abstraction**
- Capability: Enable advanced exploration & diverse needs
  - **Flexible multi-agent orchestration**
- Reusability: Minimize unnecessary complexity
  - **Composable agentic design patterns**

Agentic Abstraction

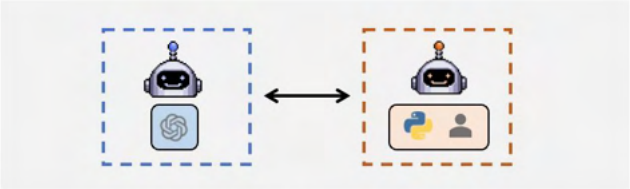
Multi-Agent Orchestration

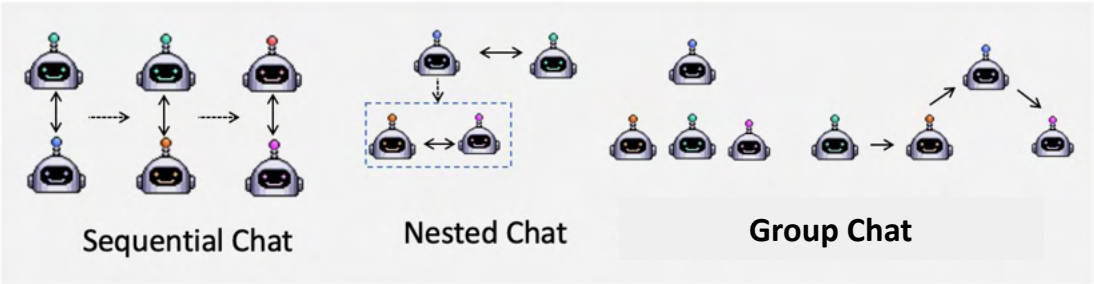
Step 1. Define AG2 agents:
Conversable & Customizable



Agent Customization

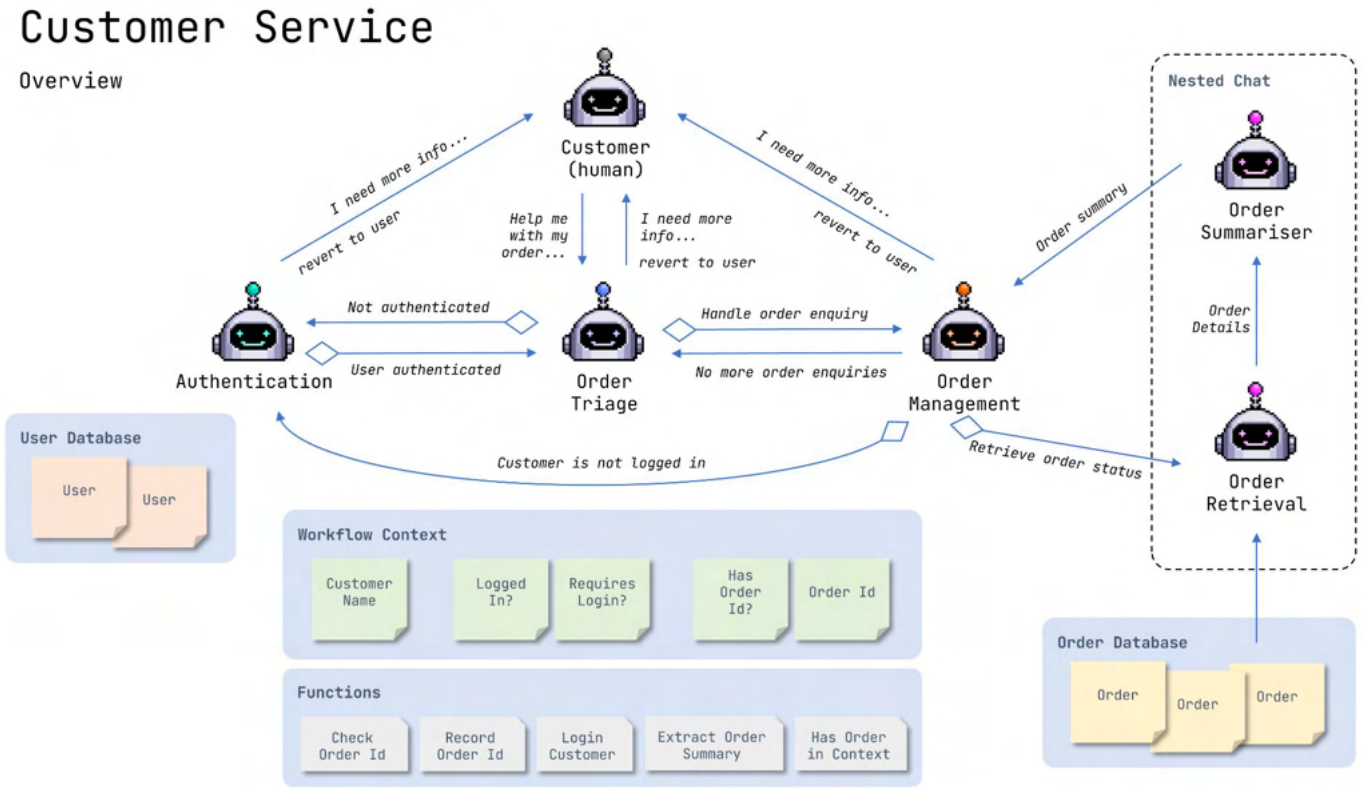
Step 2. Get them to talk:
Conversation Programming





Flexible Conversation Patterns

Multi-Agent Orchestration Example



FLAML → AG → AG2

Top 100
Open source achiever

5K
Forks

Forbes, The Economist, WIRED...

40K
Stars

20K
@Discord

1M
Downloads /month

Top 1
OSWorld
SWE-Bench Lite
WebVoyager
GAIA...


AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation Framework

Qingyun Wu[1], Gagan Bansal[2], Jieyu Zhang[3], Yiran Wu[1], Shaokun Zhang[1], Erkang Zhu[2], Beibin Li[2], Li Jiang[2], Xiaoyun Zhang[2], and Chi Wang[2]

[1]Pennsylvania State University
[2]Microsoft
[3]University of Washington

Abstract

1 Introduction

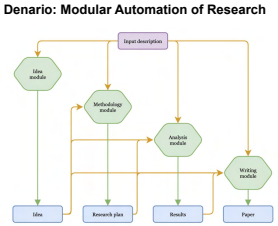
arXiv:2308.08155v1 [cs.AI] 16 Aug 2023

08.2023: Research paper

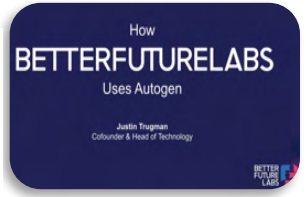
03.2023: Initial Prototype
- Flexible multi-agent conversation framework
- Code/function execution

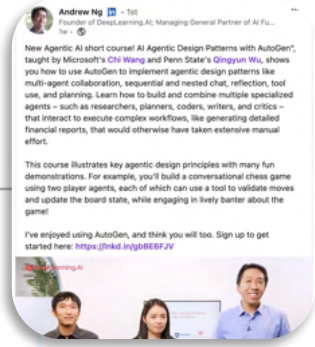
10.2023: Top trending on GitHul

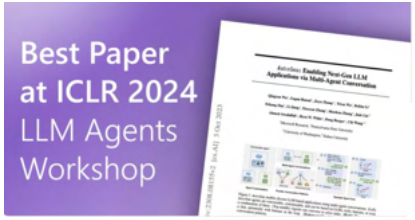
12.2023:
5 favorite AI papers by The Sequence



https://github.com/AstroPilot-AI/Denario







COLM

Example Production Use Cases

Agent Platform (Google)

Autonomous Trading

Business Automation (Cegid)

Chip Design (Nvidia)

Customer Support (Parker Hannifin)

Cyber Security

Data Engineering (Nexla)

Farming

Investment Research (BFL)

Marketing (Walmart)

Patient Support

Recommendation (Walmart)

Software Engineering

Software Testing

Task Management

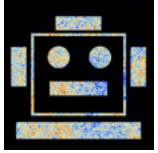
Web Automation (Emergence)

Automation of Research Workflows with AI Agents

Boris Bolliet
University of Cambridge

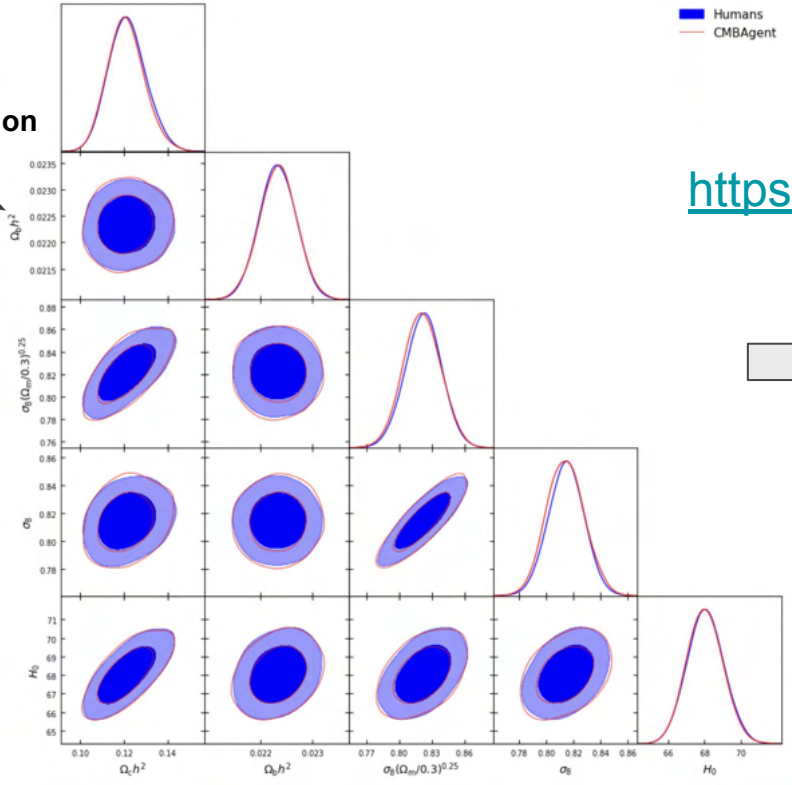
Cmbagent

Self-Driving Cosmological Laboratories
Boris Bolliet (University of Cambridge)



Agents reproduces the lensing analysis, with minimal human input

Matter fraction constraint



https://github.com/CMBAgents/cmbagent

Analysis fully reproduced in ~10'
Pipeline entirely written by **cmbagent**

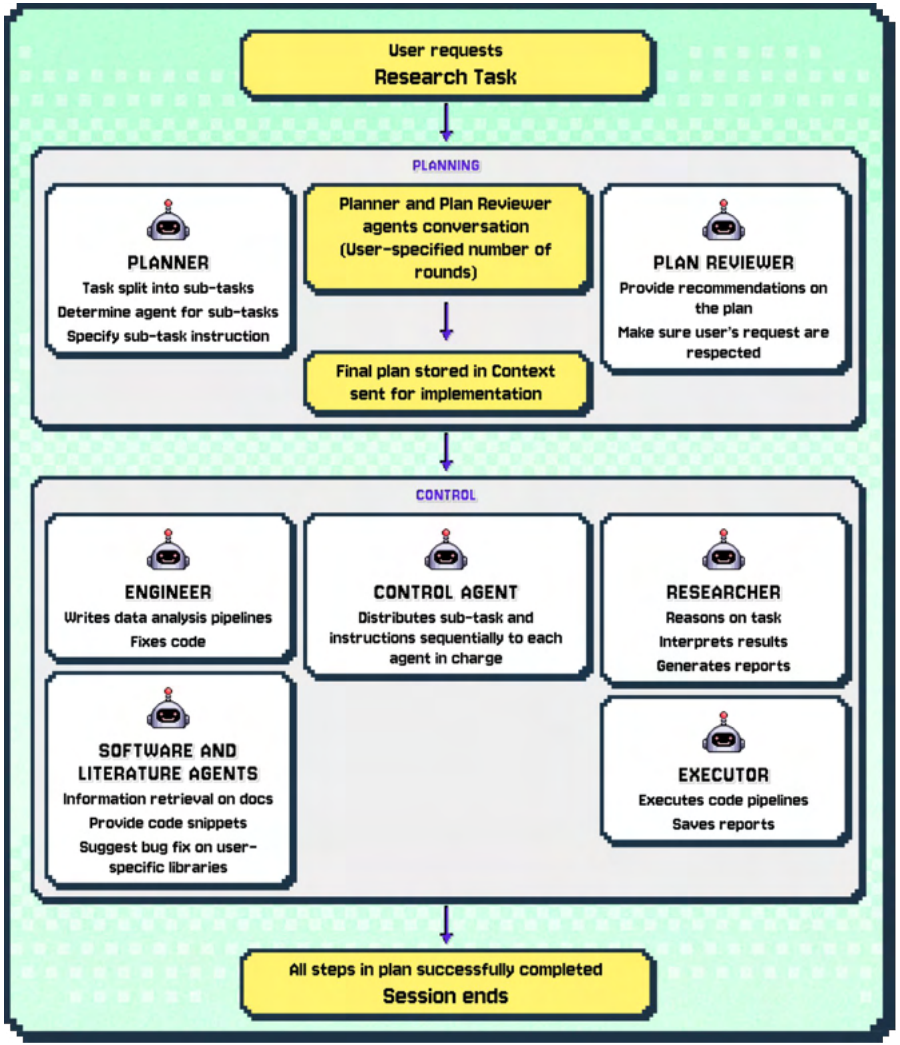
Would have taken ~ a full day of work to an experienced cosmologist.

Hubble parameter constraint

AG2 × UNIVERSITY OF CAMBRIDGE × CMBAGENT

Features That Helped:

- 🤖 Tool Calls
- 🤖 Multi-LLM Calls
- 🤖 Group Chat
- 🤖 Structured Output
- 🤖 Agentic RAG

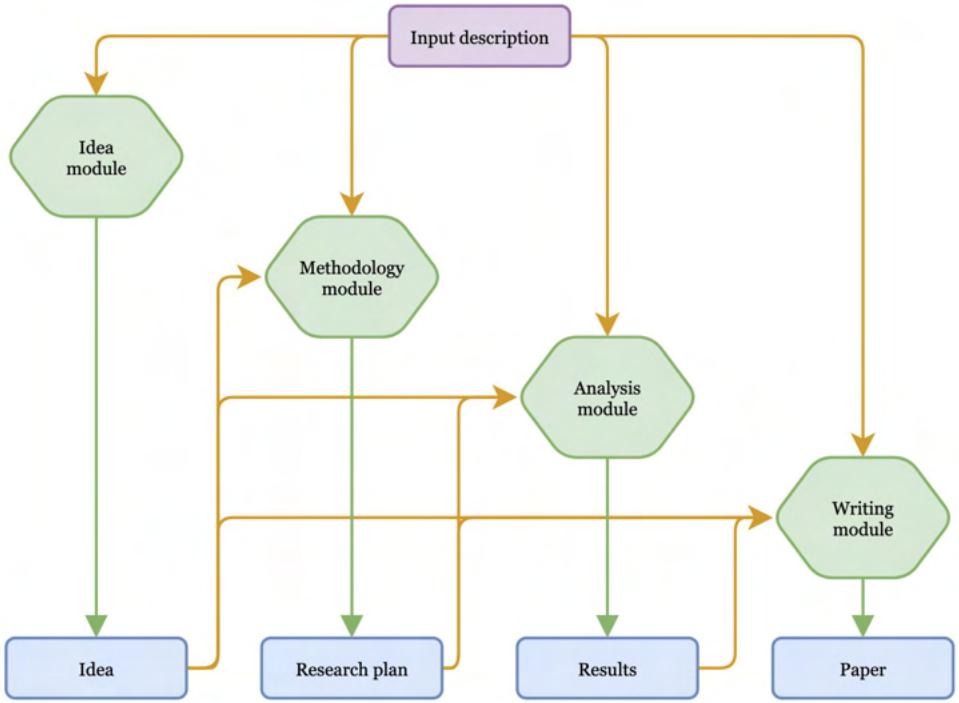



Start
CONTROL

The `Denario` Project: Modular Automation of Scientific Research with Multi-Agent Systems

Francisco Villaescusa-Navarro[1,2,*], Boris Bolliet[3,4,*], Pablo Villanueva-Domingo[5,*], Adrian E. Bayer[1,2], Aidan Acquah[6], Chetana Amancharla[7], Almog Barzilay-Siegal[8], Pablo Bermejo[9,10,11], Camille Bilodeau[12], Pablo Cárdenas Ramírez[13,14,15], Miles Cranmer[16], Urbano L. França[17,18], ChangHoon Hahn[19,20], Yan-Fei Jiang[1], Raul Jimenez[21,22], Jun-Young Lee[1], Antonio Lerario[23], Osman Mamun[13], Thomas Meier[24], Anupam A. Ojha[25,26], Pavlos Protopapas[27], Shimanto Roy[12], Pedro Tarancón-Álvarez[21,28], Ujjwal Tiwari[7], Matteo Viel[23,29,30,31,32], Digvijay Wadekar[33], Chi Wang[34], Bonny Y. Wang[35], Licong Xu[36,4], Yossi Yovel[8,37], Shuwen Yue[13], Wen-Han Zhou[38], Qiyao Zhu[25], Jiajun Zou[39], Íñigo Zubeldia[36,4]

Denario: Modular Automation of Research



https://github.com/AstroPilot-AI/Denario

Multi-agent orchestration with

AG2

https://ag2.ai/

LangGraph

Deep Research with

CMBAGENT

https://github.com/CMBAgents/cmbagent

Community Talks

Journey of taking Testzeus-Hercules to production…

Interactive Speculative Planning

Globant Code Fixer Agent: #1 on SWE-Bench Lite

Agentic Commerce: Enabling AG2 with identity, payment and monetization

Building Multi-Agent Systems for Investment Analysis

Introducing FastAgency - the fastest way to bring AutoGen workflows to production

Agent-Model Orchestration in Multi-Agent Applications

Copilot Agent Architecture Designing

Physics-Aware AI: Bridging Science Through Multi-Agent Systems

Trace-ing the Path to Self-adapting AI Agents

NOVA: Building Agentic Workflows for Structured Data Intelligence at Nexla

Investigating Group Decision-Making Mechanism in Decentralized Multi-Agent Collaboration

Multi-AI Agents for Chip Design

From Content to Conversations: Building Social Media & WhatsApp Automation with AG2

Make AI Agents Collaborate: Drag, Drop, and Orchestrate with Waldiez

Crypto Trading with AG2

Frontiers of LLM Agents: Memory, Tool Use, Multi-Modal Input, and RL with LLMs

Pedagogy Support using AG2

Exploring Pragmatic Patterns in Agentic Systems

A Multi-Agent Approach to Podcast Generation with AG2

Maris: A Security Controlled Development Paradigm for Multi-Agent Collaboration Systems

Example Projects for Learning

https://github.com/ag2ai/build-with-ag2

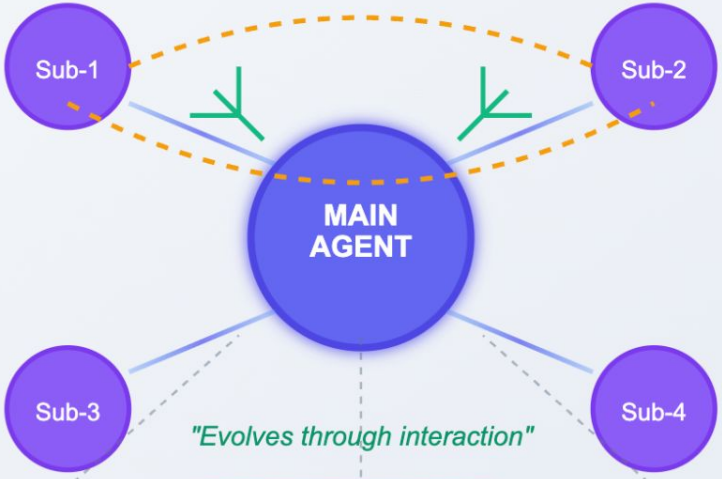
AG2

THE OPEN-SOURCE AGENTOS

GitHub ⭐

Join our growing community of over 20,000 agent builders

Discord



"Evolves through interaction"

2023: AutoGen (AG2)

Foundation laid

2025: Industry Progress

What remains?

North Star: AI Agent That Grows With You

NATURAL INTERFACE

Sight • Sound • Context

STRONG CAPABILITY

Code • Action • Learning

SCALABLE ARCHITECTURE

Multi-Agent • Parallel

Natural Interface – Computers with Senses

Today:

- Gemini Live photo/video guidance
- Screen-aware prototypes
- Real-time interactions

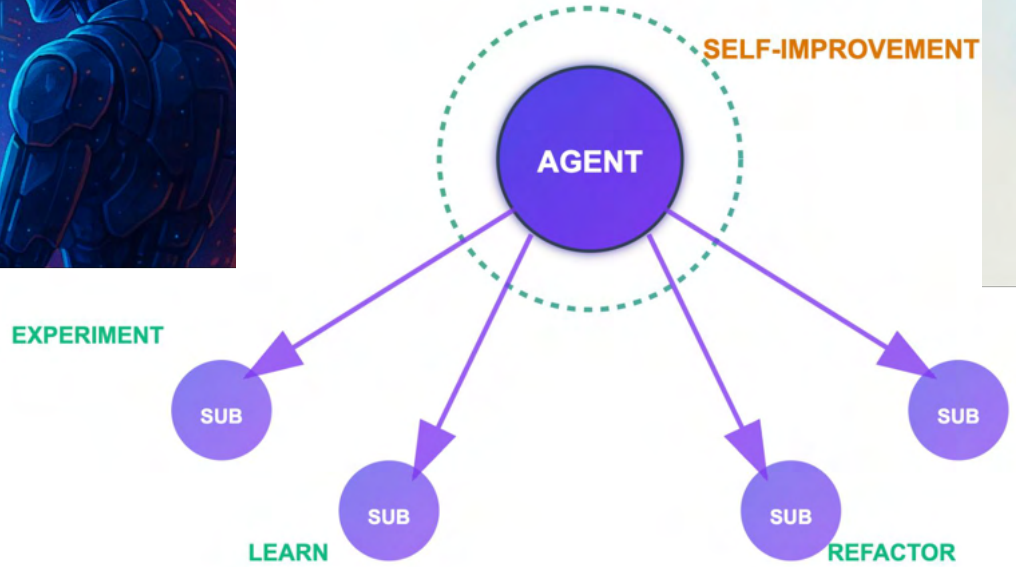
Missing: Context Flow Across Devices

Challenge: Know When to Engage vs. Step Back

Intelligent timing + Seamless context = Natural interface

CODE IS META-POWER

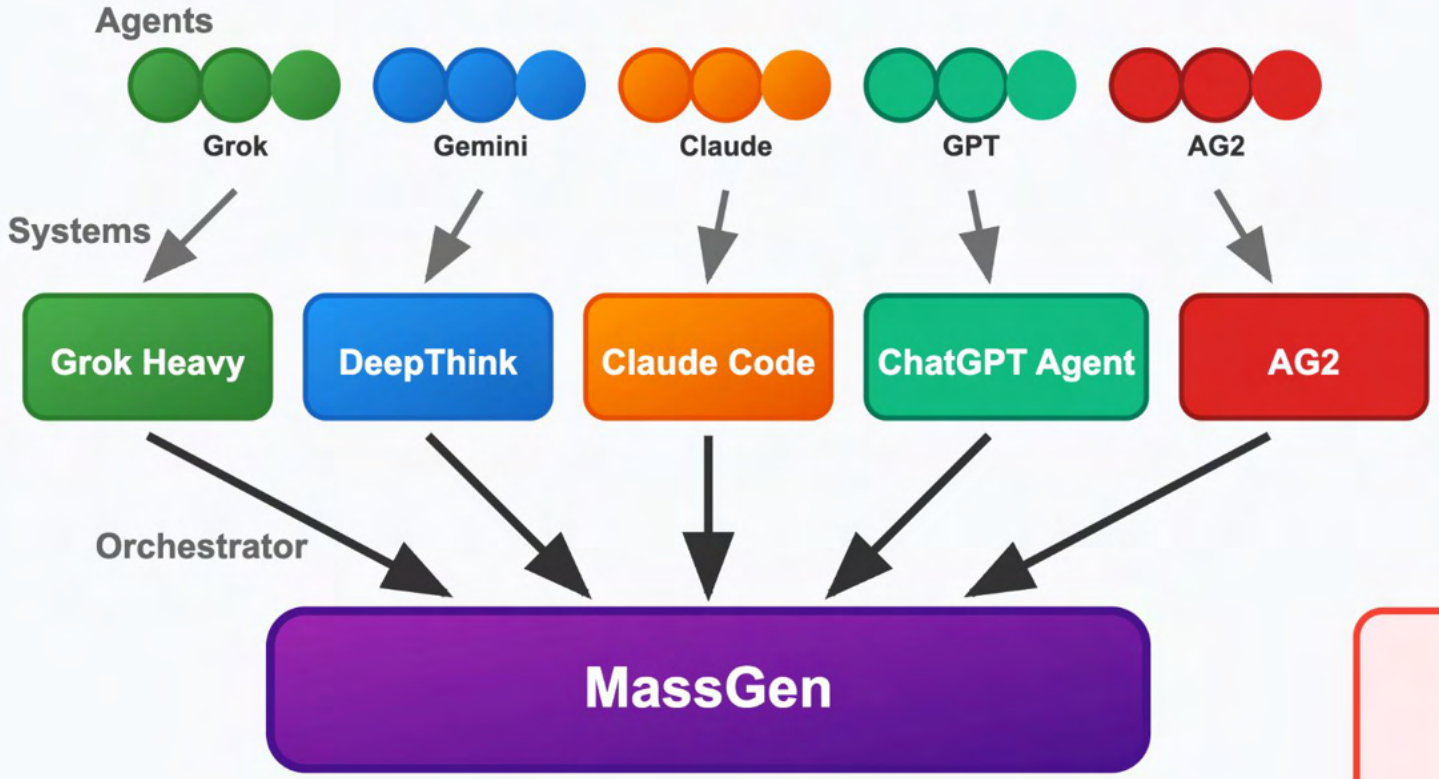
Claude Code • Gemini CLI • Codex • ChatGPT Agent

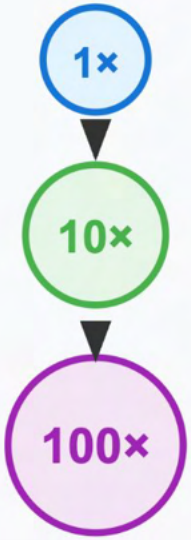


To Realize This Vision:

- Long-term Memory
- Cross-session Reflection
- Learning by Interaction

Many Minds Working as One





Challenges

Consensus

Shared Context

Interop

Building the Future Together

Natural interfaces
Turn intention into understanding

Strong capabilities
Turn understanding into action & self-improvement

Scalable architectures
Enable co-creation across many agents

AI that grows with us

@Chi_Wang_

MassGen

Scaling AI Through Multi-Agent Collaboration

🎯 **RecSys'25 Tutorial**

Agentic Recommendation Systems

📍 Prague, Czech Republic • September 26, 2025

🌐 massgen.ai | GitHub

How Do We Scale Up AI?

Traditional Scaling Laws Hit Limits

Power Crisis

4-16 GW by 2030

Enough to power entire cities

Data Depletion

Depleted by 2026-2028

Quality text data exhausted

Performance Plateau

GPT-5 delayed >1 year

Early training runs failed

⚠ **Inference-time scaling**

No universal way to leverage improvements & address limitations

The New Paradigm:

Model → Agent → **Multi-Agent Systems**

🤝

The Promise of Multi-Agent Collaboration

- → **Study Group Dynamics:** Like humans collaborating on complex problems
- → **Cross-Ecosystem Integration:** Bridge Claude, Gemini, GPT, Grok, and specialized coding agents
- → **Emergent Intelligence:** Collective problem-solving beyond individual capabilities
- → **Real-time Intelligence Sharing:** Agents learn and adapt from each other

The Promise of Collaborative Reasoning

📖 root.massgen.ai - "Myth of Reasoning"

🏗️ Built on AG2's foundational multi-agent research and community

Proven Performance Gains

Grok-4 Standard

1

Single Agent Processing

38.6%

Last Human Exam Score

$30/month

Grok-4 Heavy

A1 A2 A3

Multi-Agent Collaboration

44.4%

Last Human Exam Score

$300/month

Gemini 2.5 DeepThink

Competition Gold Medals

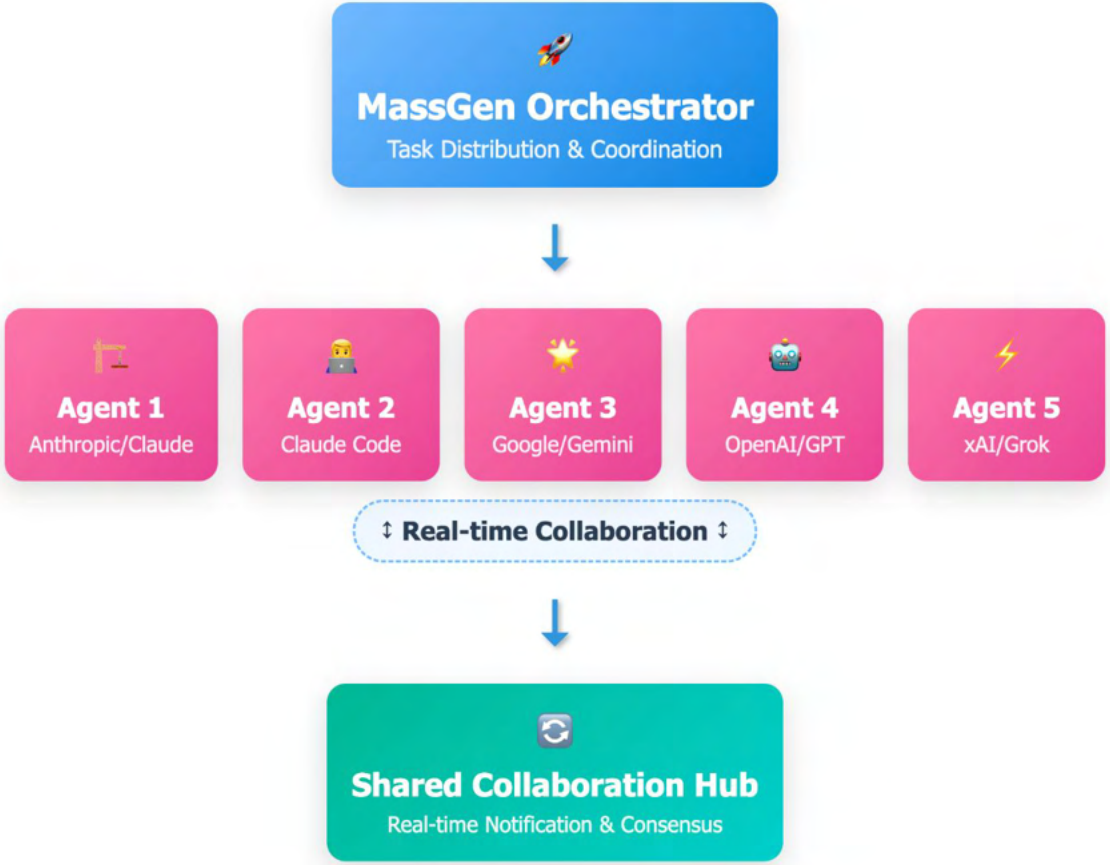
IMO + ICPC

5/6 IMO Problems (2024)
10/12 ICPC Problems (2025)

First AI Gold Medals

🚀 Multi-Agent Revolution

"Individual AI excellence + Multi-agent coordination = Next frontier of AI capabilities"


MassGen Orchestrator
Task Distribution & Coordination
Agent 1
Anthropic/Claude
Agent 2
Claude Code
Agent 3
Google/Gemini
Agent 4
OpenAI/GPT
Agent 5
xAI/Grok
↕ Real-time Collaboration ↕
Shared Collaboration Hub
Real-time Notification & Consensus

Key Features & Capabilities

Multi-Backend Support

Parallel Processing

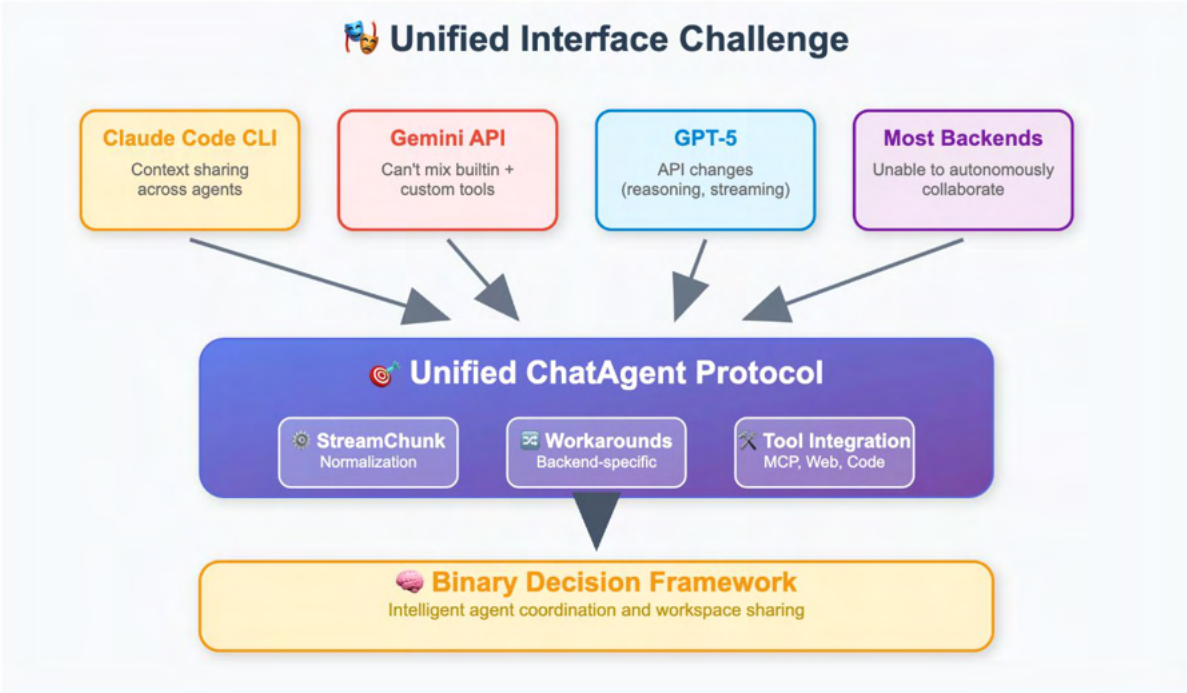
Iterative Refinement

The Reality of Reasoning

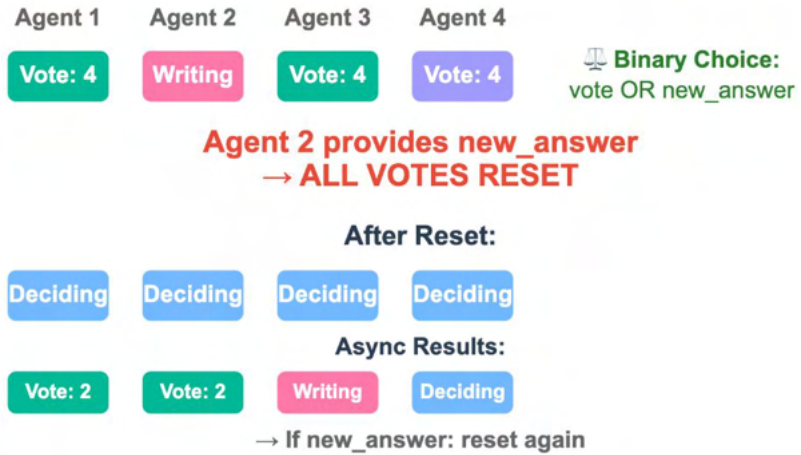
Intelligence Sharing

Consensus Building

Tech Deep Dive: Backend Abstraction Challenges



Tech Deep Dive: Binary Decision Framework Solution



🎯 **Key Innovation: Vote Invalidation Creates Dynamic Consensus**

Case Study: Success Through Peer Correction

Graduate-level physics question from GPQA-Diamond benchmark

🌌 The Problem

A quasar shows a peak at 790 nm wavelength. Given Lambda-CDM cosmological parameters (H_0 = 70 km/s/Mpc, Ω_m = 0.3, $\Omega\Lambda$ = 0.7), what is the comoving distance?

Options: A) 8 Gpc B) 7 Gpc C) 6 Gpc D) 9 Gpc

🎯 Final Result

✅

Correct Answer: A (8 Gpc)

Orchestration succeeded where individual agents initially failed

🤖 Round 1: Initial Answers

Claude: "I calculate ~6 Gpc → Answer C"
GPT-5: "I get ~8.95 Gpc → Answer D"
Gemini: "~6.1 Gpc → Answer C"

🔄 Self-Correction Process

Claude observes: *"There is significant discrepancy in calculations: Agent1 gets ~6.1 Gpc, Agent2 gets ~8.95 Gpc. Let me re-examine..."*

✨ Breakthrough Moment

Claude revises: *"Standard cosmological calculators yield 8000-8500 Mpc for z=5.5. This equals 8.0-8.5 Gpc, closest to option A."*

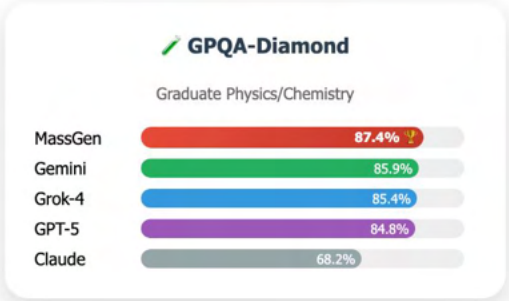
Result: *3/4 agents converge on correct answer*

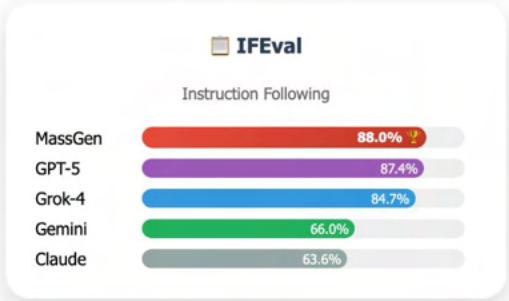
💡 Success Mechanism:

Peer observation → Discrepancy detection → Self-correction → Consensus

Benchmarking: Preliminary Results

Scientific evaluation across graduate-level reasoning, instruction-following, and narrative tasks





Overall Champion

MassGen: 81.2%

Wins 2/3 benchmarks • Statistically significant

Key Results:

- **Highest on 2/3** benchmarks
- **Best overall** average
- **Consistent** performance

Statistical:

- vs Claude: **p = 1.4e-07**
- vs Gemini: **p = 1.1e-28**
- Not due to chance

Research Gap:

- **Oracle:** 95.5% (GPQA)
- **Actual:** 87.4%
- **Potential:** 8.1 points

Agentic Recommendation Applications

Multi-Agent Personalization Pipeline

Content Analysis

Multiple agents analyze user behavior, content features, and contextual signals simultaneously

Preference Fusion

Collaborative filtering, content-based, and deep learning agents debate optimal recommendations

Dynamic Ranking

Real-time consensus building for personalized item ranking and diversity optimization

Cross-domain expertise, multi-objective optimization, explainable recommendations

Live Demonstrations

🌐 **LLM Fun Facts Website (v0.0.14):** Claude Code agents create interactive websites with enhanced logging and workspace isolation

Result: Conflict-free parallel development with comprehensive versioning

🗂️ **Unified Filesystem (v0.0.16):** Cross-backend collaboration between Gemini and Claude Code agents creating educational content with shared workspace management

Result: First-time cross-backend coordination producing comprehensive 25-slide presentations

🏆 **IMO 2025 Winner Research:** Multi-agent fact-checking → unanimous consensus on Google DeepMind victory

Result: Accurate identification despite conflicting information

💰 **Technical Analysis:** Complex Grok-4 HLE pricing calculation through iterative refinement

Result: Accurate cost estimates through collaborative validation

📚 **case.massgen.ai - Complete Case Studies**

Get Started in 60 Seconds

```
# 1. Clone and setup
git clone https://github.com/Leezekun/MassGen
cd MassGen && pip install uv && uv venv

# 2. Configure API keys
cp .env.example .env # Add your API keys

# 3. Run single agent (quick test)
uv run python -m massgen.cli --model gemini-2.5-flash "When is your knowledge up to"

# 4. Run multi-agent collaboration
uv run python -m massgen.cli --config three_agents_default.yaml "Summarize latest news of
github.com/Leezekun/MassGen"
```

✅ Supported Models & Providers

🏢 Major Providers:

Anthropic Claude & Claude Code • Google Gemini • OpenAI GPT • xAI Grok • ZAI GLM

🏠 Local & Extended:

Cerebras • Fireworks • Groq • LM Studio • OpenRouter • Together...

🛠️ Advanced Tools

Web Search • Code Execution • MCP Tools • File Operations • Browser Automation • Advanced Permissions

Considerations in Building Practical Agentic AI Recommender Systems

September 2025

Considerations in Building Successful Multi-Agentic AI Recommender System

- **Background:** Prior to the introduction of GenAI, customer facing typically had limited world knowledge outside of pretrained embeddings. This limited 1) ***dynamic content generation***, and 2) ***building an even more powerful ML-driven solutions***. GenAI provided path to solving these. However…

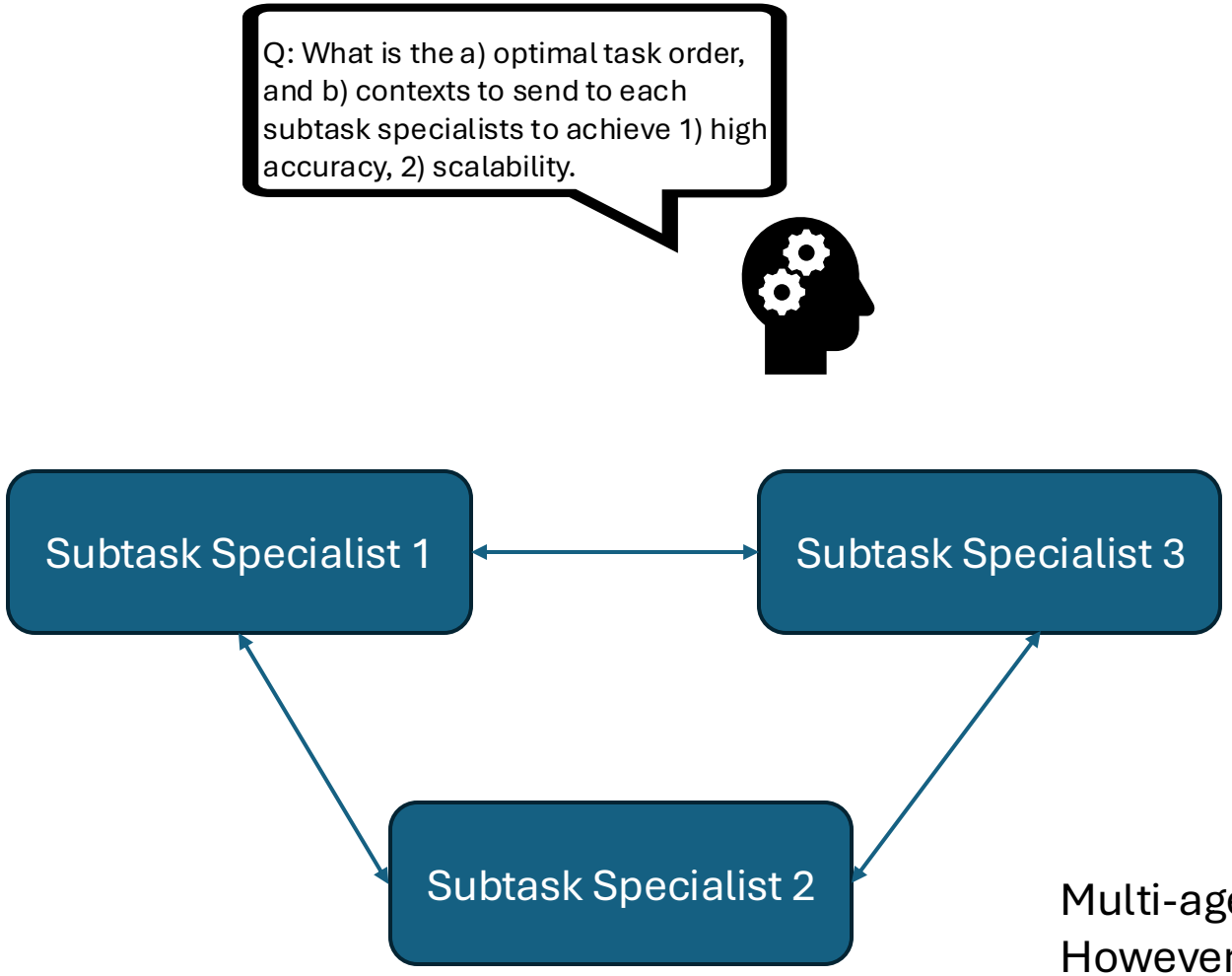
Constraints:

Product Goals	Brand Guidelines	Improve Business KPI's	Scalability	Accuracy
Are we building the product we want?	Ensure the product sounds/feels like Walmart	We also care about moving the needle	Build a product that scales to 100M+ customers	Outcome needs to be factually correct

Balancing these using Out-of-the-box LLM's were difficult.
You also can't context-window your way to your solution!

Multi-Agentic Framework: How can Subtask specialists help?

- **Problem Setting:**
  System that can a) orchestrate conflicting sub-tasks to b) achieve common task.



- **Must have**
  - Flexibility to customize (potentially growing list of) subtasks
  - Ability to converge upon satisfying subtask constraints
  - Not bound to any specific LLM's
- **Good-to-have**
  - Scalable (cheaper) solution
  - Easy to prototype new agentic architecture

Multi-agentic framework allows for subtask specialists.
However, what is the best conversation (orchestration) pattern?

Business Constraints: A multi-faceted optimization Problem!

- **Problem Setting:**
  Before embarking on the Agentic System, understand the landscape that these agents are operating on! Legal, Marketing, Creative, and Operation teams can all impose different types of constraints. Identify if these constraints are good to have vs must have.

Legal

- Are we allowed to show/claim this?
- Trademark/Copyright infringement
- Do we need a disclaimer?

Marketing

- What is the marketing goal?
- What worked before and what did not? Incorporate the findings into the generation process

Creative

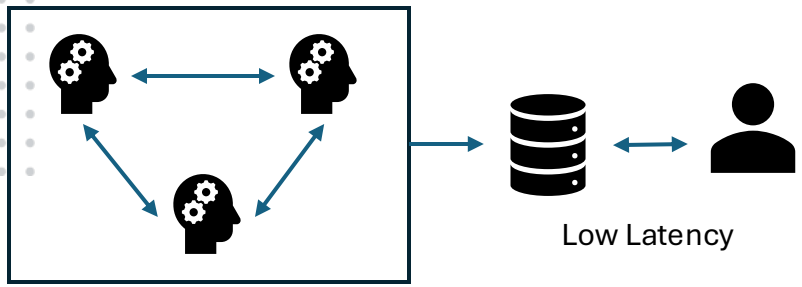
- What is our brand's voice?
  - Friendliness, down-to-earth or formal, etc's
- Ensure consistent tone and grammar

Operations

- Ease of use – how do we deploy this framework
- Any blacklisted products or messages we should not be showing?

Engineering Constraints: Scalability, Cost, and Latency

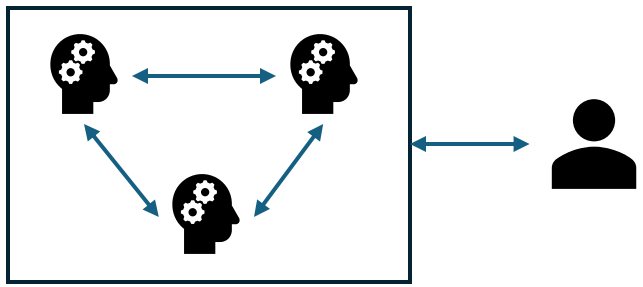
- **Problem Setting:**
  Agentic Framework takes time for real time inferencing. Identify whether the end user expects to wait couple of seconds vs they expect an immediate response.



Batched Inferencing

Existing touch points (search, recommendations, catalog, derived attributes, etc's)

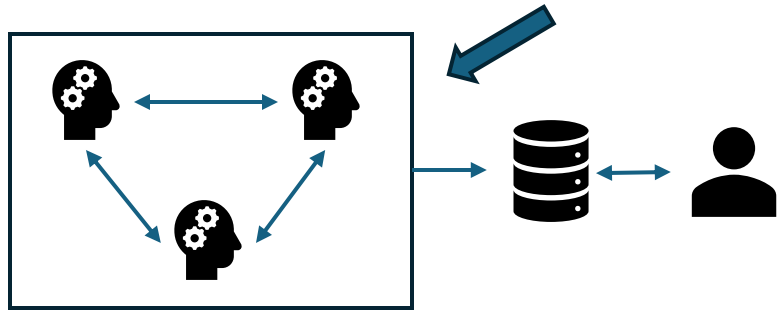
Users expect to get results immediately – so any agentic AI based inferencing should be computed offline



Real-Time Inferencing

Interactive Usecases (Chatbot, planner, research agents, etc's)

Users do not expect immediate results. That said, important to build a specialized agents that's not just a wrapper.



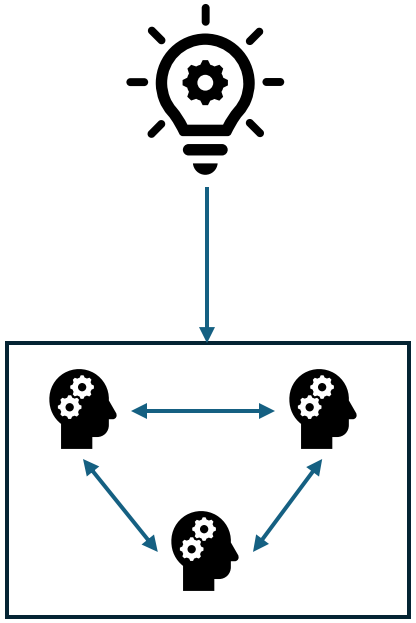
Hybrid (Near-real time)

Enhancements of existing touch points, passive in-session listener to improve

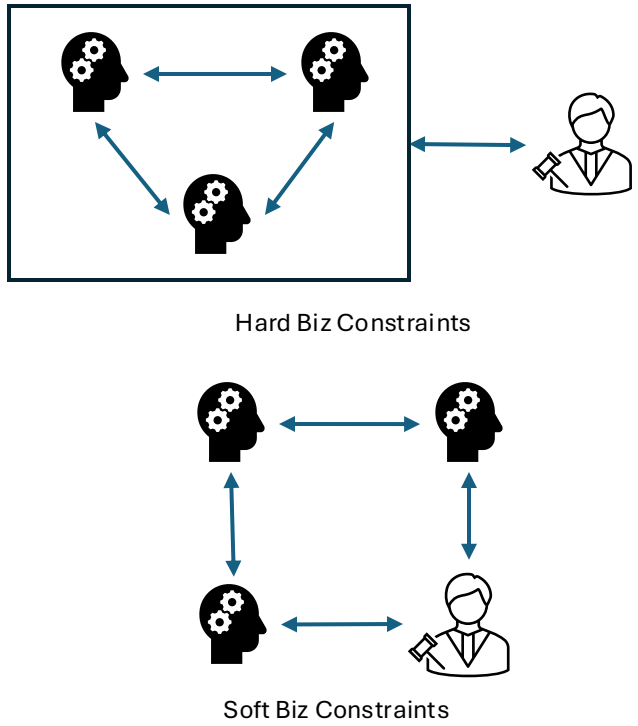
Users expect to get results immediately – so any agentic AI based inferencing should be computed asynchronously

Metrics: What are we trying to solve?

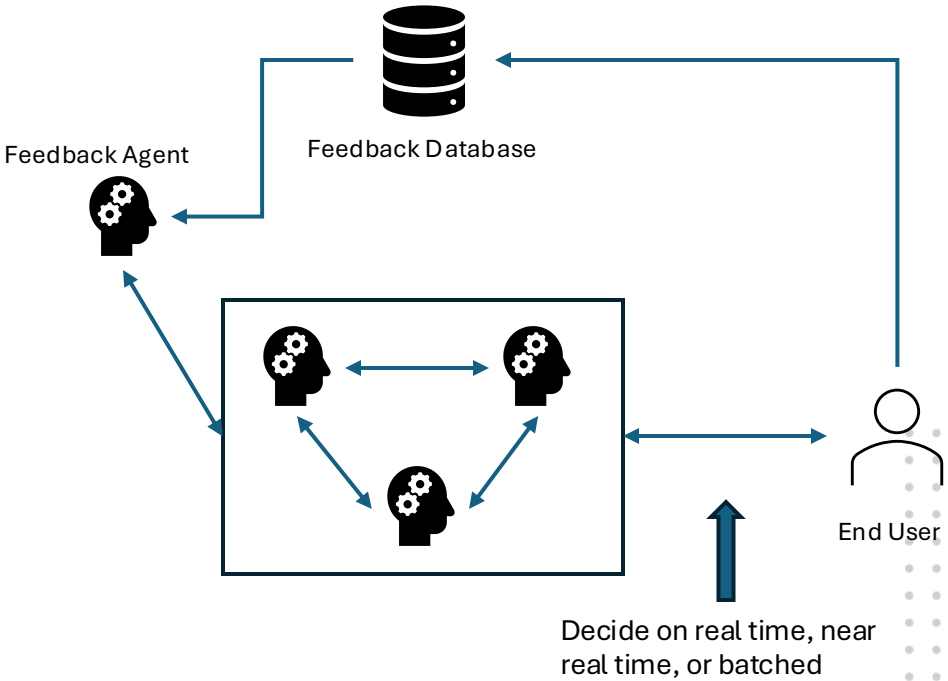
- **Problem Setting:**
  Once the business constraints and the types of acceptable scale/latency are defined, we can build the most reasonable workflow. The workflow should aim to **optimize the metric of interest**.



Define and Build Recommender System using Agentic AI

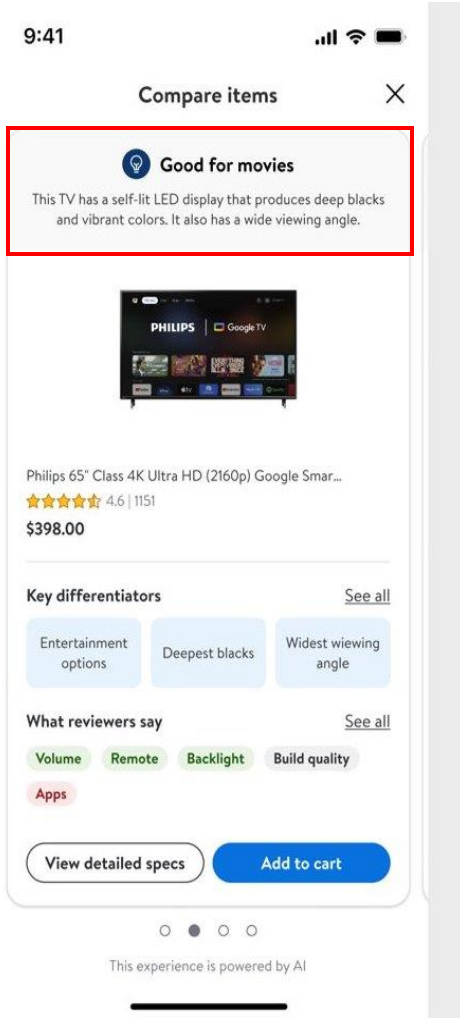


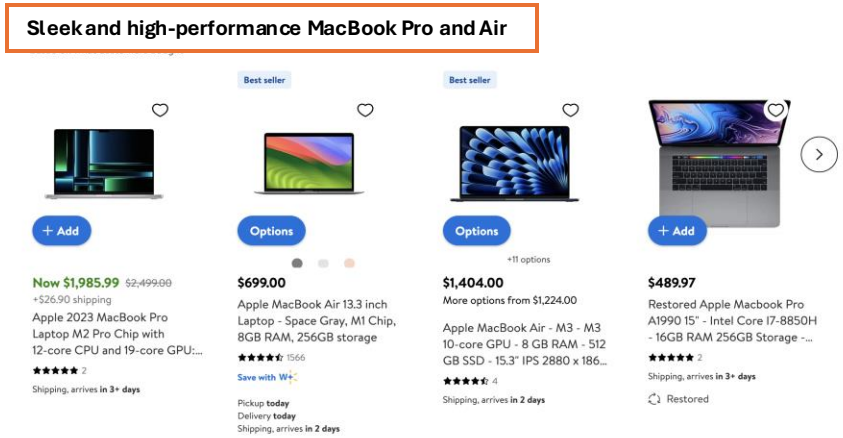
Identify soft & hard biz constraints and incorporate the constraints into the framework accordingly

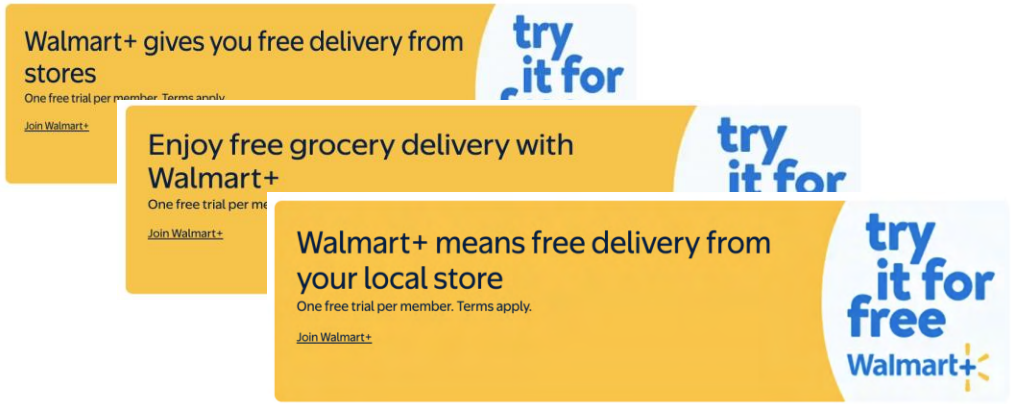


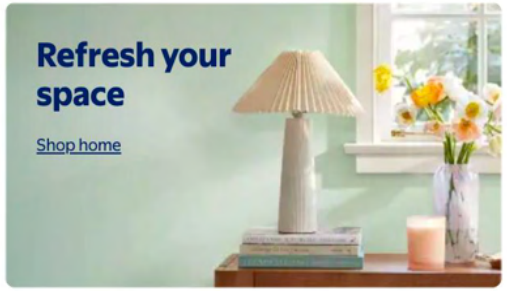
Ensure proper instrumentation & feedback loop to iteratively enhance both user experience and metrics

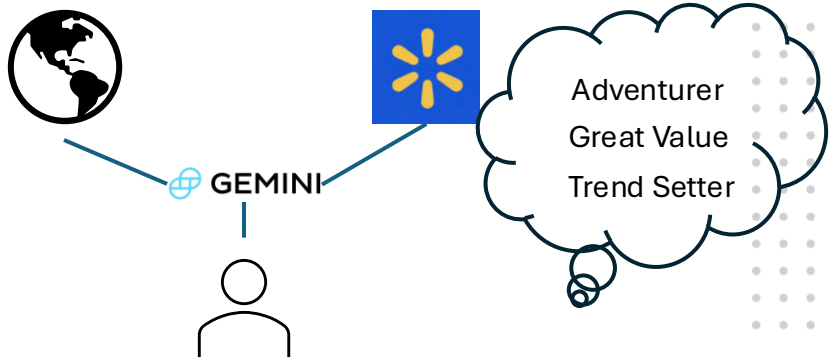
E-comm and Multi-Agentic AI











All powered by Smart Creatives Initiative

Challenges and practical lessons building self-improving agents

Derek Cheng

RecSys Tutorial: Multi-agent Recommendation Systems :: Sep 26, 2025

Disclaimer: I’ve got more questions than solutions.

What do I do?

- Run a research team focusing on:
  - Model & Data Efficiency;
  - Gen AI Monetization, especially for Ads & Commerce.
- End to end from idea, research, prototyping, all the way to production.
- Notable work from my team:
  - Cross-Batch Aggregation for Streaming Learning from Label Proportions. Jonathan Valverde et al. RecSys 2025.
  - Training data efficient LLMs: AskLLM. Noveen Sachdeva et al. -> Gemma & Gemini.
  - Unified Embedding. Ben Coleman & Wang-Cheng Kang et al. -> RecSys / Ads
  - Deep Cross Network V2. Ruoxi Wang et al. -> RecSys / Ads
- One recent focus: building self-improving agents.

Why building self-improving agents is HARD?

- Adding unique value
- Zero-shot v.s. Fine-tuning
- Evaluation
- Scaling
- Productionization

Adding Unique Value

- The water is rising with DeepSeek, Gemini 2.5 Pro, GPT 5, Grok 4, Claude 4, ...
- How do we make sure stuff we build is not yet another Gemini-wrapper?
- Keep adding unique value on top of a base model?

- A winning formula:
  - Success = product definition + problem formulation + innovations + connections + go to market + sales / marketing

Zero shot v.s. Fine-tuning

- Zero-shot can achieve a LOT.
  - Maybe even superhuman capabilities in some cases.
- But is that enough?
- A good example is Cursor:
  - Great company and product, built even more unique value and moat with improved cursor tab w/ online RL.

Evaluation

- Metrics
  - Definition
  - Offline v.s. online
- Human-raters
  - Speed, quality, cost
- Auto-raters
  - Side by side, pointwise

Scaling

- Model Size Scaling
  - A solution worked well with small data, would it scale well with bigger data?
- Feedback Data Scaling
  - We know more RL data is helpful, but could you build a scalable RL feedback loop with fast and accurate responses?
- Task Scaling
  - Would an agent work well on one scenario, generalize well to another scenario?

Productionization

- Scaling down
  - An agent worked well with a bigger model, would it distill well scaling down?
- QPS, latency, # of chips, ROI
- Regression while migrating to new & (supposedly) better models

Thanks!

Hiring research scientists and engineers in LLM & RecSys.
Derek Cheng: zcheng@google.com