Transitioning to Autonomous Discovery Loops in Agentic RecSys

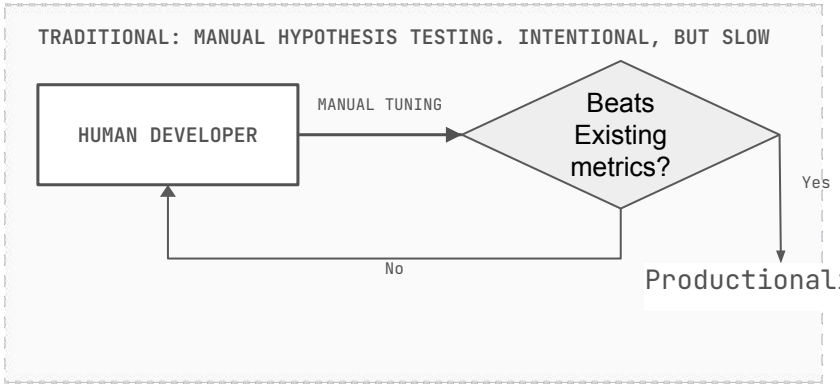
Jason Cho
Pretask.AI

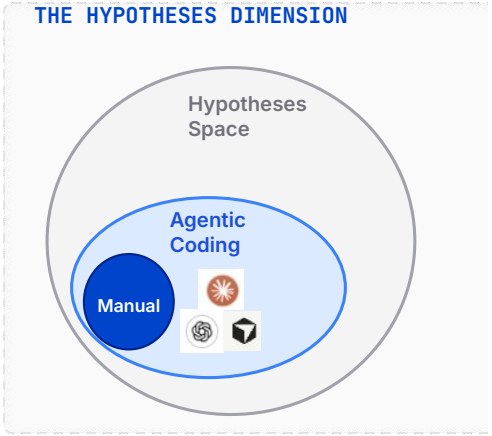
Manual Hypothesis Testing is labor and time intensive

THE HYPOTHESIS SPACE BOTTLENECK

Historically RecSys Applied Scientists have conducted research and picked the top k candidate works to validate against metrics (CTR, NDCG, MRR, etc's).

Recent developments in Agentic Coding did enhance the velocity in proving/disproving hypotheses, but the space still remains large.





How do we bridge the gap between agentic coding enabled space and the larger hypotheses space?

Key Observation: Metric remains fixed, utilize rich literature!

Autonomous Recommendation Discovery

Google DeepMind 2026-5-22

Advancing Mathematics Research with AI-Driven Formal Proof Search

George Tsoukalas[1†], Anton Kovsharov[1†], Sergey Shirobokov[1†], Anja Surina[1], Moritz Firsching[1], Gergely Bérczi[2], Francisco J. R. Ruiz[1], Arun Suggala[1], Adam Zsolt Wagner[1], Eric Wieser[1], Lei Yu[1], Aja Huang[1], Miklós Z. Horváth[1], Andrew Ferrauiolo[1], Henryk Michalewski[1], Codrut Grosu[3], Thomas Hubert[1], Matej Balog[1], Pushmeet Kohli[1] and Swarat Chaudhuri[1†]
[1]Google DeepMind[1], [2]Aarhus University, [3]Google

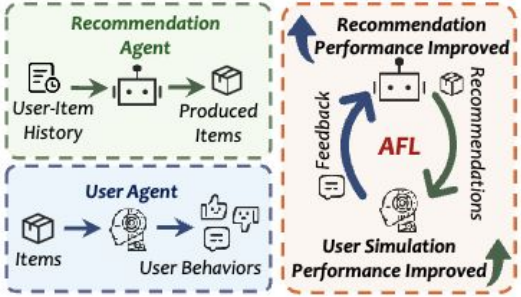
Large language models (LLMs) increasingly excel at mathematical reasoning, but their unreliability limits their utility in mathematics research. A mitigation is using LLMs to generate formal proofs in languages like Lean. We perform the first large-scale evaluation of this method's ability to solve open problems. Our most capable agent autonomously resolved 9 of 353 open Erdős problems at the per-problem cost of a few hundred dollars, proved 44/492 OEIS conjectures, and is being deployed in

2024-9-4

The AI Scientist: Towards Fully Automated Open-Ended Scientific Discovery

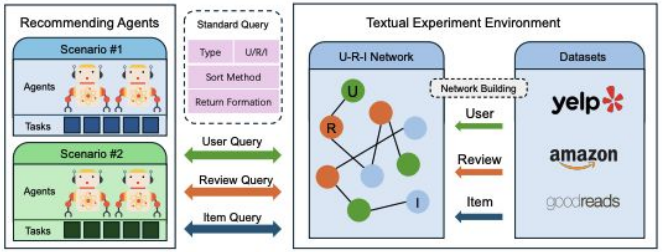
Chris Lu[1,2,*], Cong Lu[3,4,*], Robert Tjarko Lange[1,*], Jakob Foerster[2,†], Jeff Clune[3,4,5,†] and David Ha[1,†]
[*]Equal Contribution, [1]Sakana AI, [2]FLAIR, University of Oxford, [3]University of British Columbia, [4]Vector Institute, [5]Canada CIFAR AI Chair, [†]Equal Advising

One of the grand challenges of artificial general intelligence is developing agents capable of conducting scientific research and discovering new knowledge. While frontier models have already been used as aides to human scientists, e.g. for brainstorming ideas, writing code, or prediction tasks, they still conduct only a small part of the scientific process. This paper presents the first comprehensive framework for fully *automatic scientific discovery*, enabling frontier large language models (LLMs) to perform research independently and communicate their findings. We introduce THE AI SCIENTIST, which generates novel research ideas, writes code, executes experiments, visualizes results, describes



Agentic Feedback Loop Modeling Improves Recommendation and User Simulation, Shihao Cai et al (SigIR 25)

User Agent simulates a reaction (critique, click, or rejection), and the Recommender iteratively refines its internal strategy and policy based on that mutual feedback before finalizing a result.

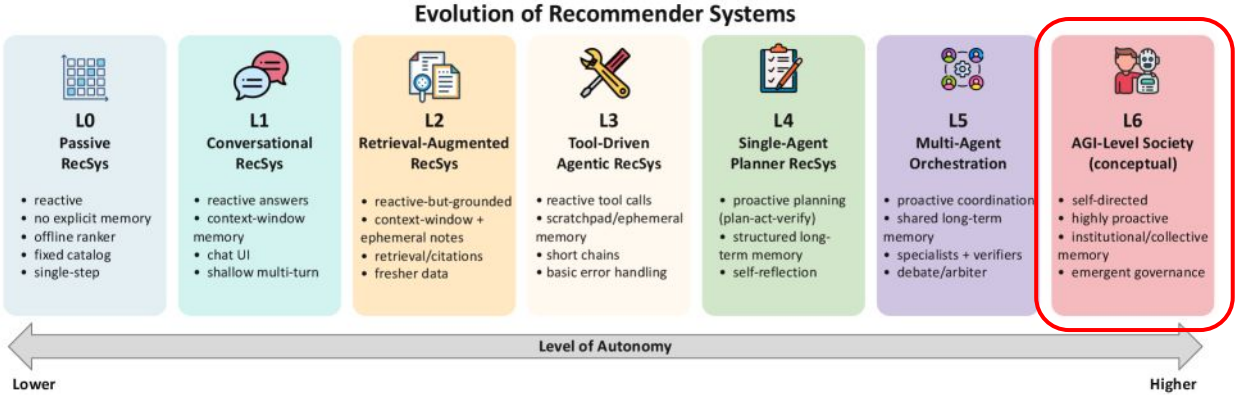


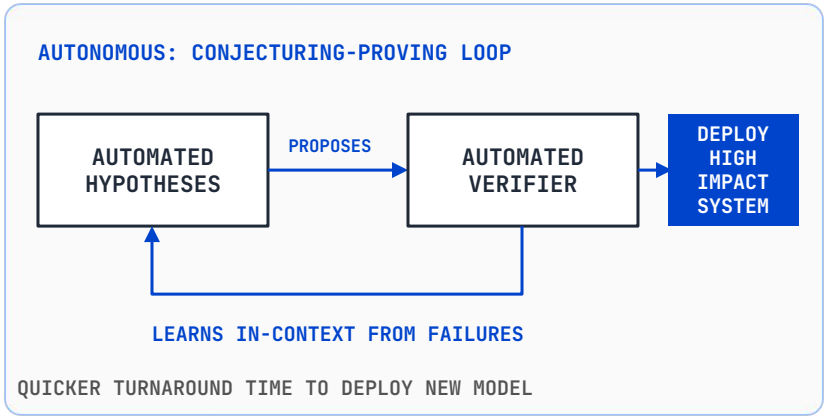
AgentRecBench: Benchmarking LLM Agent-based Personalized Recommender Systems, Shang et al (NeuRIPS 25)

Introduces a benchmark specifically built because traditional RecSys datasets fail to capture agent capabilities.

Autonomous agents are already making scientific discoveries. Recommender Systems are also driving innovations - both from technical sophistications and benchmark test data

Autonomous Recommendation Discovery





RecSys literature has predefined set of metrics

Typical Industrial RecSys focuses on predefined set of metrics:

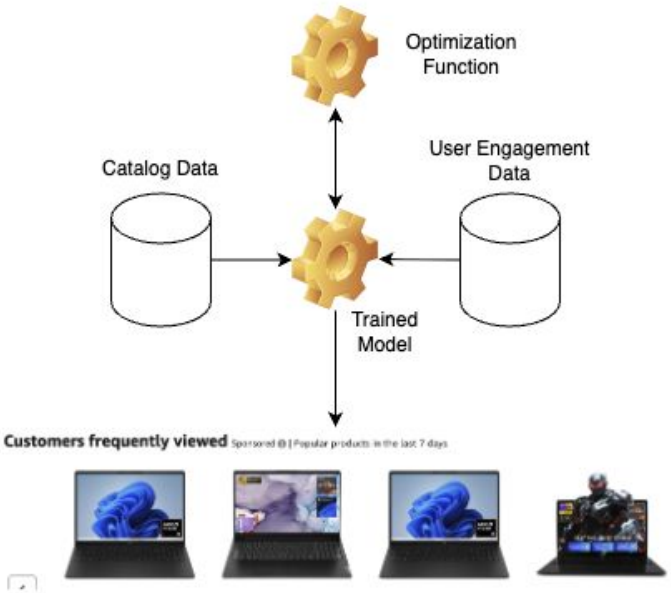
- NDCG/MRR
- Precision/Recall
- Diversity
- Cold-Start

Typical applied ML/AI scientists focus on optimizing the same set of metrics by experimenting both existing and novel feature sets and recommender system models.

Predefined objective function enables dynamically exploring hypothesis space.

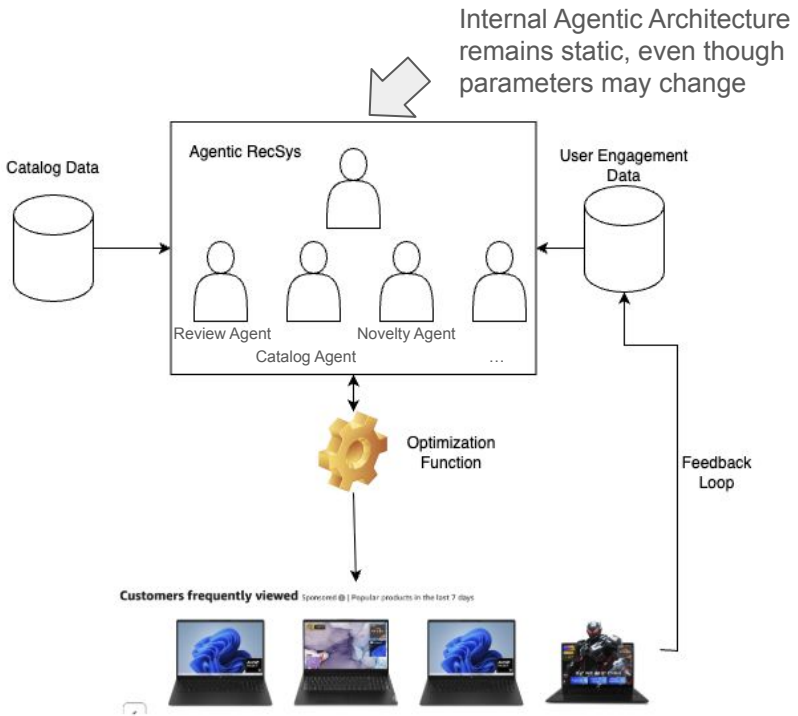
Autonomous Information Seeking: A Roadmap for Agentic Recommender Systems, Lin et a
https://www.langchain.com/blog/the-art-of-loop-engineeringl

Traditional RecSys



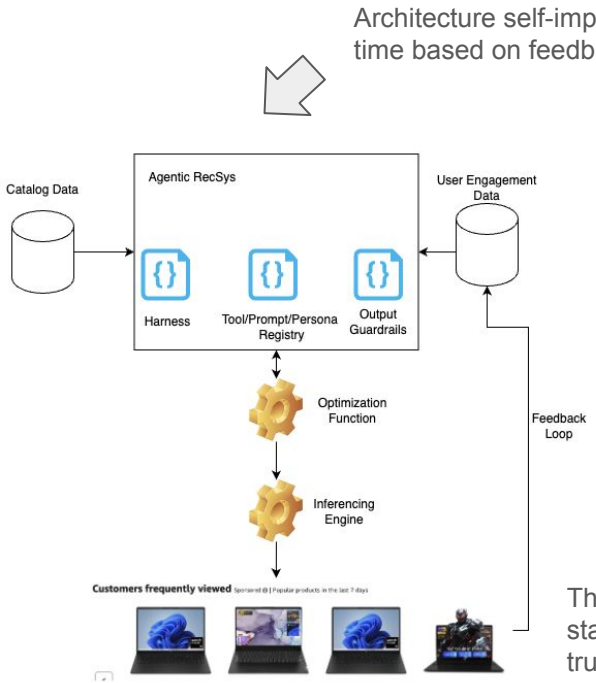
Researchers finetune & experiments with different variations of the model

Multi Agentic Orchestrator



Researchers comes up with agentic architecture for recommendation engine ranking & retrieval

Self Improving RecSys



The inference engine can be static (batched prediction), or truly real time & dynamic.

Researchers preset goals and constraints, agents figures out the best recommendation engine for ranking & retrieval

The Harness: State, Tools, and Boundaries

The Harness provides the essential MLOps scaffolding that equips a Large Language Model with state, tools, and boundaries. It replaces standard context windows with dynamic memory and active tools.

01. DYNAMIC MEMORY

Replaces standard context windows, connecting the agent directly to User Profiles and Item Vector Databases.

02. ACTIVE TOOLS

Empowers the agent to query external system states, including Feature Stores and the Inventory API.

03. HARDCODED GUARDRAILS

Enforces absolute business logic and brand safety (such as filtering out-of-stock items) that the model cannot override.

RL ENVIRONMENT ANALOGY

The Harness functions as a reinforcement learning environment. It dictates exactly what the recommendation agent can observe and interact with, establishing hard boundaries for autonomous actions.

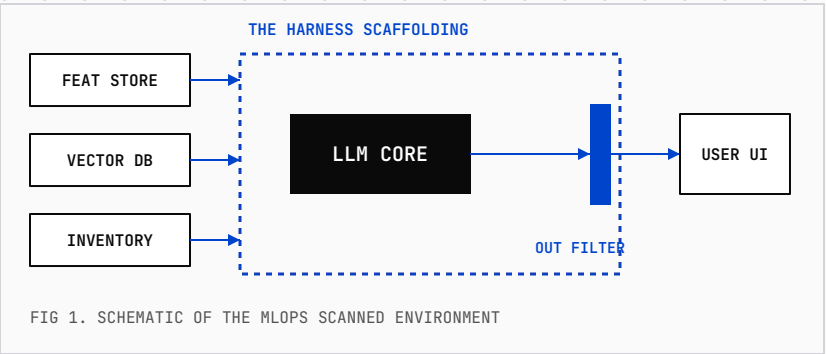


FIG 1. SCHEMATIC OF THE MLOPS SCANNED ENVIRONMENT

Agentic Feedback Loops

[01] THE LOOP DEFINITION

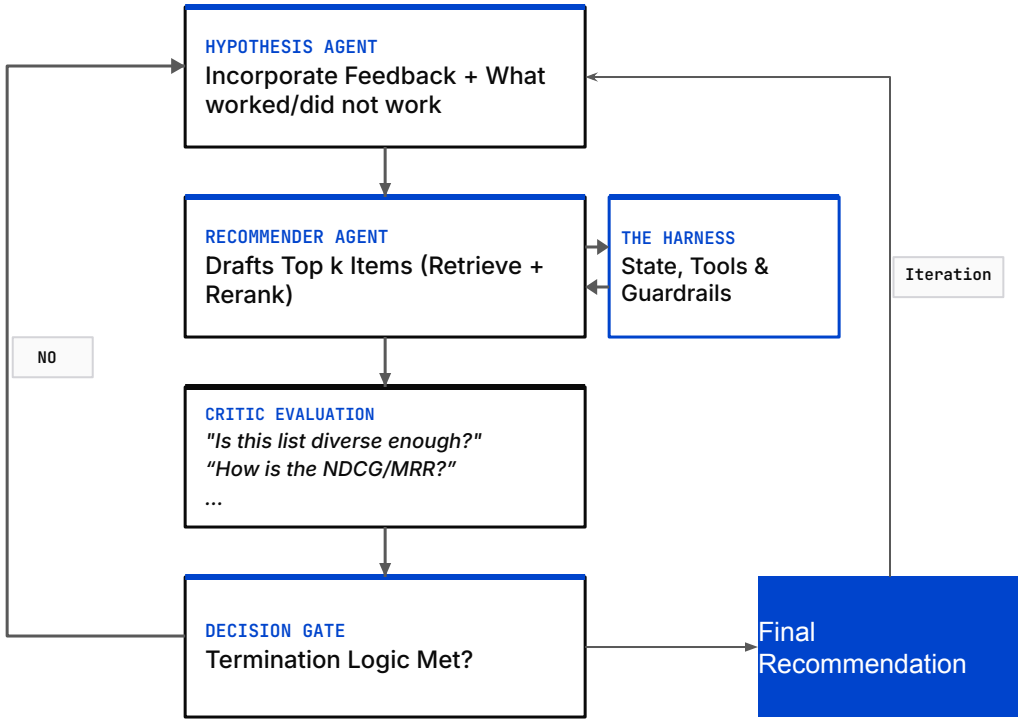
The Loop is the automated control flow governing how an agent acts, observes, and iterates before returning a final result.

[02] AGENT-CRITIC DYNAMICS

A Recommender Agent drafts a slate, which a Critic Agent evaluates against diversity and relevance metrics. This cycle acts as policy optimization, forcing the agent to evaluate its own recommendations.

[03] TERMINATION LOGIC

Iterative refinement continues until specific termination logic is satisfied, such as reaching a specific metric threshold or exhausting a compute budget.



Agentic Feedback Loops: GEPA

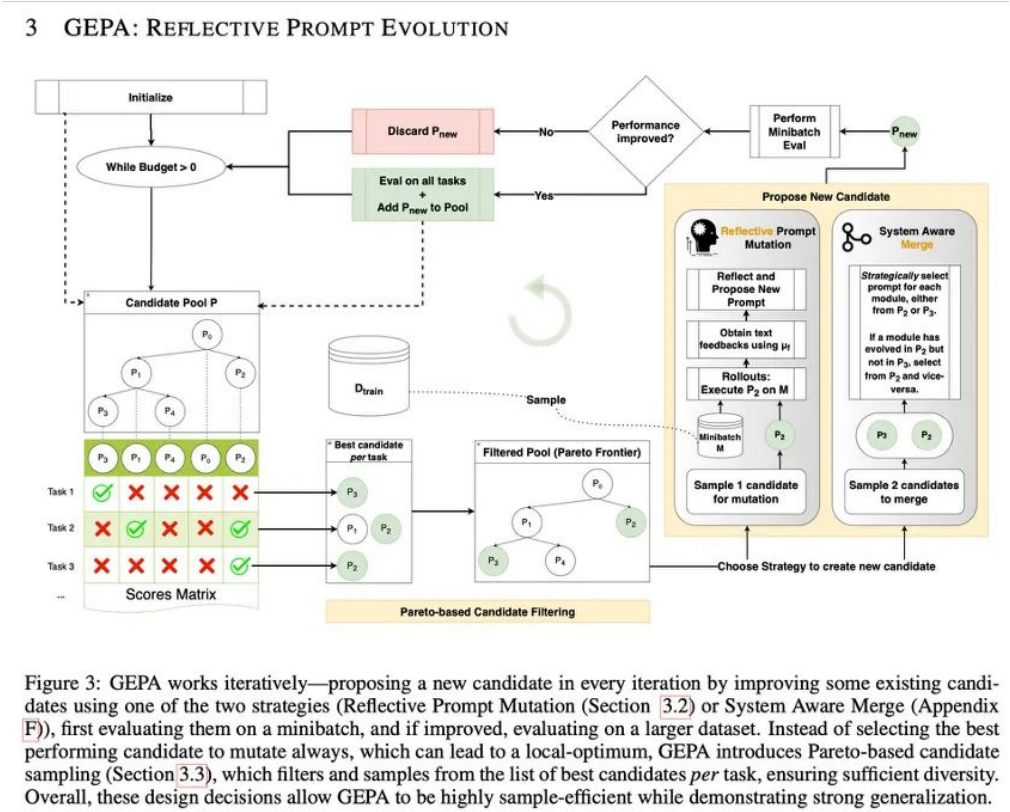


Figure 3: GEPA works iteratively—proposing a new candidate in every iteration by improving some existing candidates using one of the two strategies (Reflective Prompt Mutation (Section 3.2) or System Aware Merge (Appendix F)), first evaluating them on a minibatch, and if improved, evaluating on a larger dataset. Instead of selecting the best performing candidate to mutate always, which can lead to a local-optimum, GEPA introduces Pareto-based candidate sampling (Section 3.3), which filters and samples from the list of best candidates *per* task, ensuring sufficient diversity. Overall, these design decisions allow GEPA to be highly sample-efficient while demonstrating strong generalization.

High Level Approach:

- Mutate prompts, workflow and orchestration process
- Keep successful candidates
- Merge complementary discoveries

Applicable for both multi-agentic recommendations & self-improvement loops

Shifting the RecSys Paradigm

Autonomous multi-agent loops vs. traditional manual experimentation

Customer intent is hyper-dynamic. Traditional models optimizing for historical averages struggle with sudden trend shifts. In contrast, the recommendation hypothesis space contains thousands of potential strategies.

Traditional Model: Manual A/B Testing (Slow)

THROUGHPUT & SPEED

Deploys only one new retrieval & ranking strategy per sprint. Manual bottleneck limit.

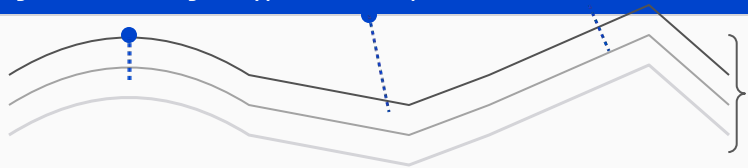
ADAPTATION SCOPE

Struggles with sudden shifts; optimizes heavily for static historical averages.

CORE PARADIGM

Predictive: Focused strictly on guessing what a user wants beforehand.

Agentic Paradigm: Hypothesis Exploration



THROUGHPUT & SPEED

Explores numerous distinct retrieval & ranking strategies in parallel autonomously (Interleaving / Causal Inferencing)

ADAPTATION SCOPE

Navigates thousands of retrieval/ranking pathways dynamically in real-time.

CORE PARADIGM

Exploratory: Actively discovers how to best serve users per session.

Causal Validation of Agentic Systems

Methodology & Obstacles

01 / High Variance Noise

Because agent outputs have high variance, standard A/B tests become noisy and unreliable. Traditional statistical metrics fail to capture structured reasoning paths.

02 / Surrogate & Uplift Modeling

Practical validation requires treating the agent's logic as a causal treatment effect. Uplift modeling isolates whether specific reasoning caused the conversion.

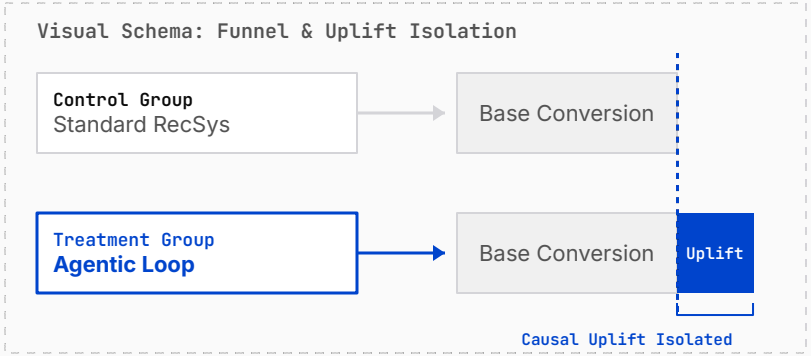
03 / Shadow Mode Baselines

Running loops in shadow mode parallel to production models allows engineering teams to compare reasoning logs directly against actual outcomes to establish a baseline.

Proof: Causal Treatment Effect

Treating the agent's reasoning chain as an active intervention $T \in \{0,1\}$, the causal uplift is formulated as:

$$\tau_i = E[\, Y_i(1) - Y_i(0) \mid X_i \,]$$



Interleaving Tests: Enabling Rapid Experimentation

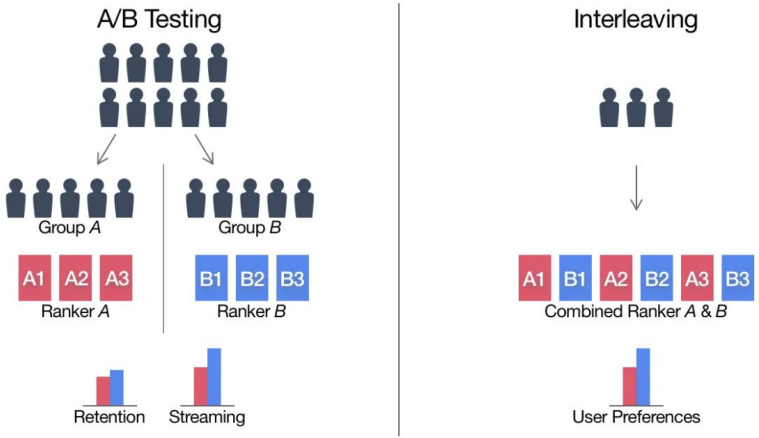
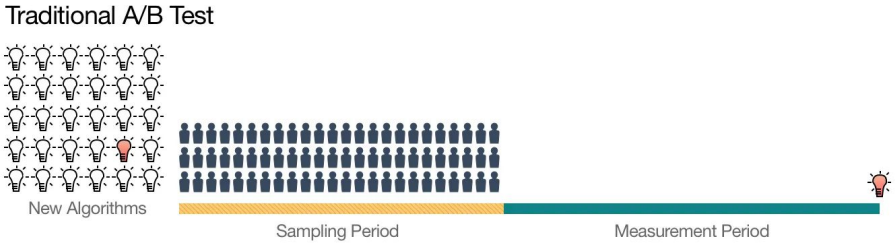
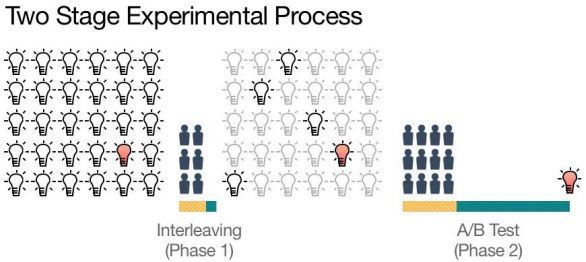


Fig. 3: A/B Testing vs. Interleaving. In traditional A/B testing, the population is split into two groups, one exposed to ranking algorithm A and another to B. Core evaluation metrics like retention and streaming are measured and compared between the two groups. In contrast, interleaving exposes one group of members to a blended ranking of rankers A and B. User preference for a ranking algorithm is determined by comparing the share of viewing hours coming from videos recommended by rankers A or B.





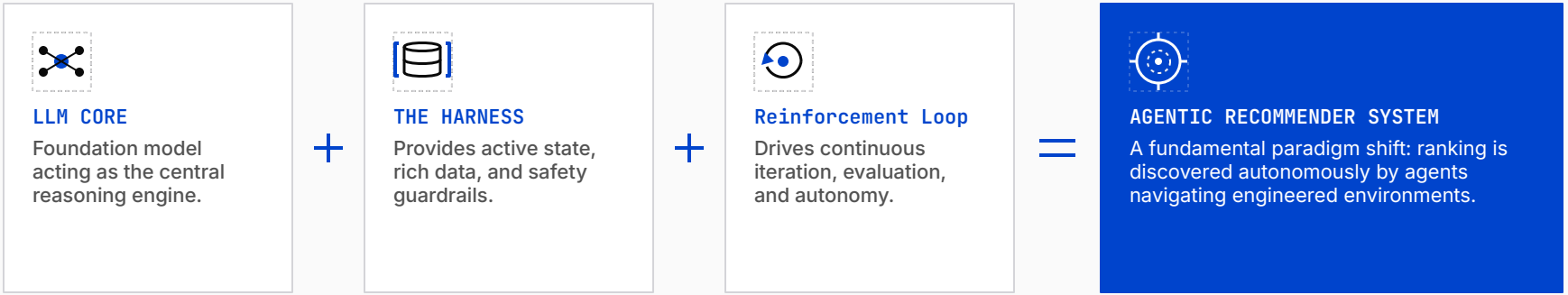
https://netflixtechblog.com/interleaving-in-online-experiments-at-netflix-a04ee392ec55

Evaluating Agentic RecSys

The critical shift from static item-ranking outcomes to dynamic, process-level execution traces

Traditional RecSys Metrics	Agentic RecSys Metrics
01. CLICK-THROUGH RATE (CTR) Tracks immediate, binary user interactions. Completely ignores the agent's internal reasoning loop, context usage, and long-term trajectory viability.	01. TOKEN EFFICIENCY Measures the exact context window and LLM token cost consumed per successful user conversion loop to manage processing overhead.
02. DISCOUNTED CUMULATIVE GAIN (NDCG) Assesses isolated, position-based list ranking quality. Blind to multi-step tool invocations, active tool accuracy, or conversational pathing.	02. TRAJECTORY SUCCESS RATE Monitors the percentage of agent actions that navigate to a valid, helpful output without encountering fallback errors or looping indefinitely.
03. RECALL AT K Evaluates if relevant items appear within the top K results. Misses the user's explicit reasoning alignment and active feedback during generation.	03. SEMANTIC ALIGNMENT Deploys an LLM-as-a-judge framework to systematically grade the correctness of the inner reasoning trace against real-world user engagement.

The Agentic RecSys Formula



A Fundamental Paradigm Shift in Personalization: Engineering leaders no longer write static algorithms to rank items; instead, they engineer the environments, tools, and incentives that allow autonomous agents to discover the best rankings themselves.

Real world Simulating, Retrieving, and Generating Recommendations

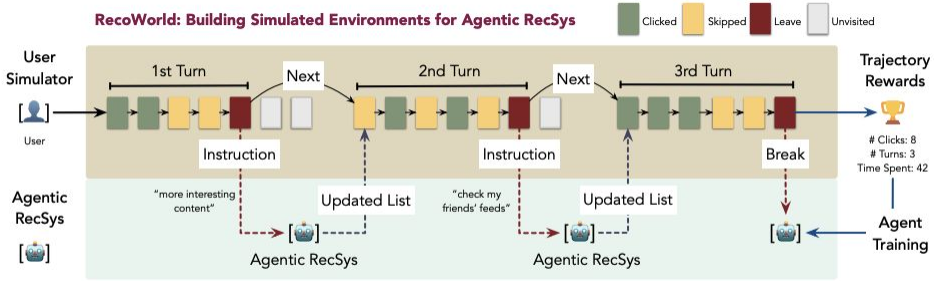
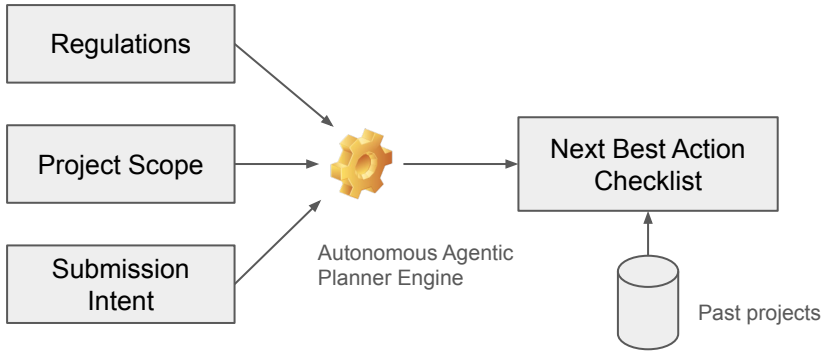
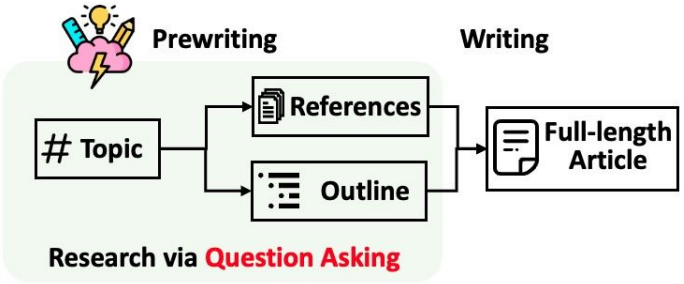


Figure 1: A simulated user interacts with an agentic RecSys over multiple turns within a session.

RecoWorld: Simulated Environment for Agentic Recommender Systems



Pretask: Autonomous Next-best-action Recommendation



Storm: Automated Article Writing

Agentic Frameworks

AG2 crewai LangGraph MassGen

Agent Memory

mem0 CLOUDFLARE

Tools used for Agentic Recommendations

Considerations in Building Practical Agentic AI Recommender Systems

Jason Cho
Pretask.AI

*Work done at Walmart Global Tech

Pretask.AI

Considerations in Building Successful Multi-Agentic AI Recommender System

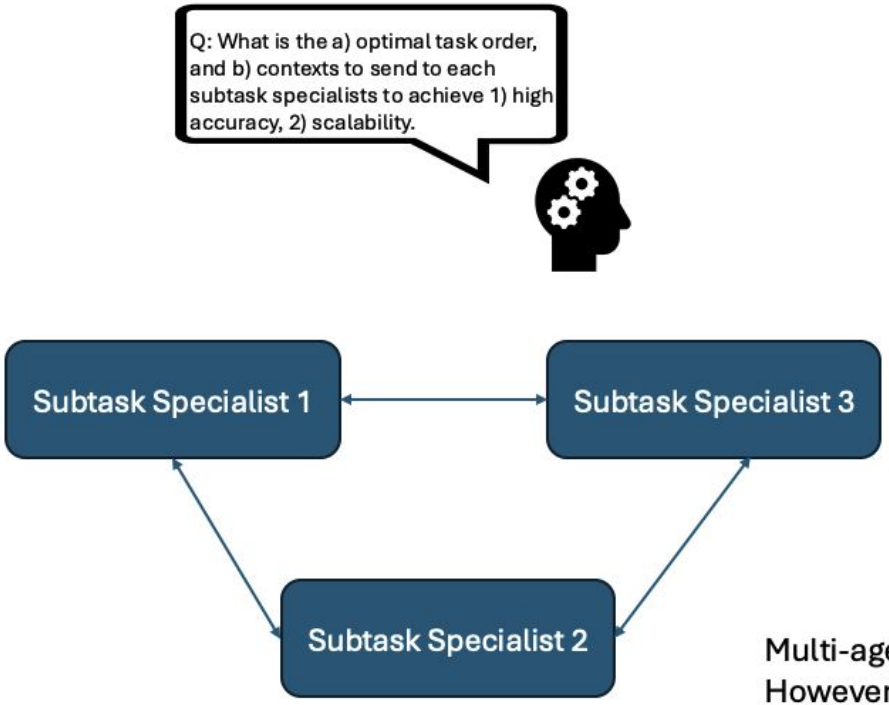
- **Background:** Prior to the introduction of GenAI, customer facing typically had limited world knowledge outside of pretrained embeddings. This limited 1) ***dynamic content generation***, and 2) ***building an even more powerful ML-driven solutions***. GenAI provided path to solving these. However…

Constraints:

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

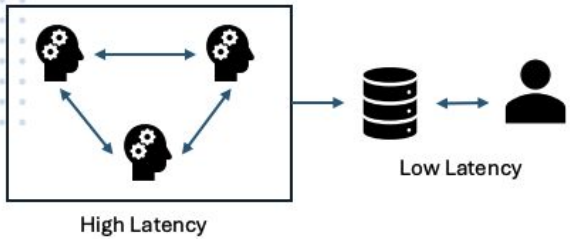
- What is our brand's voice?
  - Friendliness, down-to-earth or formal, etc's
- Ensure consistent tone and grammar

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

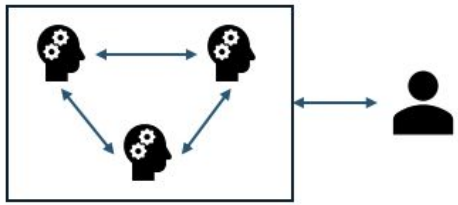
- **Problem Setting:**
  Agentic Framework takes time for real time inferencing. Identify whether the end user expects to wait couple of seconds vs they expect an immediate response.



Batched Inferencing

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

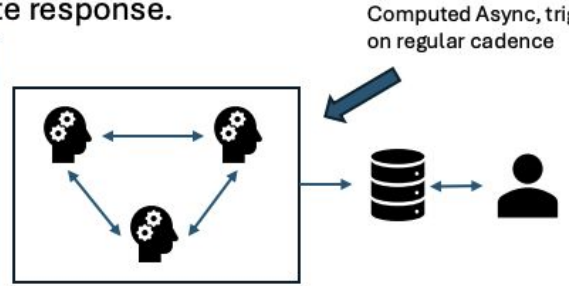
Users expect to get results immediately – so any agentic AI based inferencing should be computed offline



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.



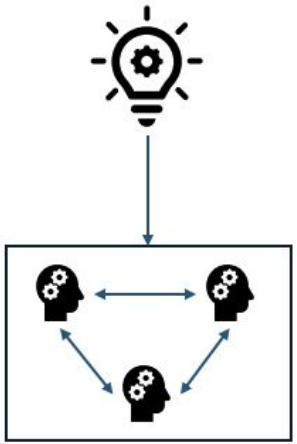
Hybrid (Near-real time)

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

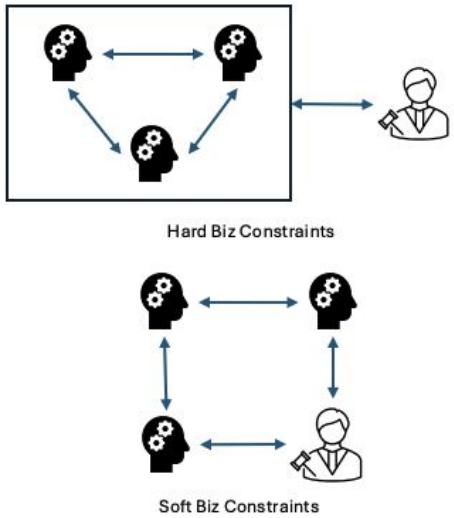
Users expect to get results immediately – so any agentic AI based inferencing should be computed asynchronously

Metrics: What are we trying to solve?

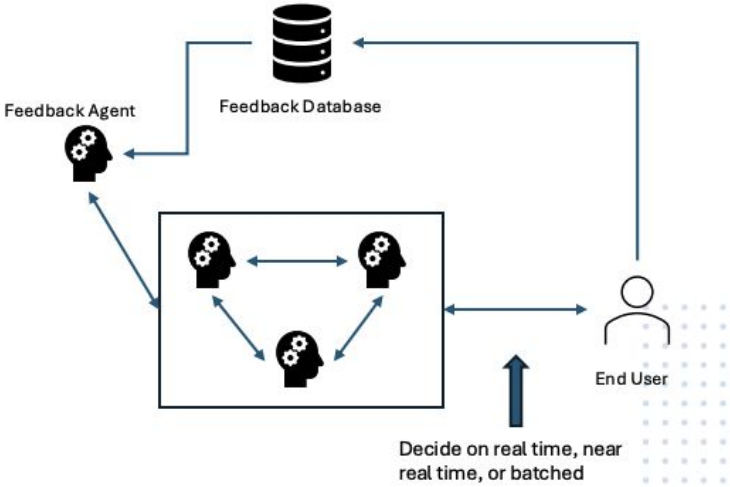
- **Problem Setting:**
  Once the business constraints and the types of acceptable scale/latency are defined, we can build the most reasonable workflow. The workflow should aim to **optimize the metric of interest**.



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

Before we go…

The AI Agent That Cost $47,000 While Everyone Thought It Was Working

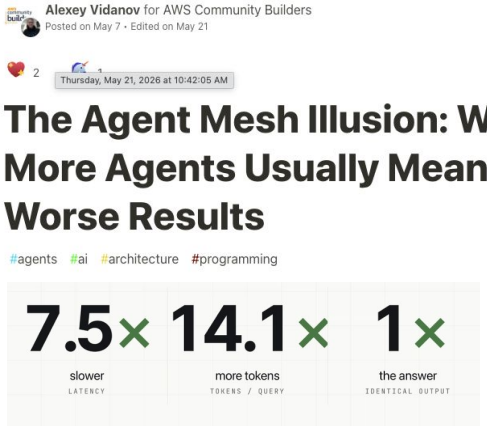
#ai #aiops #agents

For eleven days, a multi-agent research system sat in production doing exactly what it was designed to do: agents talking to agents, processing requests, passing messages. Dashboards showed activity. Latency looked normal. No errors fired.

The system was healthy. Except it wasn't doing anything useful. Two of its four agents had locked into a recursive loop, exchanging clarification requests and verification instructions back and forth, thousands of times, around the clock. By the time anyone looked at the invoice, the bill was $47,000.

Nobody on the team knew until the cloud bill arrived.

Improperly set up Agentic System



Complexity for the complexity's sake

Uber Blew Its Entire AI Budget in 4 Months. Here's What Finance Teams Can Learn.

July 6, 2026 | Cody Leach, CPA | Accounting

Uber had rolled out Anthropic's Claude Code to roughly 5,000 engineers, and adoption surged from 32% of the engineering org in February to 84% by March. Nearly 95% of engineers were using AI tools monthly, meaning that token consumption was off the charts.

Wrong KPI's, analogous to LoC as metric

Run cost/benefit analysis to decide when and when not to run agentic system for self-improving systems

Let's connect!

Ben and Chi will cover the second part of the Tutorial.

Ben will cover Self improving memory and learned experiences, and Chi will cover AgentOS (AG2 & MassGen).

We will resume at 11:05am.

agenticrecsys.github.io

We will upload all the presentation files