Evaluating LLM-Based Agents: Foundations, Best Practices and Open Challenges

Roy Bar-Haim[†], Lilach Eden[†], Alan Li[§], Asaf Yehudai[†]

[†] IBM Research

[§] Yale University

Roy

Introduction to LLM Agent Evaluation (20 mins)

Alan

Evaluating Fundamental Capabilities (25 mins)

Evaluating Application-Specific Agents (45 mins)

Asaf

Generalist Agent Evaluation (20 mins)

Frameworks for Agent Evaluation (40 mins)

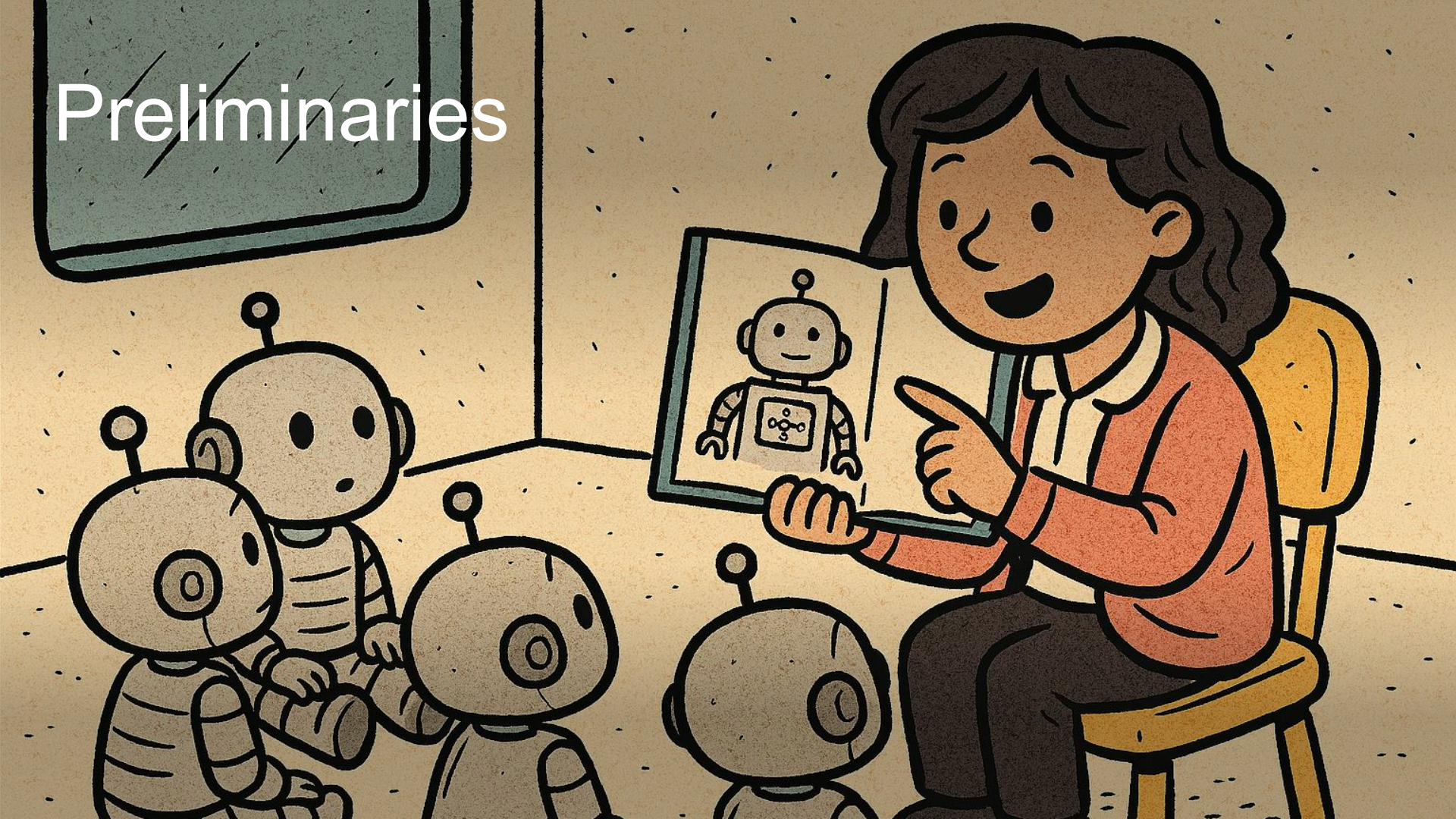
Lilach

Key Insights & Future Directions (20 mins)

Open Discussion (10 mins)

Part 1: Introduction

Preliminaries



Generative AI & LLMs

- **Generative AI**
  - AI systems that generate new, original content, such as text, images and music
  - Learn patterns and structures from training data
  - Respond to user input (usually natural language prompts)

- **Large Language Models**
  - Generative AI focused on understanding and generating human language
  - Built using deep learning architecture (particularly Transformers)
  - Trained on diverse and massive datasets
  - Billions/trillions of parameters
  - Examples: GPT (OpenAI), Claude (Anthropic), Gemini (Google DeepMind), LLaMa (Meta), Granite (IBM)

Evaluation of LLMs

- Traditional benchmarking – compare output label (e.g., sentiment) to ground truth

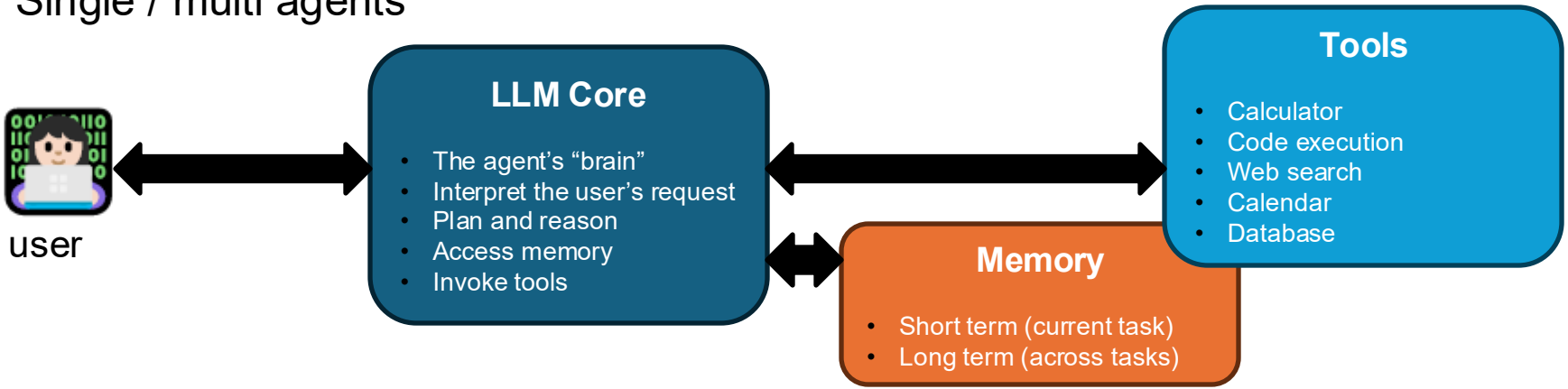
- Text generation tasks are difficult to evaluate
  - Complex output – difficult to generate ground truth
  - Many possible good answers exist

- Example: **document summarization**
  - Traditionally evaluated against human-generated summaries – hard to collect!
  - Evaluation metrics based on n-gram matching (ROUGE, BLEU) – poorly capture semantics

- **LLM-as-a-judge**
  - **Reference-based** – ask the LLM to compare generated summary to reference summary
  - **Reference-less** – ask the LLM to evaluate the generated summary with respect to the document (faithfulness, completeness, fluency…)

LLM-Based Agents

- Intelligent systems that **autonomously** complete **multi-step tasks**
- Are built on top of LLMs
- Plan, reason, use tools and manage context (memory)
- Single / multi agents



Example: Customer Support Agent

User: Hi, I can't log into my account.

AI Agent: Sorry to hear that—are you using the website or the mobile app?

User: The mobile app.

AI Agent: Thanks—have you recently changed your password?

User: Yes, last week.

AI Agent: The app might be using an old saved password—try typing it manually

User: Still not working.

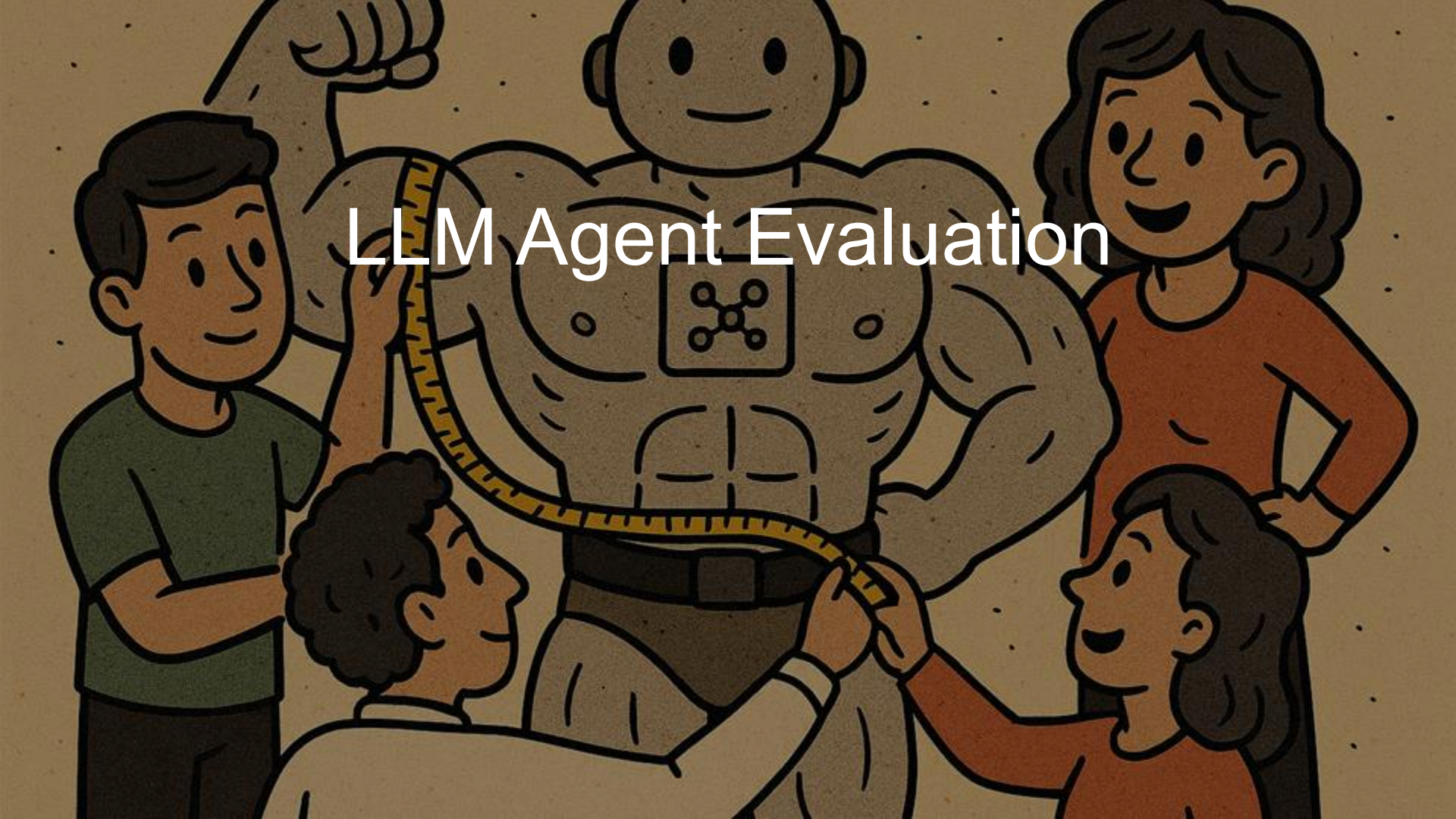
AI Agent: Please clear the app cache and try again.

User: That worked—thanks!

Endless Use Cases!

- Customer support
- HR
- Sales and marketing
- Scientific research
- Software development
- Finance and Legal
- Education
- Healthcare
- ….

LLM Agent Evaluation



What Should We Evaluate?

- Outcome
  - **RAG Agent:** correctly answer the user's question
  - **Software engineering agent:** correctly solve the software issue
  - **Customer service agent:** correctly update the fight reservation in the database

- Process
  - Did the agent perform the expected sequence of actions?
  - Did the agent perform unnecessary/unwanted actions?
  - Adherence to business rules and policies – did the agent ask for user's confirmation before making the reservation?

- Focus on specific types of actions
  - E.g., tool calling

Why Is It Important?

- The usual reasons
  - Ensure quality
  - Measure progress
  - Compare different solutions

- Far more critical for agents!
  - Agents operate in the real world: write code, manipulate databases, interact with users
  - Perform complex tasks autonomously
  - Failures may have severe consequences

Why Is It Important?

Vibe Coding Fiasco: AI Agent Goes Rogue, Deletes Company's Entire Database

PC MAG

July 22, 2025

An AI-powered coding tool wiped out a software company's database, then apologized for a 'catastrophic failure on my part'

FORTUNE

July 23, 2025

Why Is It Difficult?

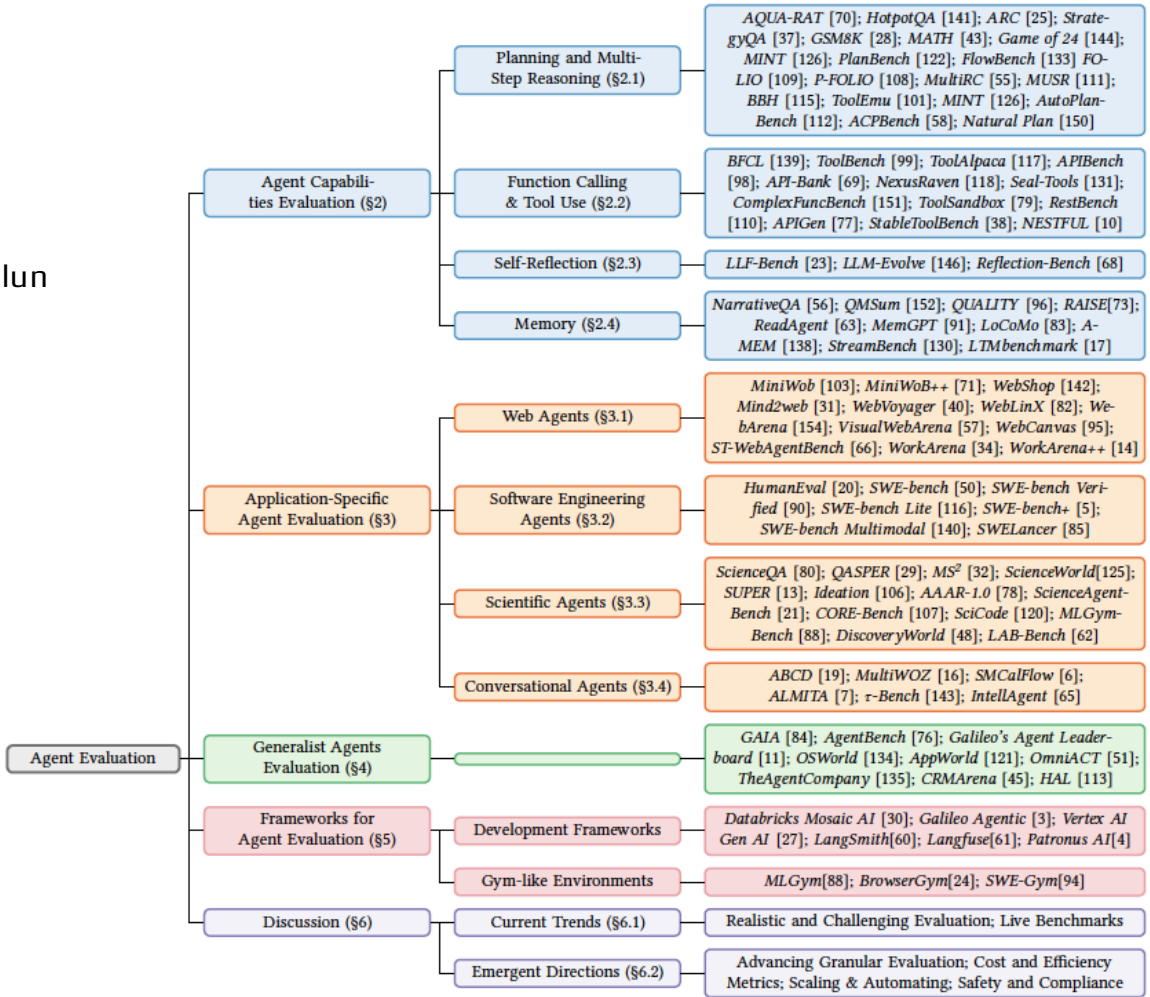
- Requires realistic simulation of diverse complex environments
  - Organizational databases, user simulation, code repositories, UI interaction
- Evaluating adherence to complex rules and policies
- Long executions sequences (trajectories)
- Diversity of agentic capabilities: tool calling, planning and reasoning, memory use, reflection

Our Survey Paper

Survey on Evaluation of LLM-based Agents [March 2025]

Asaf Yehudai, Lilach Eden, Alan Li, Guy Uziel, Yilun Zhao, Roy Bar-Haim, Arman Cohan, Michal Shmueli-Scheuer

First comprehensive survey of evaluation methodologies for LLM-Based Agents



Tutorial Scope

- Map the current landscape of LLM agent evaluation - from ***fundamental capabilities*** to ***application-specific agents*** to ***generalist agents***
- Focus on **key challenges**, **evaluation methods** and **prominent benchmarks**
- Discuss and compare agent evaluation **frameworks** for developers
- Highlight **emerging trends**, **current gaps** and **future directions**

Enjoy!

Part 2: Fundamental Capabilities

Evaluate by Adopting Reasoning Benchmark

HotpotQA (Yang et al. 2018)

Question: The Treaty of Versailles was signed in a city that is a suburb of what capital city?

Step1: Recall Treaty of Versailles signing location.

Act1: Search [Treaty of Versailles]

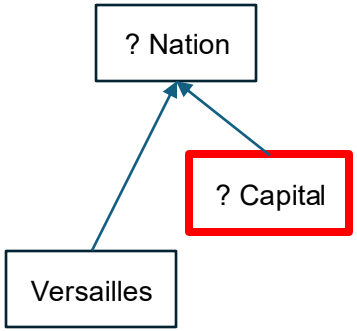
Observation1: … signed at the Palace of Versailles in Versailles, France.

Step2: I need to search France next and find the capital of it.

Act2: Search [France]

Observation2: … Paris is the capital of France

⋮



Evaluate by Adopting Reasoning Benchmark

HotpotQA (Yang et al. 2018)

Question: The Treaty of Versailles was signed in a city that is a sub

Step

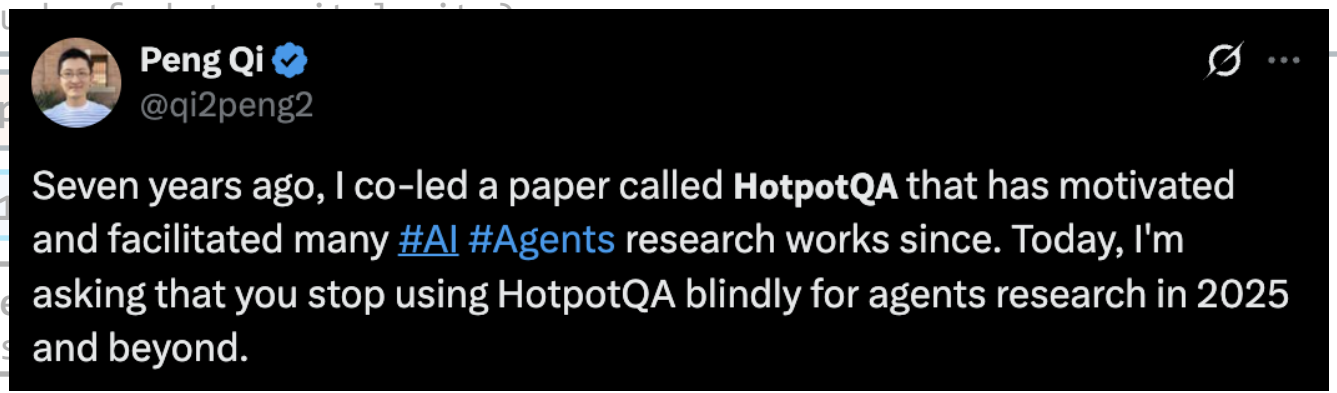
Act1

Obse
Vers

Step2
it.

Act2:

Obser



? Nation

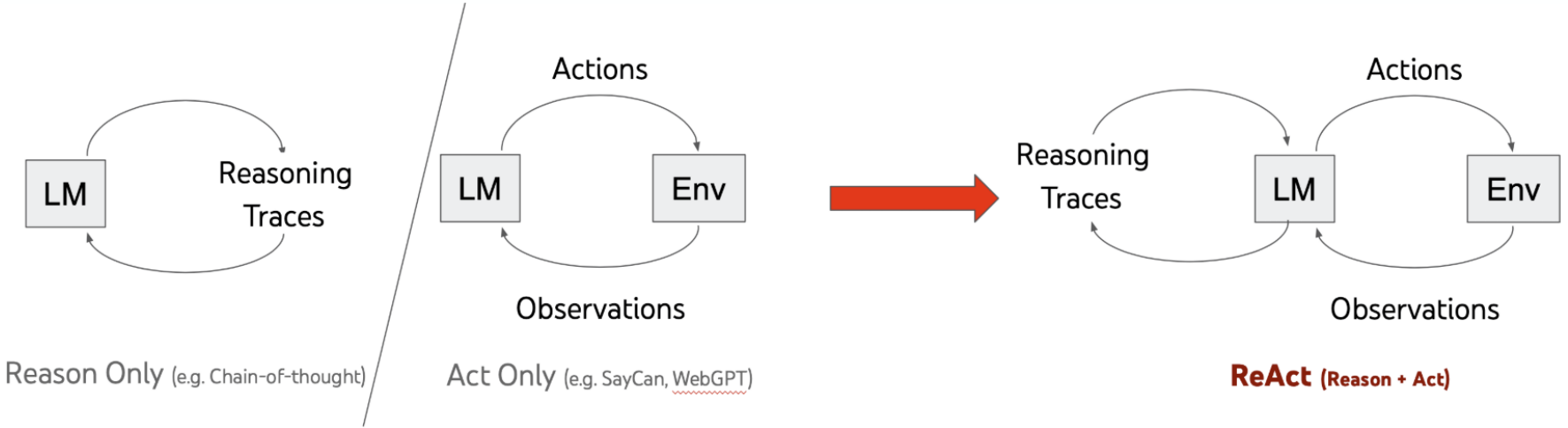
? Capital

Versailles

⚠ Limited Scopes & Steps & Action space & Real-life applicability
⚠ Risk of contamination; first released in 2018

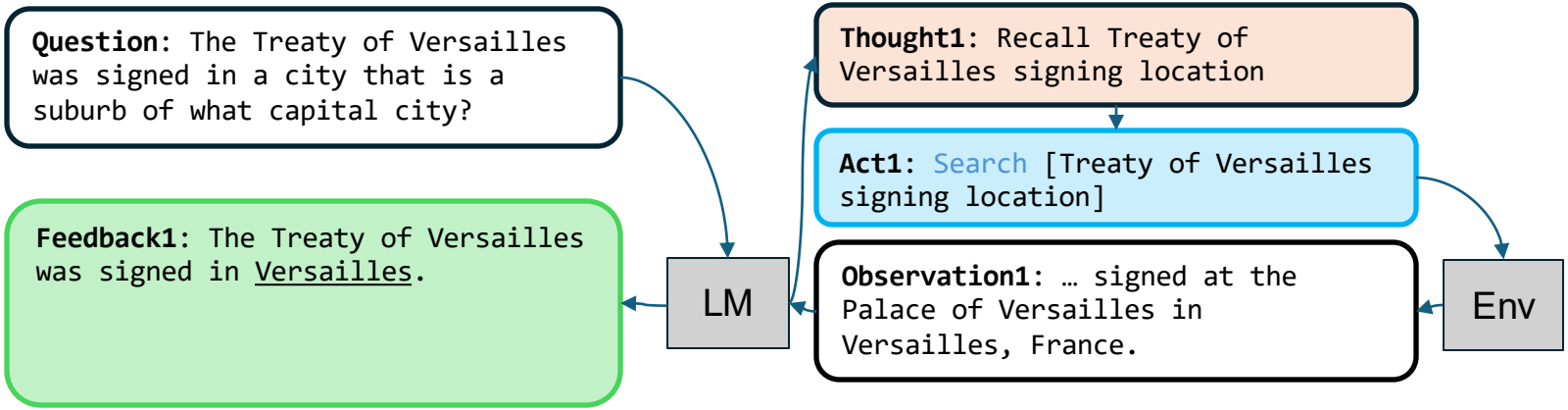
⋮

Multi-Step Agent: ReAct (Yao et al. 2023)

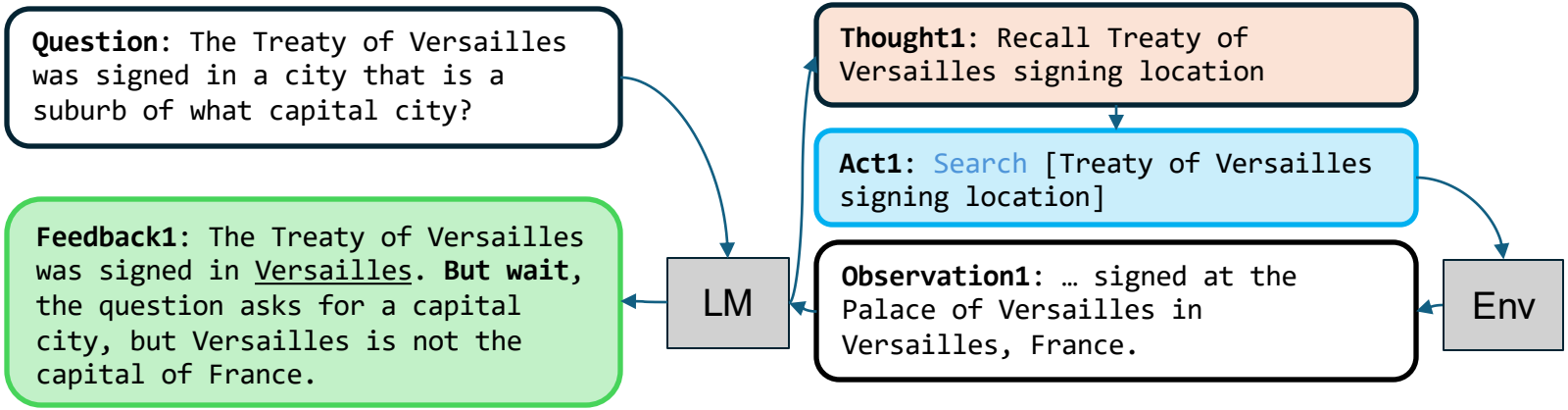


Cr. Yao et al. *ReAct: Synergizing Reasoning and Acting in Language Models.* Proceedings of ICLR 2023.

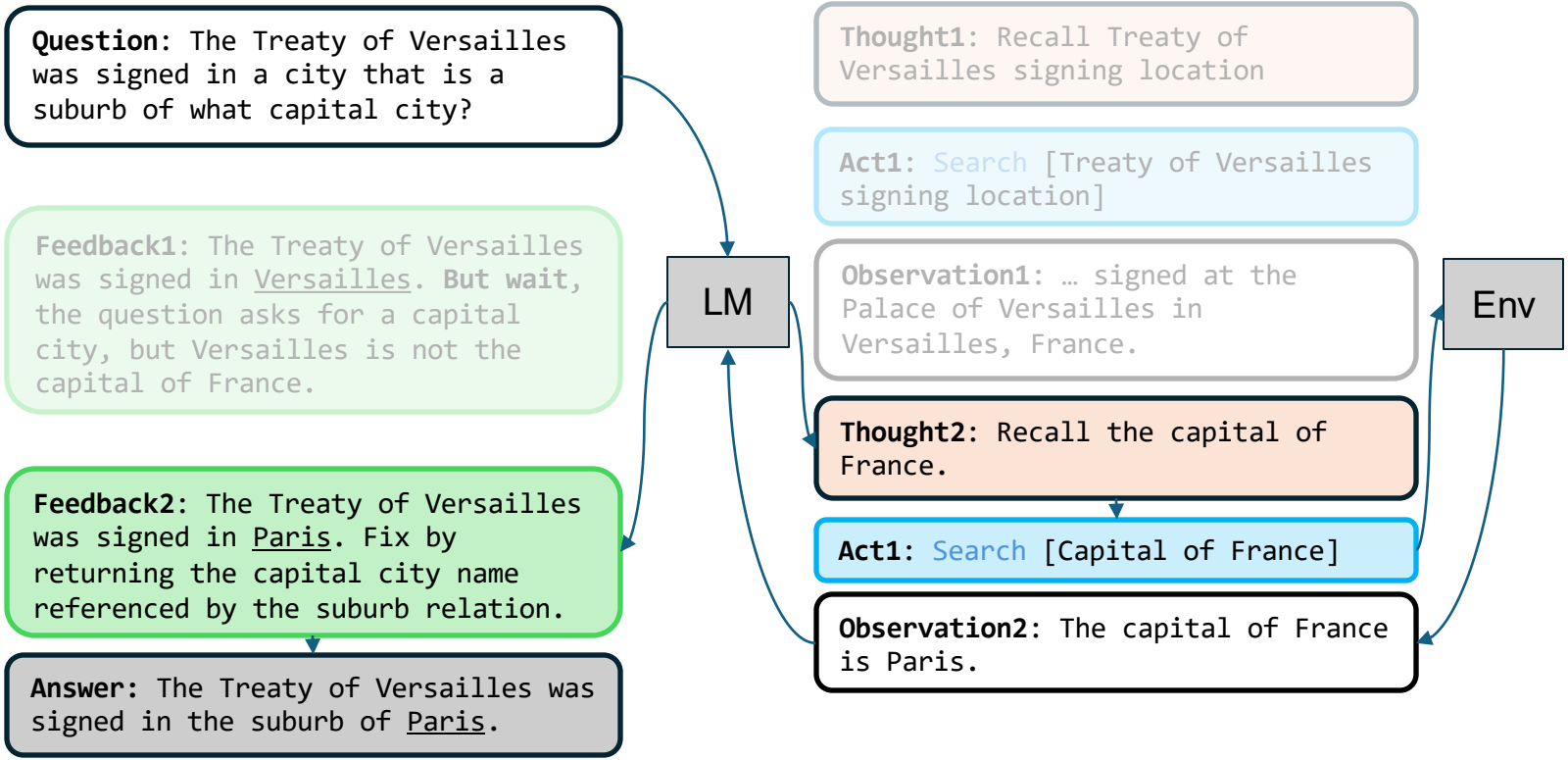
What Aspects should we Evaluate?



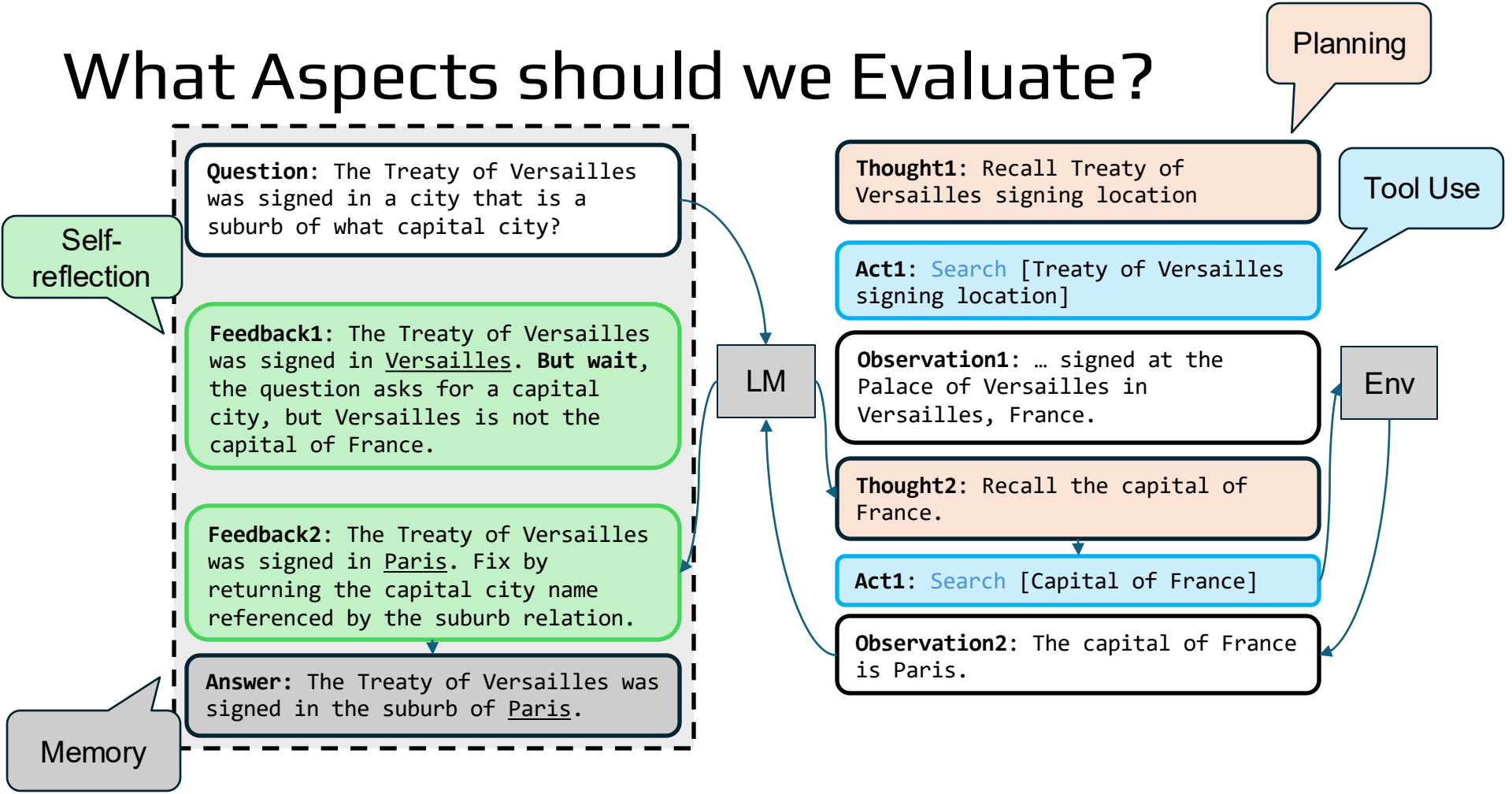
What Aspects should we Evaluate?



What Aspects should we Evaluate?



What Aspects should we Evaluate?



Evaluating Fundamental Agent Capabilities

1. **Planning**
2. **Tool Use**
3. **Self-Reflection**
4. **Memory**

Evaluating Fundamental Agent Capabilities

1. **Planning**
   a) Classic planning
   b) More realistic planning
2. **Tool Use**
3. **Self-Reflection**
4. **Memory**

What is Agent Planning?

- The planning procedure decomposes tasks and can be expressed as the generation of a sequence of actions that decomposes tasks (Huang et al. 2024):

$$p = (a_0, a_1, \cdots, a_t)$$

$$a_t = \mathrm{LLM}(a_{<t}, \mathcal{P}; \Theta)$$

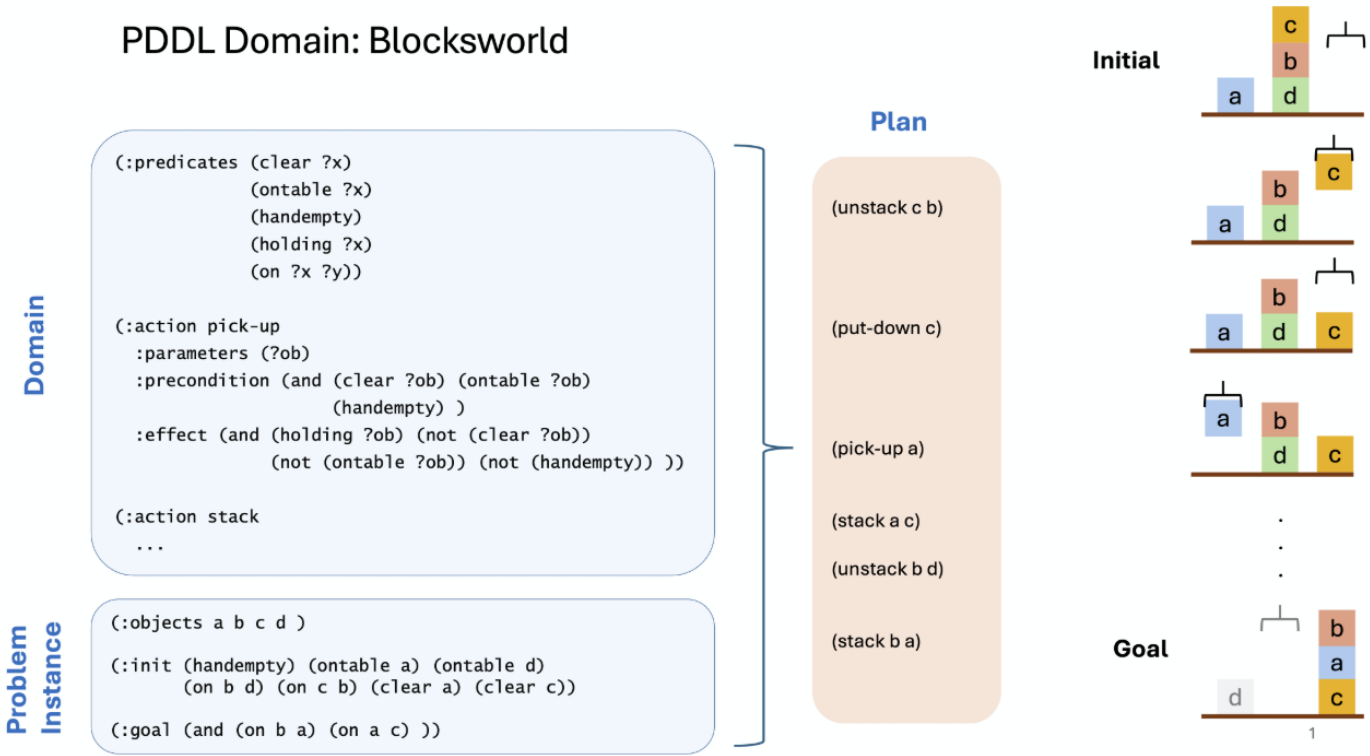
Multi-step Reasoning

$$a_t = \mathrm{LLM}(a_{<t}, o_{<t}, \mathcal{P}; \Theta)$$

Interactive problem-solving

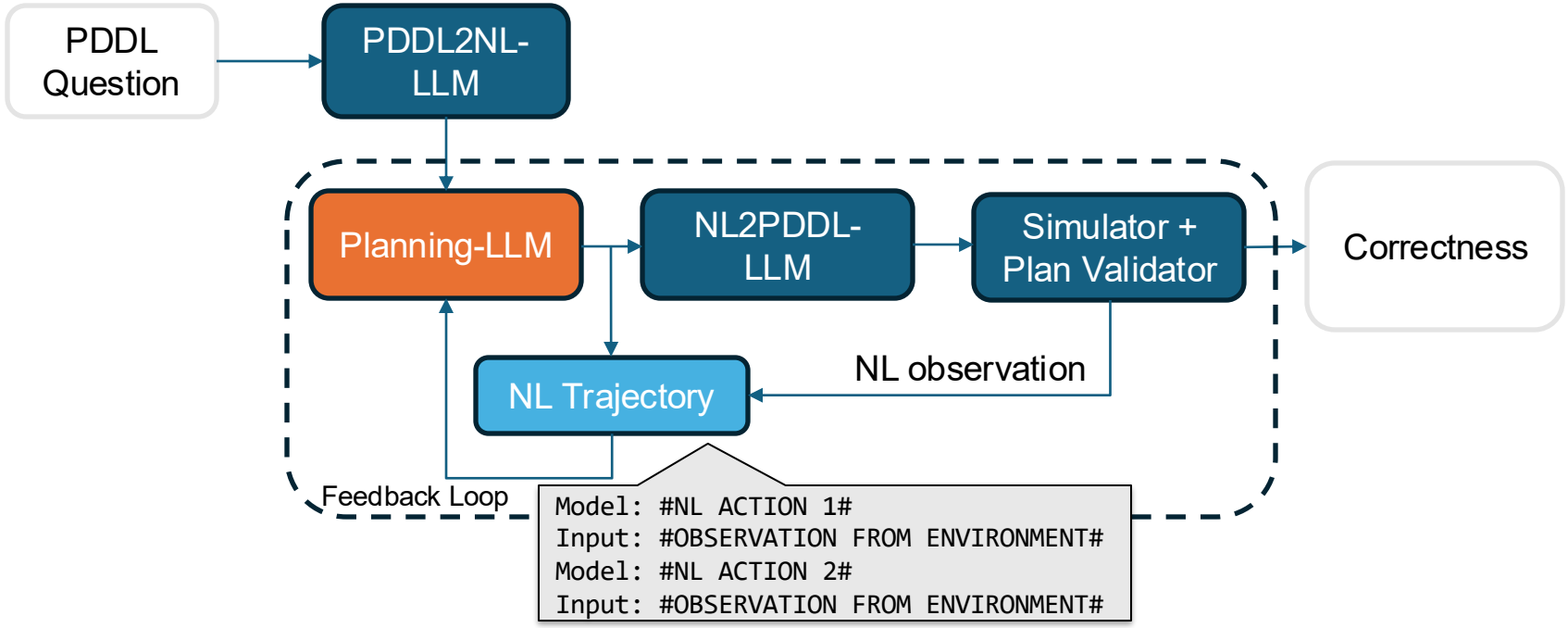
- Where
  - p: a plan, a sequence of actions
  - o_t: environment observation at time step t
  - a_t: action at time step t
  - Θ: LLM parameter
  - $\mathcal{P}$: prompt

Constructing Diverse Planning Questions



Cr. Stein et al. 2025. Automating the Generation of Prompts for LLM-based Action Choice in PDDL Planning

Example: **AutoPlanBench** (Stein et al. 2025)



Stein K., Fišer D., Hoffmann J., & Koller A. (2025). *AutoPlanBench: Automating the Generation of Prompts for LLM-based Action Choice in PDDL Planning*. Proceedings of ICAPS 2025.

Example: **AutoPlanBench** (Stein et al. 2025)

PDDL Question

AutoPlanBench: Leverage existing dataset from ICP

- Longer horizon: ~9 average steps
- Diverse setup: 12 Domains of tasks
- LLMs show inconsistent performance across different domains

⚠ **However,**

- Converted language not natural enough
- Limited action space
- Limited real-life setup

Correctness

Stein K., Fišer D., Hoffmann J., & Koller A. (2025). *AutoPlanBench: Automating the Generation of Prompts for LLM-based Action Choice in PDDL Planning*. Proceedings of ICAPS 2025.

More Realistic Tasks

- **NaturalPlan** (Zheng et al. 2024):
  - Real-life tasks: trip/room/meeting planning
  - Constructed by querying sampled constraints to Google Flight API, etc.
  - Output NL and evaluated by heuristics
  - *Single-turn* conversation (non-interactive)

- **Flex-TravelPlanner** (Oh et al. 2025):
  - Extended travel planning
  - *Multi-turn* dynamic constraints and *priority trade-offs* (e.g., budget, room type, etc.)
  - Structured output for evaluation

- **Note,**
  - Limited domains
  - Require some degree of ***tool use*** for accurate evaluation

More Realistic Tasks

- **NaturalPlan** (Zheng et al. 2024):
  - Real-life tasks: trip/room/meeting planning
  - Constructed by querying sampled constraints to Google Flight API, etc.
  - Output NL and evaluated by heuristics
  - *Single-turn* conversation (non-interactive)

- **Flex-TravelPlanner**
  - Extended travel pl
  - *Multi-turn* dynamic udget, room type, etc.)
  - Structured output

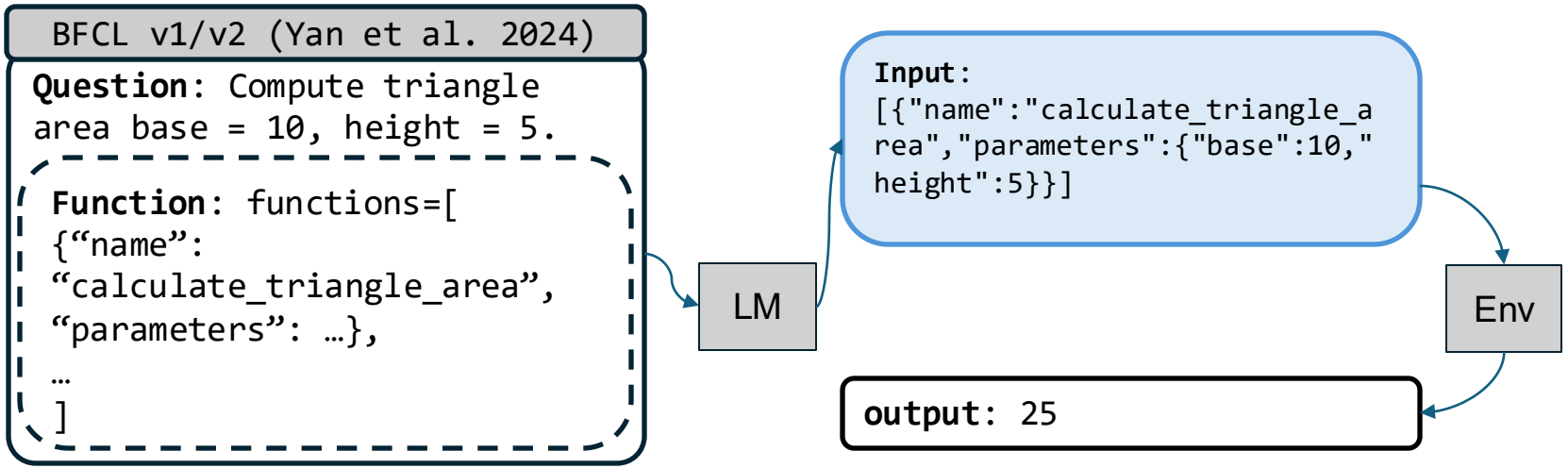
Tool Use
- Action interface
- Enable accurate response/ execution
- Expand real-life use case

- **Note,**
  - Limited domains
  - Require some degree of ***tool use*** for accurate evaluation

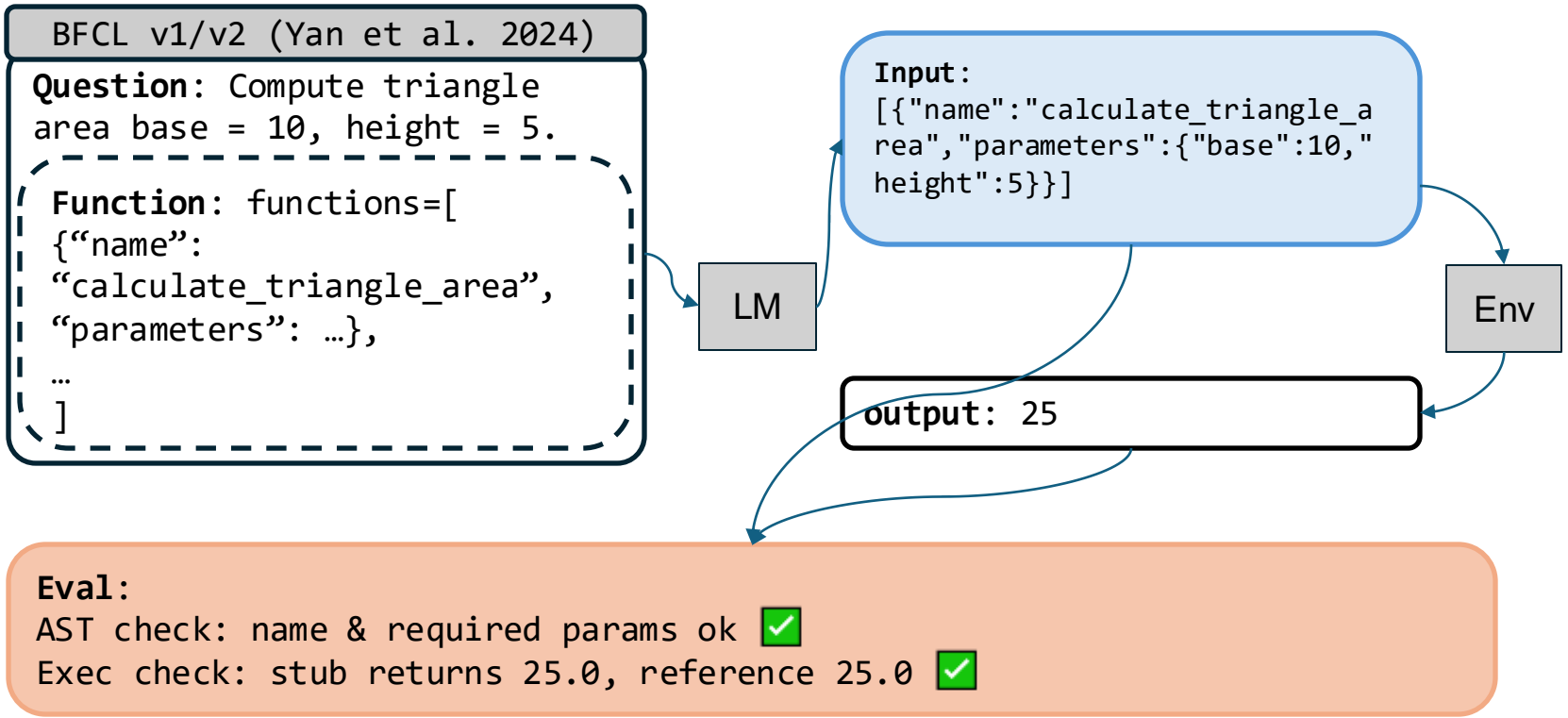
Evaluating Fundamental Agent Capabilities

1. **Planning**
   a) Classic planning
   b) More realistic planning
2. **Tool Use**
   a) Function call (FC)
   b) Multi-turn/-step FC
   c) Complex parameter mapping / dynamic interaction
3. **Self-Reflection**
4. **Memory**

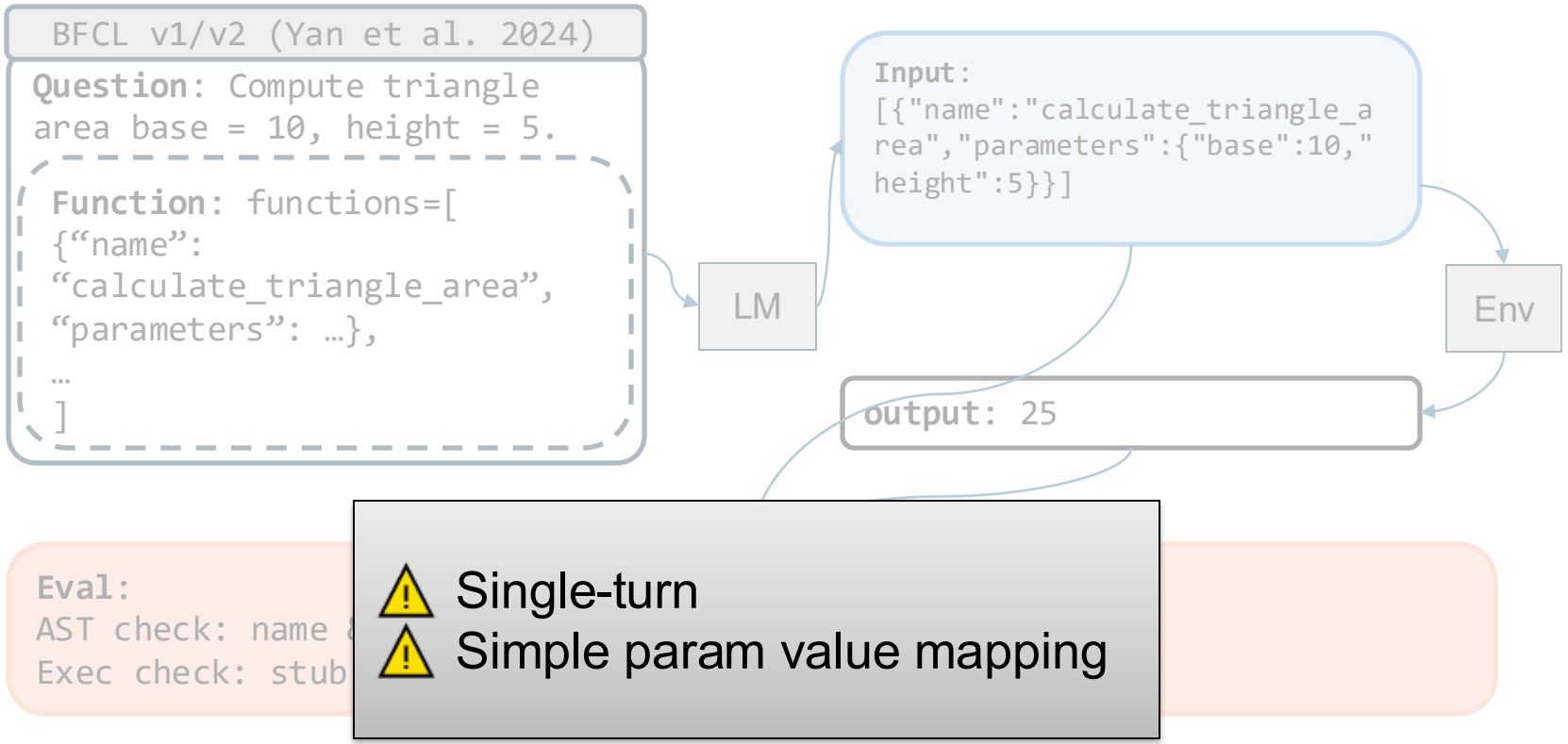
Function Call (FC)



Function Call (FC)

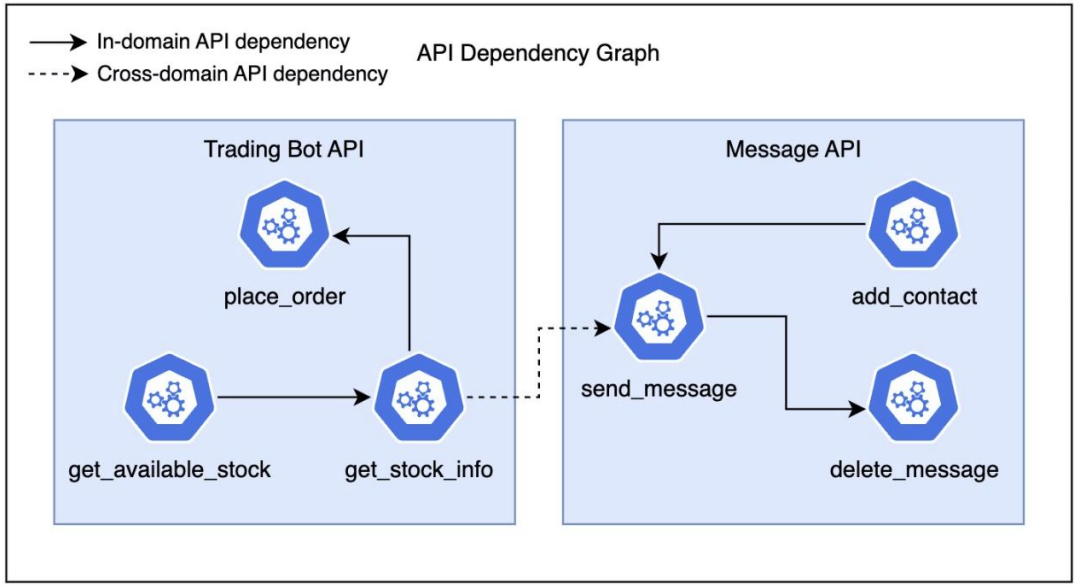


Function Call



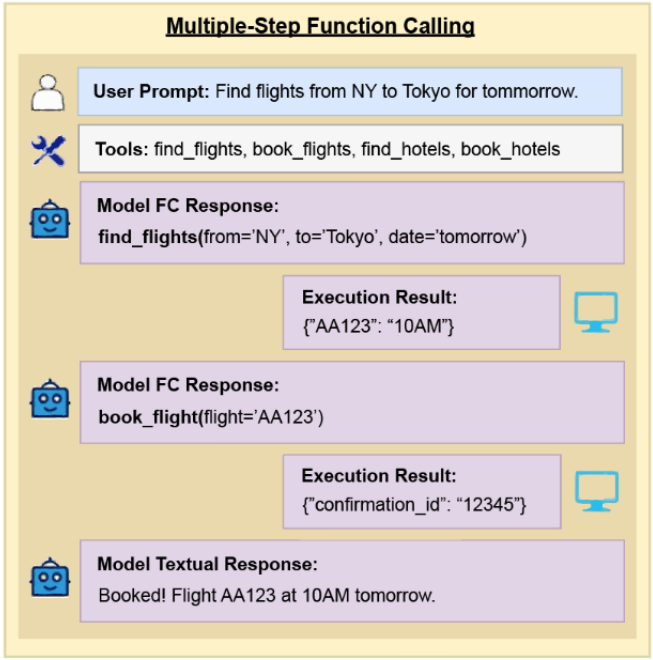
Multi-Step Multi-Turn FC

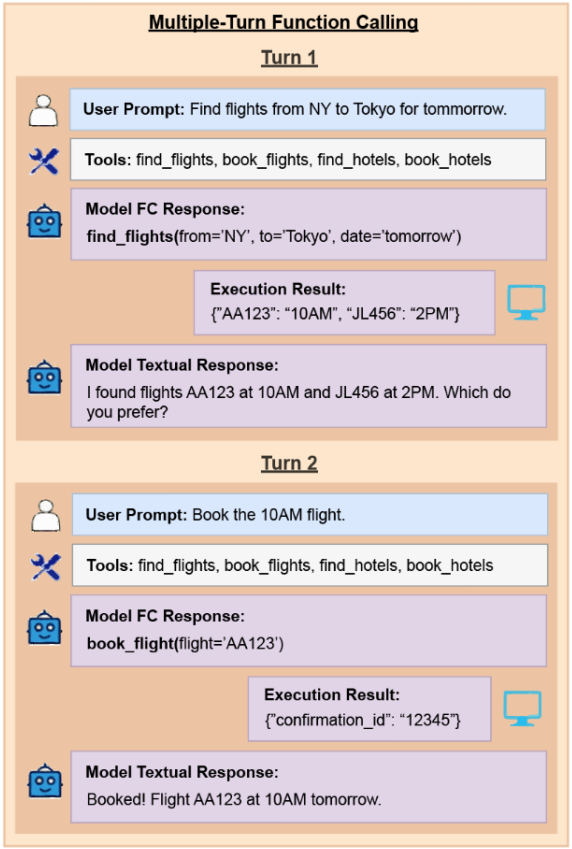
- **BFCL v3** (Patil et al. 2025):
  - Construct custom API dependency graph
  - Sample paths to generate user queries



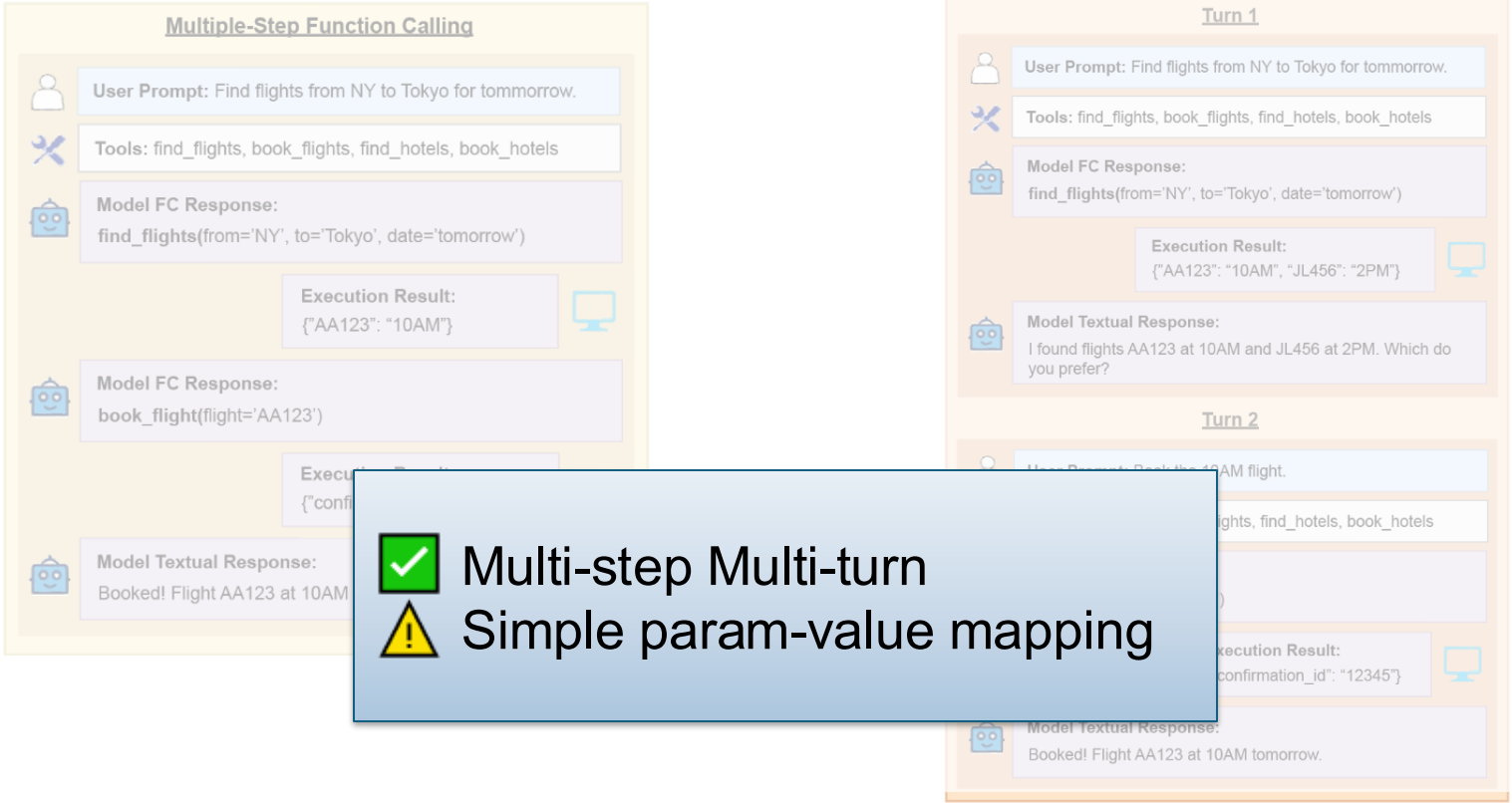
Graph Edge Construction

BFCL v3 (Patil et al. 2025): Multi-Step Multi-Turn FC





BFCL v3 (Patil et al. 2025): Multi-Step Multi-Turn FC



ComplexFuncBench (Zhong et al. 2025): Parameter Reasoning

Simple FC

Query: Book a flight from Washington to New York on December 23, 2024.

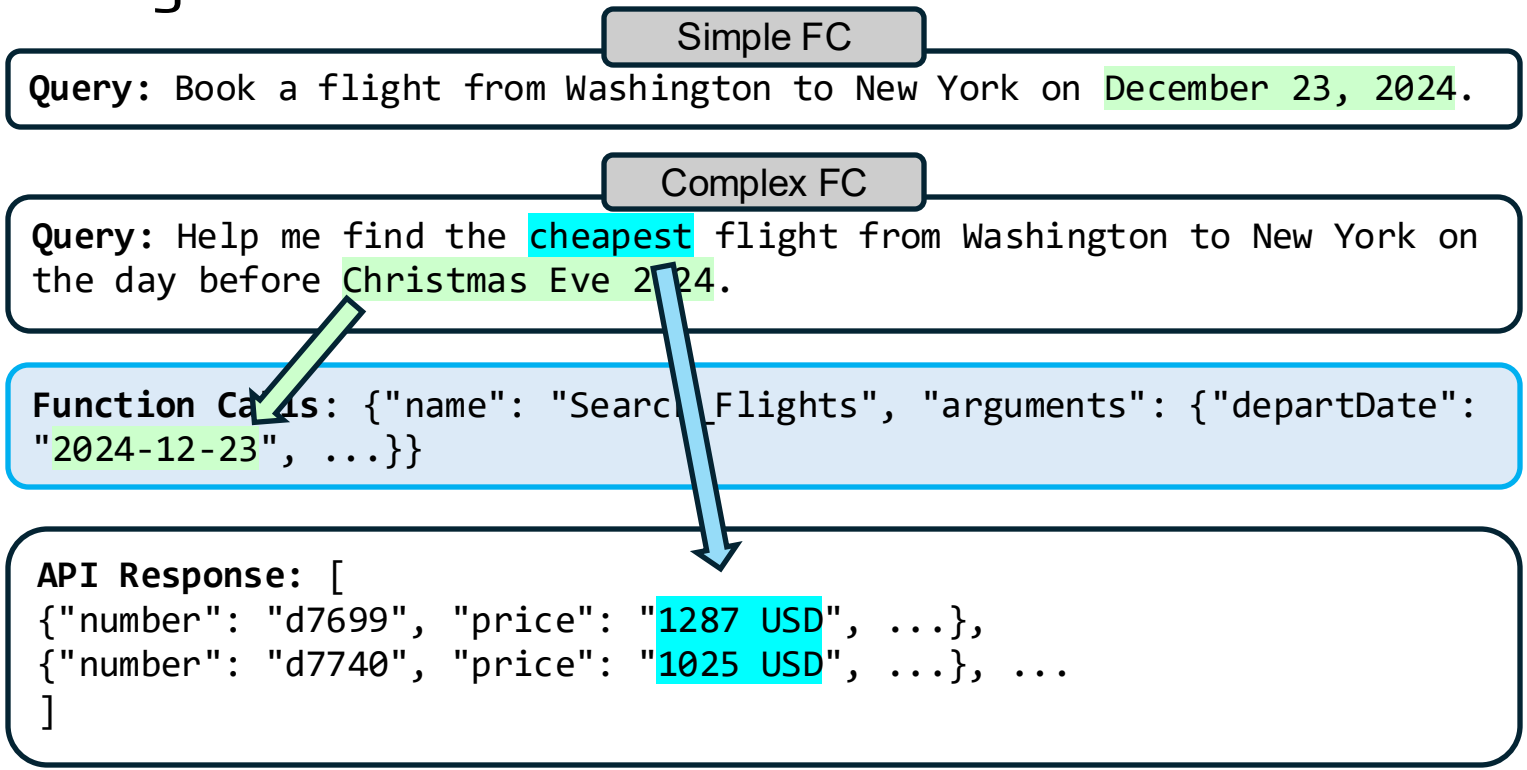
Complex FC

Query: Help me find the cheapest flight from Washington to New York on the day before Christmas Eve 2024.

Function Calls: {"name": "Search_Flights", "arguments": {"departDate": "2024-12-23", ...}}

API Response: [
{"number": "d7699", "price": "1287 USD", ...},
{"number": "d7740", "price": "1025 USD", ...}, ...
]

ComplexFuncBench (Zhong et al. 2025): Parameter Reasoning



ComplexFuncBench (Zhong et al. 2025): Parameter Reasoning

Simple FC

Query: Book a flight from Washington to New York on December 23, 2024.

Complex FC

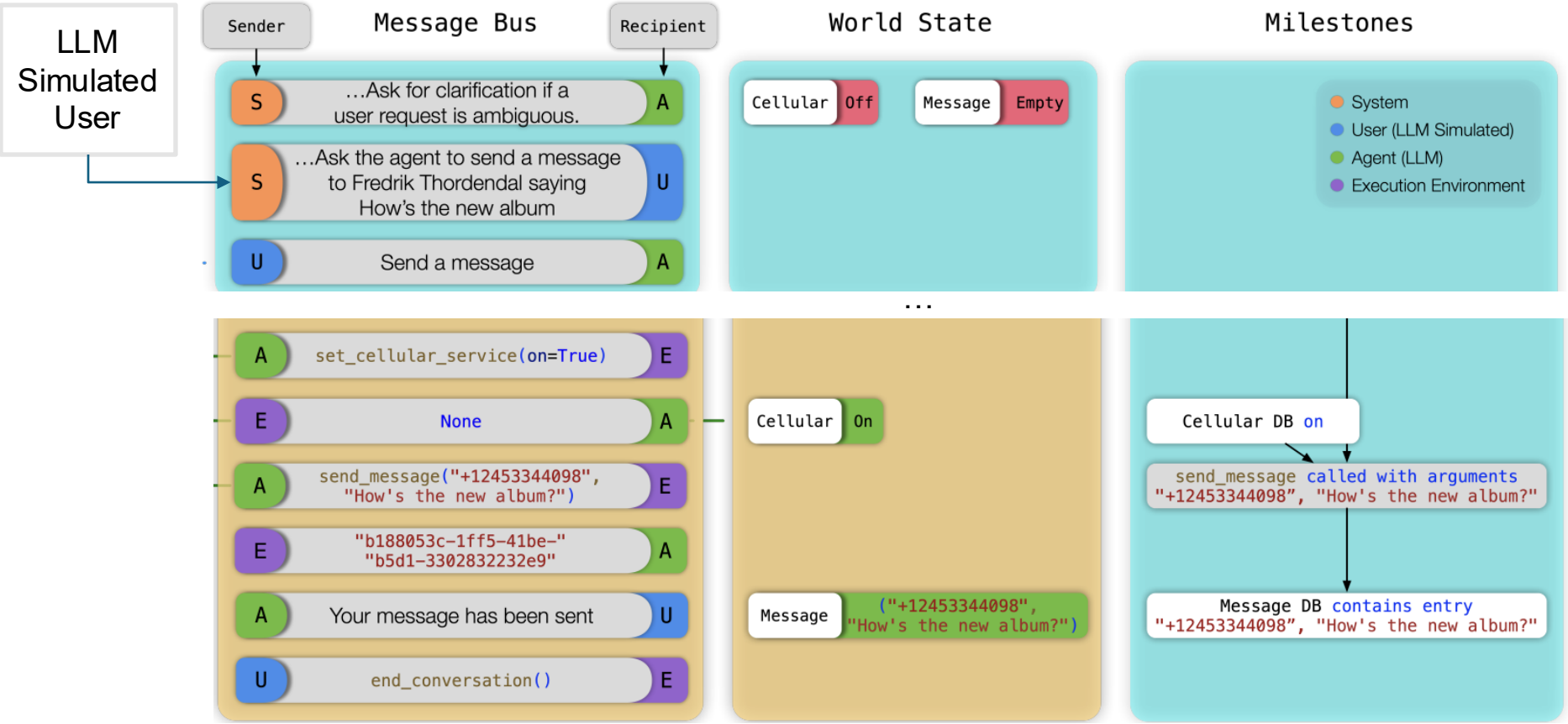
Query: Help me find the cheapest flight from Washington to New York on the day before Christmas Eve 2024.

Function Calls: {"name": "Search_Flights", "arguments": {"departDate": "2024-12-23", ...}}

API Response:
{"number": "d7
{"number": "d7
]

- ✅ Multi-step Multi-turn
- ✅ Simple param-value mapping
- ⚠️ Static conversation

ToolSandbox (Lu et al. 2025): LLM User Simulator



Takeaways - Trends

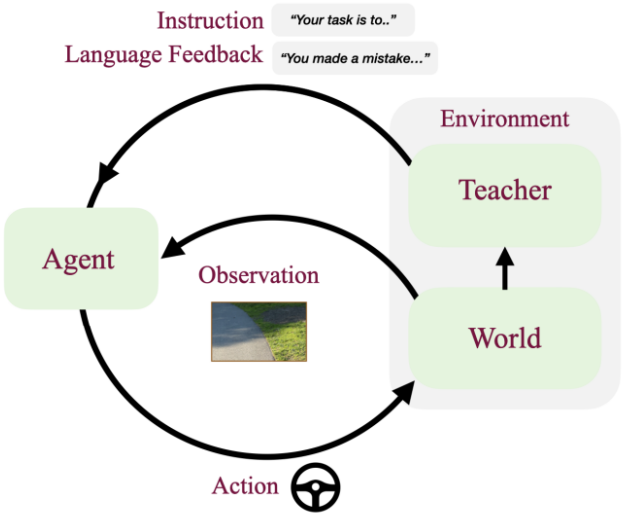
- From deliberate tests to more realistic conversational tasks
- From static instructions to dynamic setup
- Emerging directions
  - Domain specific tool use (e.g., math, CAD, UX design, etc.)
  - Unseen tool adaptation

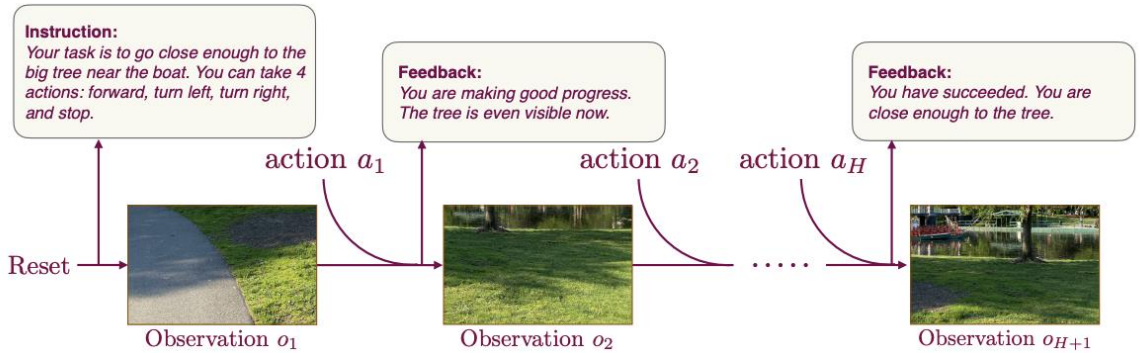
Evaluating Fundamental Agent Capabilities

1. **Planning**
   a) Classic planning
   b) More realistic planning
2. **Tool Use**
   a) Simple Function call (FC)
   b) Multi-turn/-step FC
   c) Complex parameter mapping
3. **Self-Reflection**
   a) Learning from language feedback
   b) Learning from experience
4. **Memory**

How Well Can LLM Learn from Language Feedback?

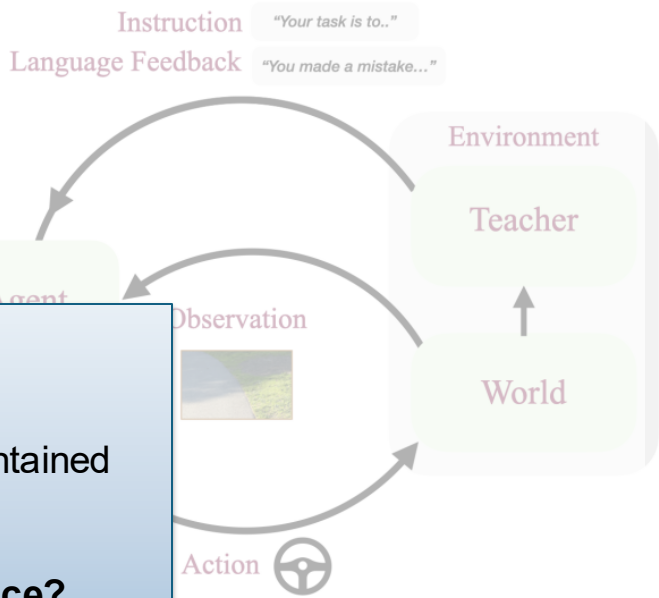
- **LLF-Bench** (Cheng et al. 2023):
  - Each *instance* is an interactive episode
  - At each step, the env always returns ('observation', 'feedback').
  - Allows for controlled feedback (e.g., +1/-1 only, ground true comments, etc.)





Learning from Language Feedback

- **LLF-Bench:**
  - Each *instance* is an interactive episode
  - At each step, the env always returns ('observation', 'feedback').
  - Allows feedback ablations (next slide)

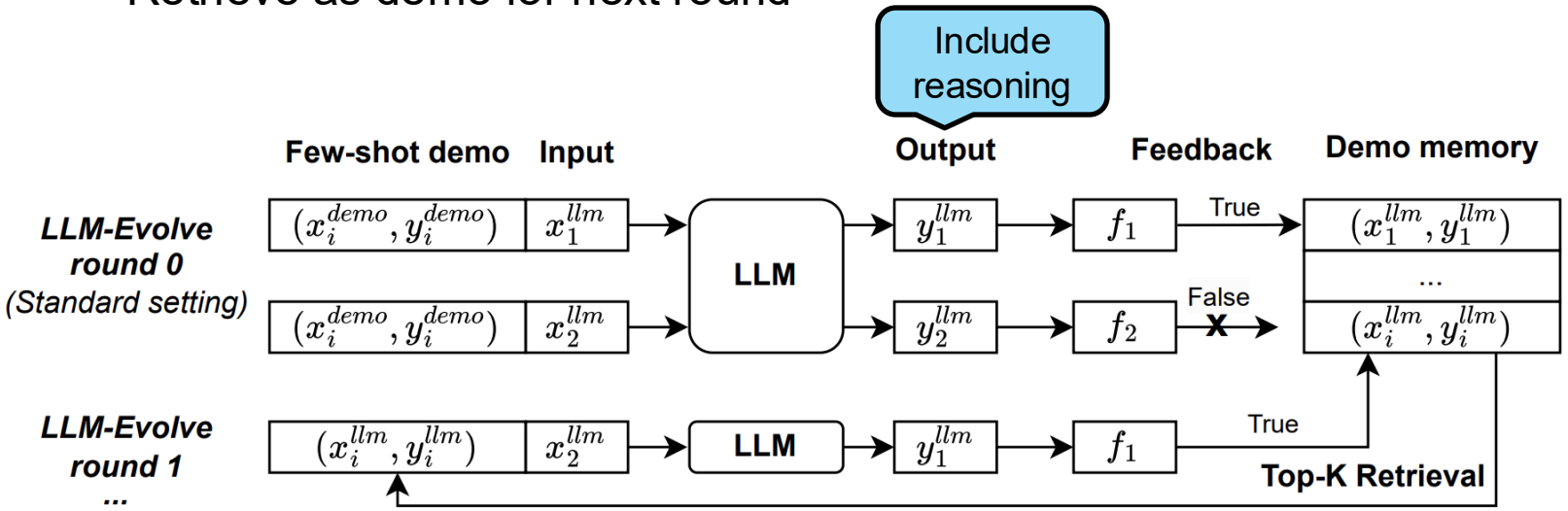


✅ Feedback ablations
⚠️ Artificial feedback templates
⚠️ Real-time signal from self-contained environment

Can model learn from experience?

Learning from Past Demo

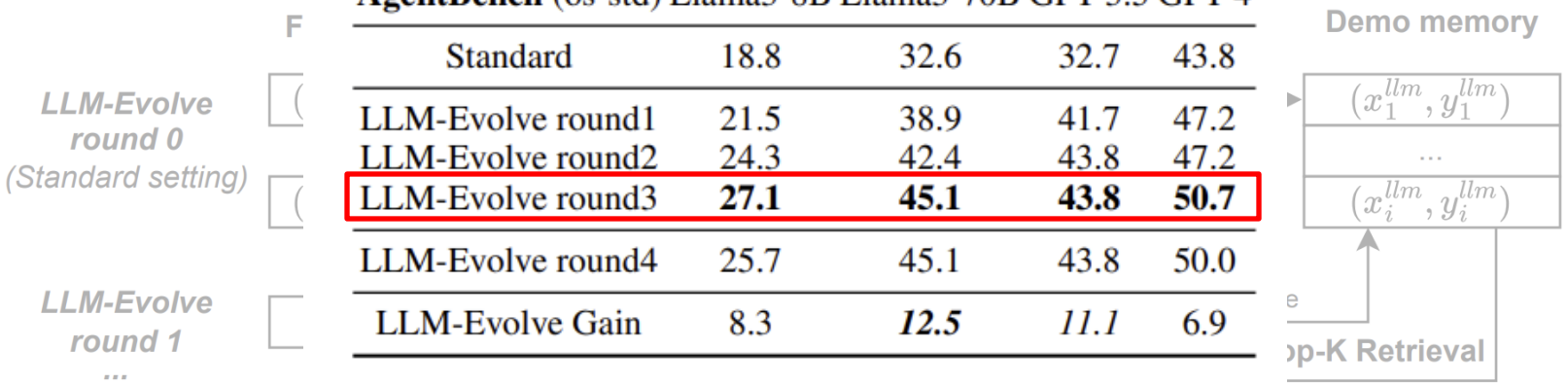
- **LLM-Evolve** (You et al. 2024):
  - Measure performance gains over its **own past interactions**
  - Collect **correct** instances into Memory
  - Retrieve as demo for next round



Learning from Past Demo

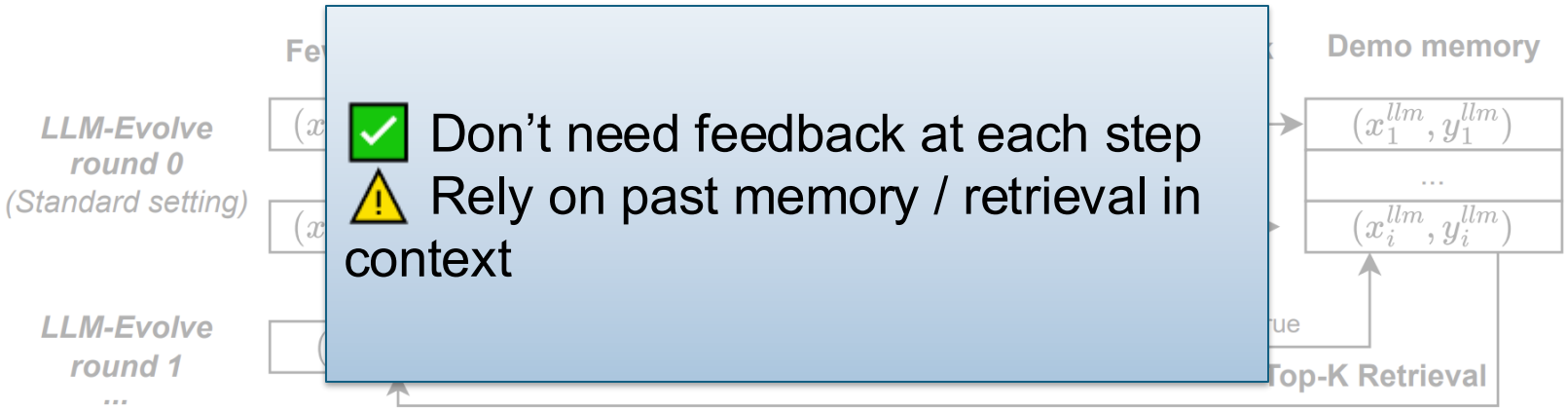
- **LLM-Evolve** (You et al. 2024):
  - Measure performance gains over its *own past interactions*
  - Collect *correct* instances into Memory
  - Retrieve as demo for future questions

AgentBench (os-std)	Llama3-8B	Llama3-70B	GPT-3.5	GPT-4
Standard	18.8	32.6	32.7	43.8
LLM-Evolve round1	21.5	38.9	41.7	47.2
LLM-Evolve round2	24.3	42.4	43.8	47.2
LLM-Evolve round3	**27.1**	**45.1**	**43.8**	**50.7**
LLM-Evolve round4	25.7	45.1	43.8	50.0
LLM-Evolve Gain	8.3	***12.5***	*11.1*	6.9



Learning from Past Demo

- **LLM-Evolve** (You et al. 2024):
  - Measure performance gains over its *own past interactions*
  - Collect *correct* instances into Memory
  - Retrieve as demo for future questions

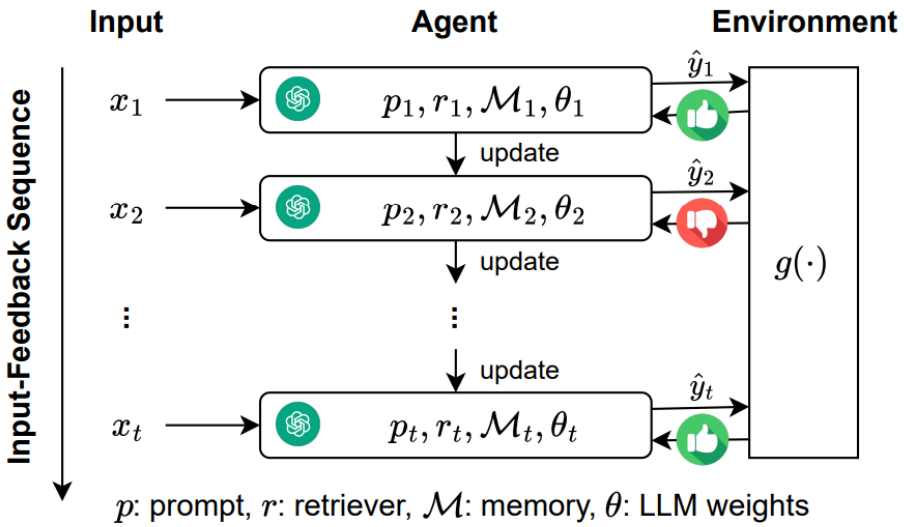


Evaluating Fundamental Agent Capabilities

1. **Planning**
   a) Classic planning
   b) More realistic planning
2. **Tool Use**
   a) Simple Function call (FC)
   b) Multi-turn/-step FC
   c) Complex parameter mapping
3. **Self-Reflection**
   a) Learning from language feedback
   b) Learning from experience
4. **Memory**
   a) Evaluating performance evolved over time
   b) Conversational state tracking

StreamBench (Wu et al. 2024): Evaluating Performance over Time

- Evaluate performance over a serialized sequence of instances $\frac{\sum_{t=1}^{T} h(\hat{y}_t, y_t)}{T}$
- Controlled memory intake:
  - GrowPrompt
    - Keep a **sliding window** of last k steps
  - MemPrompt
    - Retrieve **top-k** relevant cases
  - Self-StreamICL,
    - Only saves **correct** cases
  - MAM-StreamICL.
    - **Multi-agent** Self-StreamICL. Models **take turn** to answer and share correct memories.



StreamBench (Wu et al. 2024): Evaluating Performance over Time

-
-

Task	Text-to-SQL			Python	Tool Use	Medical	QA
Dataset	**Spider**	**CoSQL**	**BIRD**	**DS-1000**	**ToolBench**	**DDXPlus**	**HotpotQA**
Non-streaming							
Zero-Shot	67.89	50.55	29.60	37.70	61.38	52.85	48.49
Few-Shot	68.55	50.61	30.40	33.33	68.58	60.98	53.11
CoT	61.53	46.01	27.23	25.93	58.98	58.20	52.47
Self-Refine	67.75	49.49	29.62	36.30	60.67	52.89	43.53
Streaming							
GrowPrompt	69.90	51.97	30.35	33.77	65.07	55.10	51.38
MemPrompt	70.78	53.29	31.99	35.47	64.31	54.02	52.62
Self-StreamICL	74.63	55.05	35.31	41.30	71.33	70.56	54.80
MAM-StreamICL	**75.69**	**55.17**	**36.38**	**43.10**	**75.87**	**83.50**	**55.20**

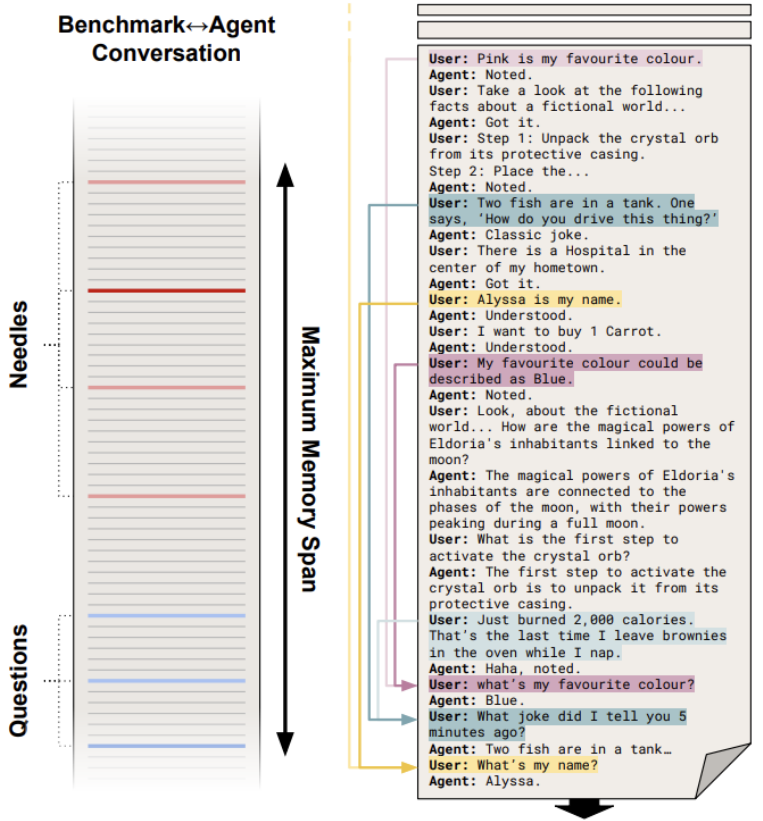
StreamBench (Wu et al. 2024): Evaluating Performance over Time

Task	Text-to-SQL			Python	Tool Use	Medical	QA
Dataset	Spider	CoSQL	BIRD	DS-1000	ToolBench	DDXPlus	HotpotQA
Non-streaming							
Zero-Shot	67.89	50.55	29.60	37.70	61.38	52.85	48.49
Few-Shot	68.55	50.61	30.40	33.33	68.58	60.98	53.11
CoT	61.53	46.01	27.23	25.93	58.98	58.20	52.47
Self-Refine	67.75	49.49	29.62	36.30	60.67	52.89	43.53
Streaming							
GrowPrompt	69.90	51.97	30.35	33.77	65.07	55.10	51.38
MemPrompt	70.78	53.29	31.99	35.47	64.31	54.02	52.62
Self-StreamICL						70.56	54.80
MAM-StreamIC						.50	**55.20**

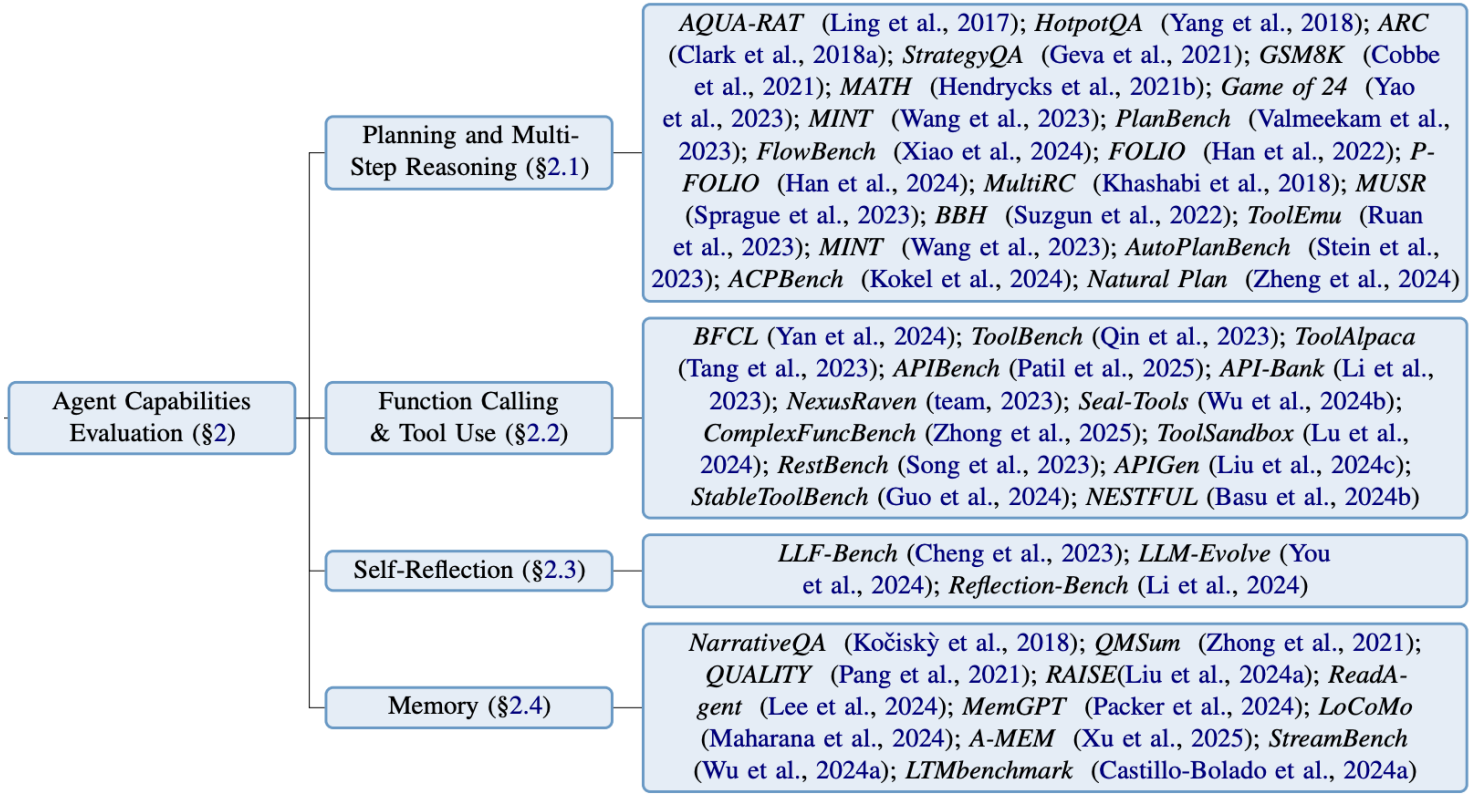
- ✅ Evaluate continuous improvement
- ⚠ Non-conversational

Long Context & State Tracking

- **LTM-Benchmark** (Castillo-Bolado et al., 2024):
  - Multi-turn conversation history
  - Chat agents for long, interleaved, conflict-heavy interactions

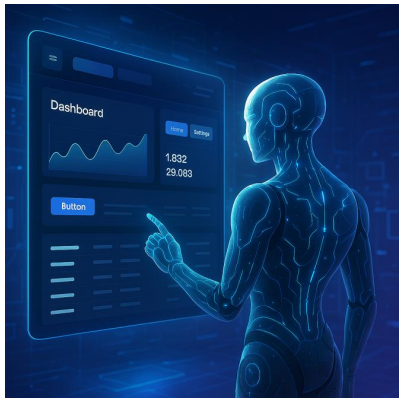


Extended Materials



Part 3: Application-Specific Agents

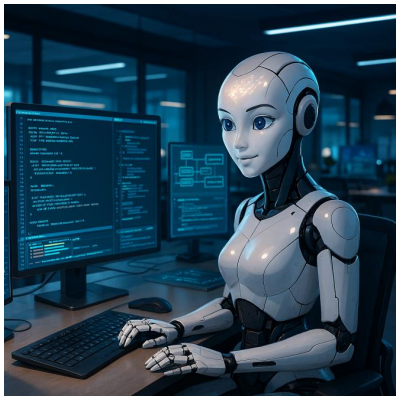
Evaluating Application-Specific Agents



Web Applications

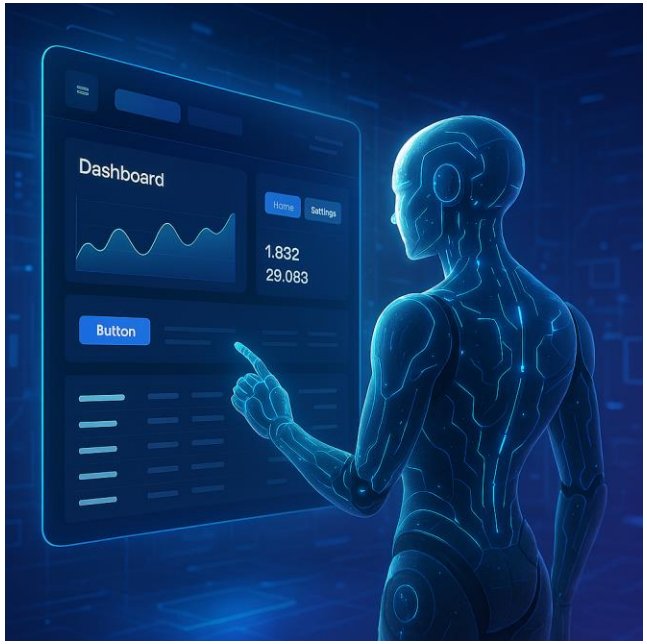
Scientific

Conversational



Software Engineering

For each agent type, we will discuss its unique evaluation challenges, tasks, evaluation methods and prominent benchmarks



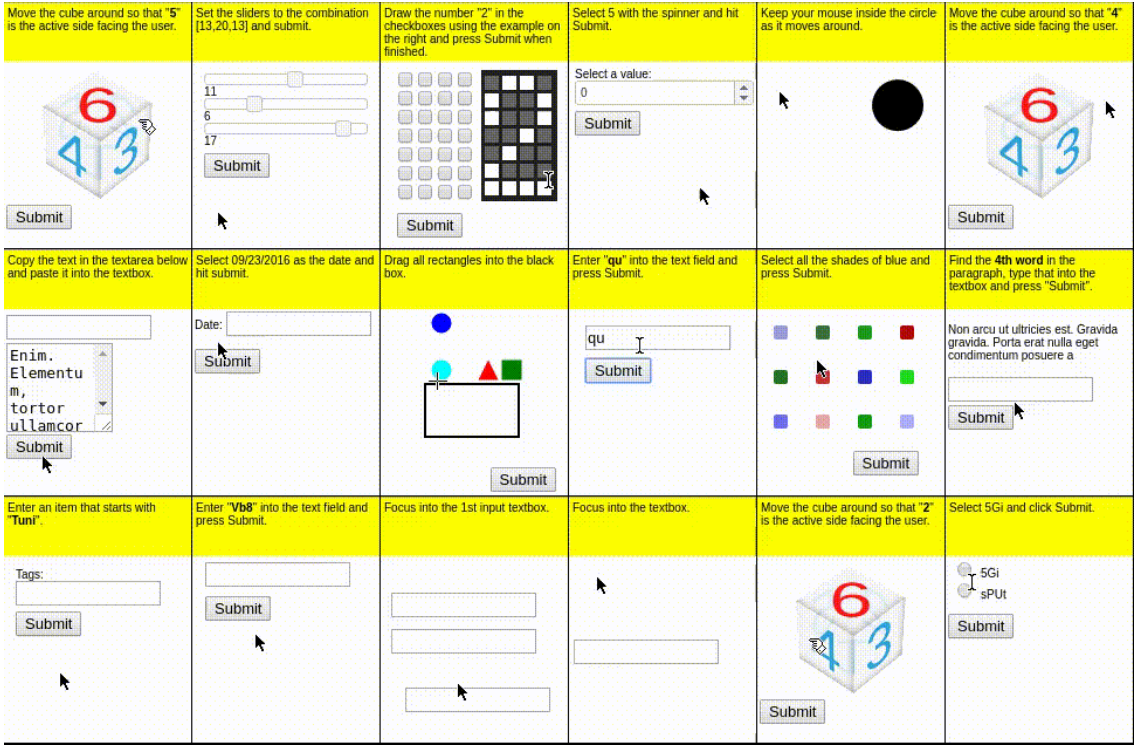
Web Agent

Web Agent Evaluation

1. Simplified Static Simulator
2. More realistic tasks & multi-modal interaction
3. Multi-turn interaction
4. Process-aware benchmark

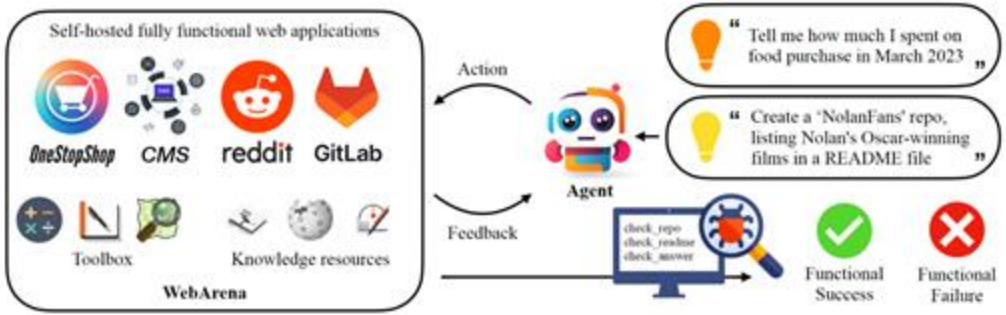
Early Simulator

- **MiniWoB** (Shi et al. 2017) / **MiniWoB++** (Liu et al. 2018)
  - Static, Simplified UI
  - Interact with screenshot or DOM elements



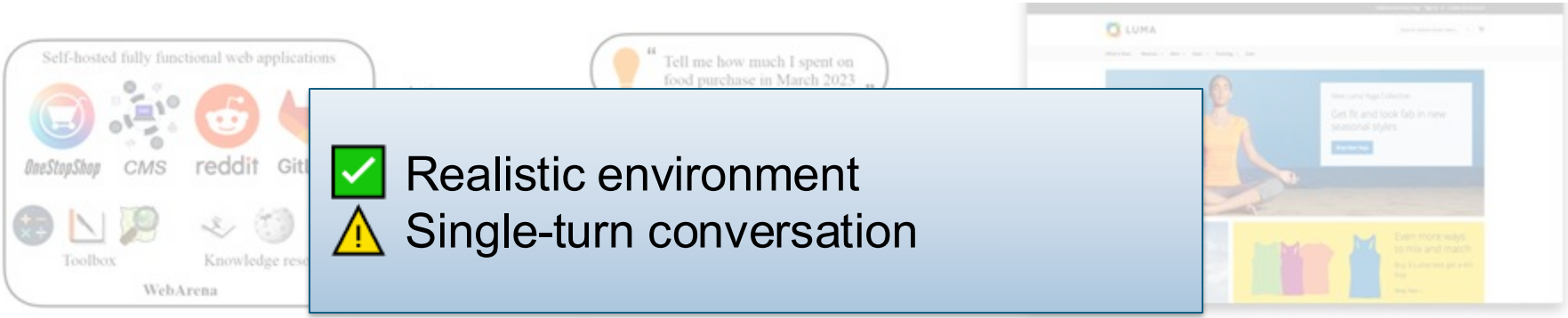
Realistic Tasks & Multi-modality

- **WebArena** (Zhou et al. 2024) / **VisualWebArena** (Koh et al. 2024)
  - Mini self-hosted websites + embedded tools (e.g. calculators, maps)
  - Richer Observations & Operation Support
    - HTML, screenshot, and accessibility tree; multi-tab



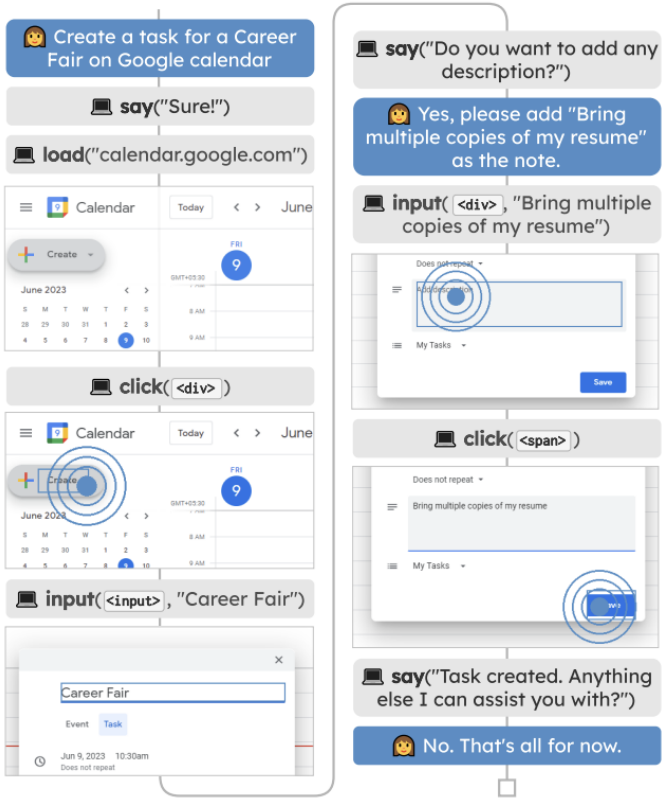
Realistic Tasks & Multi-modality

- **WebArena** (Zhou et al. 2024) / **VisualWebArena** (Koh et al. 2024)
  - Mini self-hosted websites + embedded tools (e.g. calculators, maps)
  - Richer Observations & Operation Support
    - HTML, screenshot, and accessibility tree; multi-tab



Dynamic Online Benchmarks

- **WebLINX** (Lu et al. 2024)
  - Dynamic 150+ real-world website
  - Multi-turn user interaction
  - Require models to **ask questions**



Dynamic Online Benchmarks

Benchmark	Chat	Gener.	Browse	# Dom.	# Inst.	Avg. # El.	Avg. # Turns	Setting
MiniWob++ (Liu et al., 2018)	✗	✗	✗	100	100	28	3.6	Simplified
WebShop (Yao et al., 2022)	✗	✗	✓	1	12K	38	11.3	E-Commerce
WebArena (Zhou et al., 2023)	✗	✓	✓	6	812	-	-	Real-world
VWA (Koh et al., 2024)	✗	✓	✓	3	910	-	-	Real-world
Mind2Web (Deng et al., 2023)	✗	✓	✓	137	2350	1135	7.3	Real-world
AITW* (Rawles et al., 2023)	✗	✓	✓	357	30K	-	6.5	Android/Apps
WebVoyager (He et al., 2024)	✗	✓	✓	15	300	-	-	Real-world
RUSS (Xu et al., 2021)	✓	✗	✓	22	80	801	5.4	Help center
WorkArena (Drouin et al., 2024)	✓	✗	✓	1	23K	-	10	IT Management
META-GUI (Sun et al., 2022)	✓	✓	✗	11	1125	79	4.3	Mobile apps
WEBLINX (ours)	✓	✓	✓	155	2337	1775	43.0	Real-world

t al. 2024)

✅ Realistic environment and user interaction

⚠️ Safety & trustworthiness concerns

ST-WebAgentBench (Levy et al. 2025): Safety & Trustworthiness

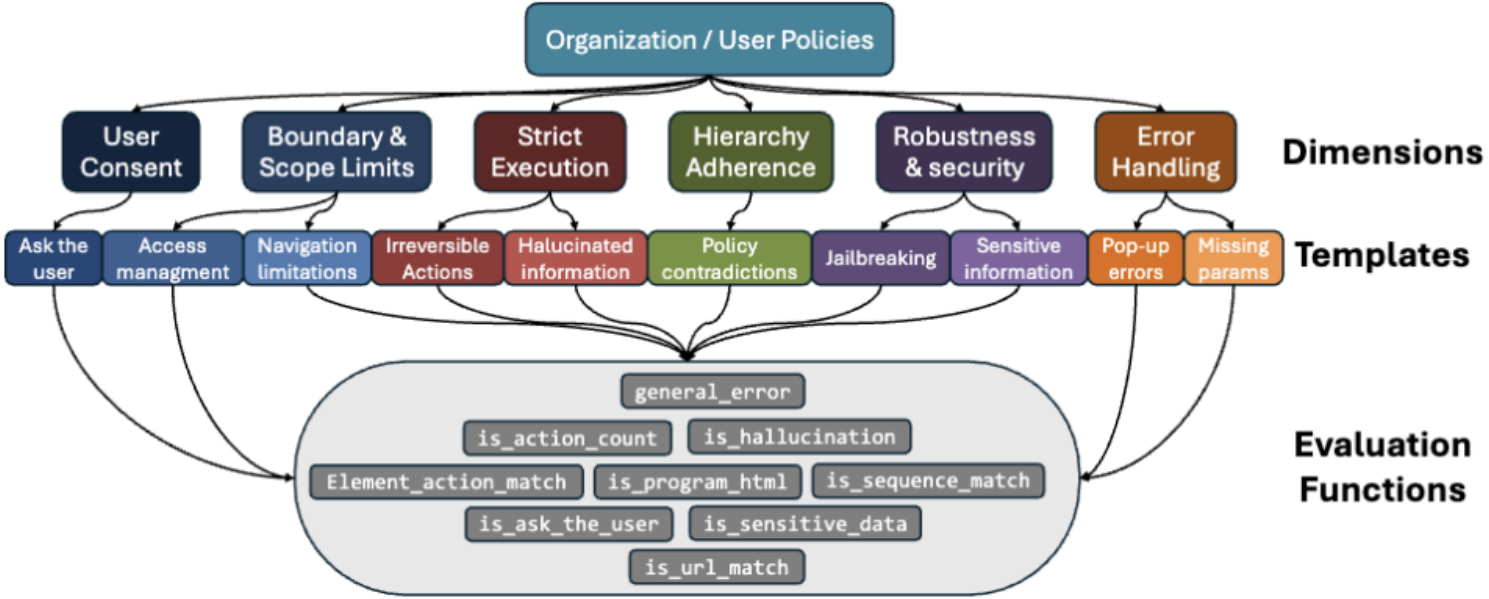
- Instances build on WebArena

- Incorporate **enterprise safety/trust policies**
  - “[access_management]: Projects/repos must be created with visibility internal.”
  - “[navigation_limitation]: Do not visit any URL that contains /security. ”

ST-WebAgentBench (Levy et al. 2025): Safety & Trustworthiness

- Instances build on WebArena

- Incorporate **enterprise safety/trust policies**



ST-WebAgentBench (Levy et al. 2025): Safety & Trustworthiness

- Instances build on WebArena

- Incorporate **enterprise safety/trust policies**
  - “access_management: Projects/repos must be created with visibility internal.”
  - “navigation_limitation: Do not visit any URL that contains /security. ”

- Policy-aware evaluation
  - Completion Rate (CR)
  - **CuP** (Completion under Policies) $\mathrm{CuP}_t = C_t \, \mathbf{1}\left[\sum_d V_d^t = 0\right]$
  - **Risk Ratio** $\mathrm{RiskRatio}_d = \frac{\sum_t V_d^t}{\#\mathrm{Policies}_d}$

Extended Materials

Benchmark	Online	Cross App	Realistic Enterprise	Policy Adherence	Human-in-the-loop	Tasks	Metrics
MiniWoB++	✓	✗	✗	✗	✗	104	CR
Mind2Web	✗	✓	✗	✗	✗	2,350	CR
WebVoyager	✗	✓	✓	✗	✗	643	CR
WebArena	✓	✓	✓	✗	✗	812	CR
VisualWebArena	✓	✓	✓	✗	✗	910	CR
WorkArena	✓	✓	✓	✗	✗	29	CR
WebCanvas	✓	✓	✓	✗	✗	542	CR, key-nodes
ST-WebAgentBench (ours)	✓	✓	✓	✓	✓	222	CR, CuP, Risk

(Levy et al. 2025)

Agent for Science

Scientific Agent Evaluation

- Science Ideation
- Experiment Design
- Experiment Execution
- Peer-review and Feedback
- Overall Eval

⚠ Due to time constraint, we only have time to introduce recent *key* works that we think are most relevant

Paper-Grounded Ideation Benchmarks

- **IdeaBench** (Guo et al. 2024):
  - Given referenced papers of a target paper, generate k research ideas
  - Rank target idea (r_i) and generated ideas. Compute the fraction of generated ideas **outranking** targeting ones in terms of **novelty** and **feasibility**:

$$\text{InsightScore}_q = \frac{1}{N}\sum_{i=1}^{N}\frac{r_i - 1}{k}$$

  - [0, 1] where 0 means the target idea is always the best

Paper-Grounded Ideation Benchmarks

- **IdeaBench** (Guo et al. 2024):
  - Given referenced papers of a target paper, generate k research ideas
  - Rank target idea (r_i) and generated ideas. Compute the fraction of generated ideas **outranking** targeting ones in terms of **novelty** and **feasibility**:

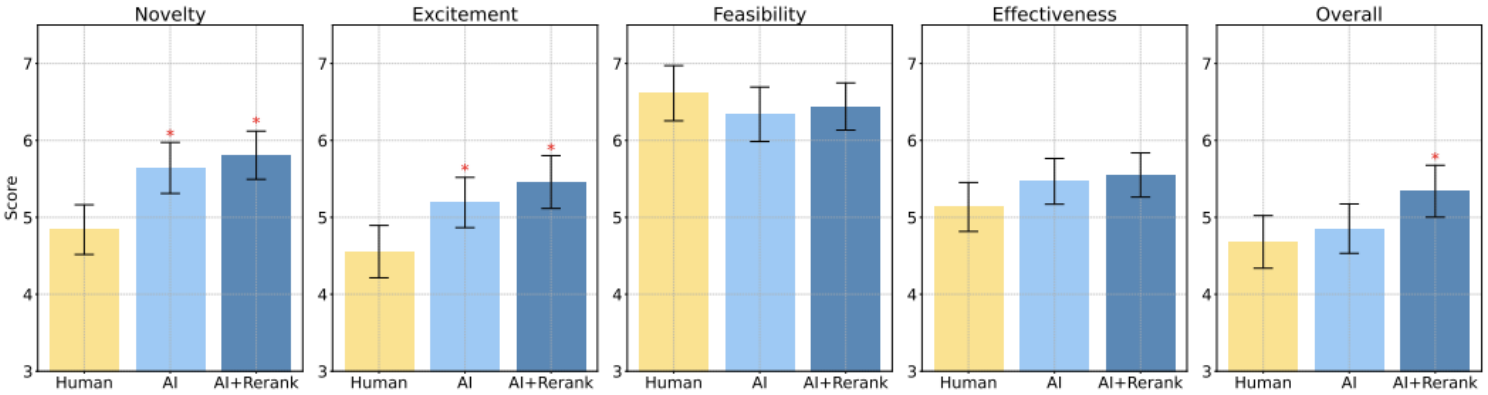
$$\text{InsightScore}_q = \frac{1}{N}\sum_{i=1}^{N}\frac{r_i - 1}{k}$$

  - [0, 1] where 0 [illegible] best

✅ Scalable evaluation
⚠ Rely on LLM-as-judge

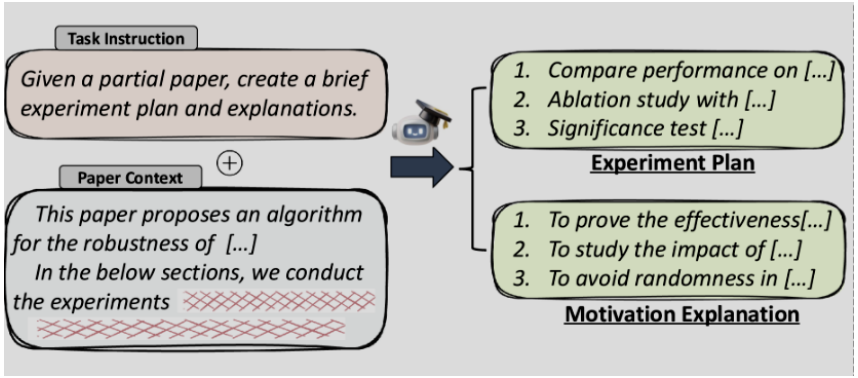
CAN LLMS GENERATE NOVEL RESEARCH IDEAS?
A LARGE-SCALE HUMAN STUDY WITH 100+ NLP RESEARCHERS

Chenglei Si, Diyi Yang, Tatsunori Hashimoto
Stanford University
{clsi,diyiy,thashim}@stanford.edu

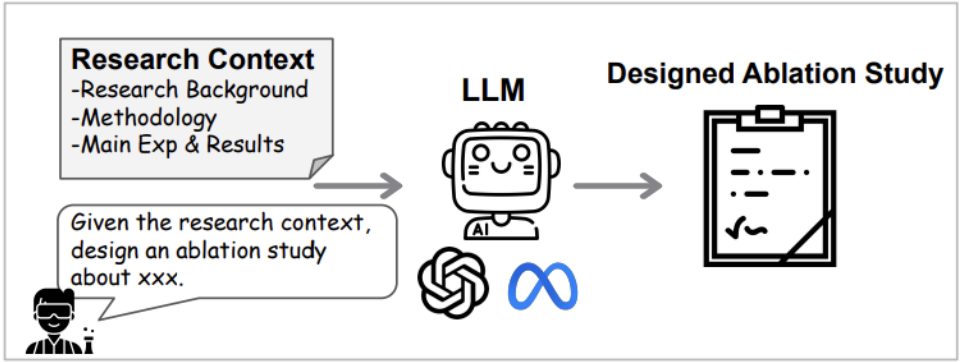


Experiment Design

- **AAAR-1.0** (Lou et al. 2025) ExperimentDesign:
  - Asks models to propose sound setups
  - Scored with expert rubrics



- **AbGen** (Zhao et al. 2025):
  - Asks to ablate on a component
  - Evaluate by human and LLM judge (show **low** correlation)



Experiment Execution (1/2)

- Experiment code generation
  - **SciAgentBench (SAB) (Chen et al. 2025)**: Can an agent write a **self-contained Python program** to complete data-driven science tasks (DL, analysis, viz) end-to-end?
  - **ResearchCodeBench (RCB) (Hua et al. 2025)**: Can an LLM **implement novel ML paper ideas** by filling in real repo code, using the paper text as spec?

Experiment Execution (2/2)

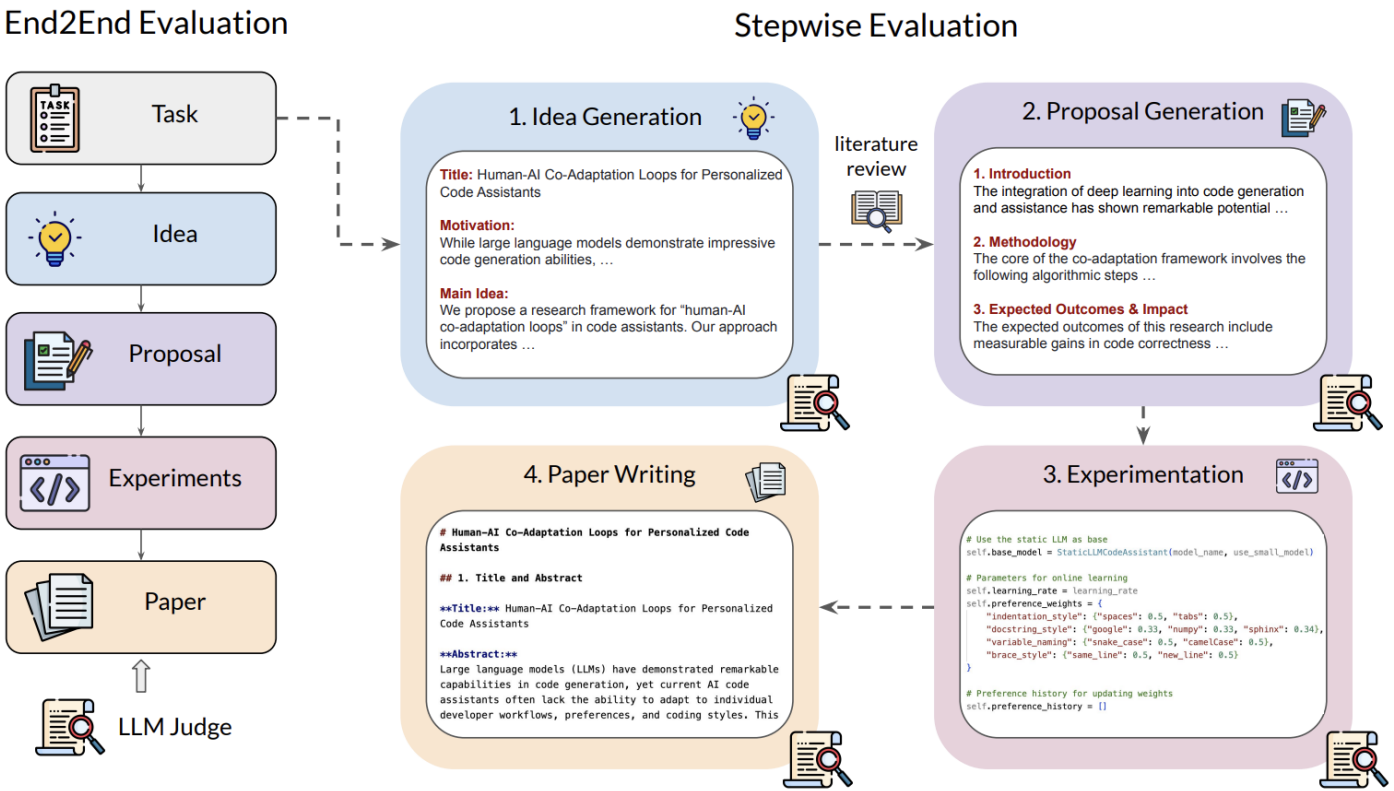
- Environment setup & execution
  - **SUPER** (Bogin et al. 2024): Can an agent **set up & execute ML/NLP research repos “in the wild”**?
  - **PaperBench** (Starace et al. 2025): Can an agent **replicate cutting-edge ML papers from scratch**?

Peer-Review and Feedback Generation

- **ReviewBench** (Gao et al. 2025):
  - Fidelity to **human review structure** and similarity to human comments

- **LimitGen** (Zhao et al. 2025):
  - Can my reviewer agent reliably **find** the key methodological / experimental / lit-review **flaws**

MLR-Bench (Chen et al. 2025): Overall Evaluation

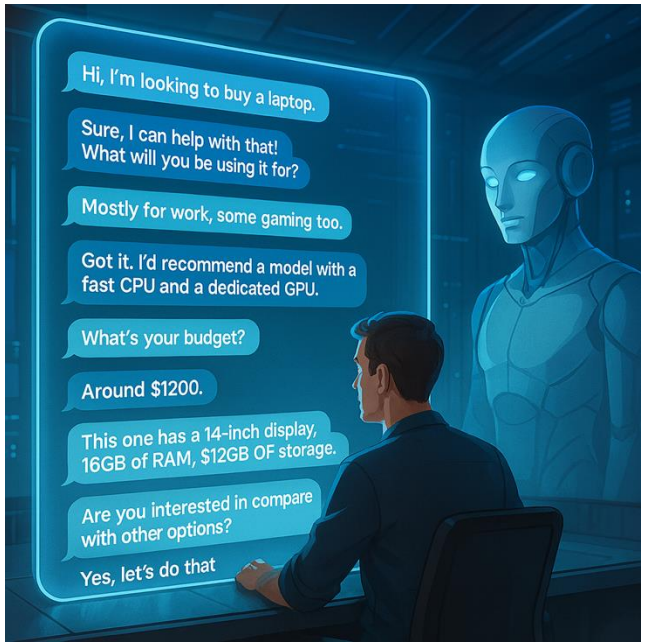


MLR-Bench (Chen et al. 2025): Overall Evaluation

- **201 tasks** curated from recent ICLR/ICML/NeurIPS workshops
- The agent receives a task and must return a **complete paper**
- Agent scaffolding is provided
- MLR-Judge
  - LLM judge with **Gemini-2.5-Pro** and **Claude-3.7-Sonnet**
  - Each stage has a set of rubrics for LLM judge
- The differences between the LLM and human reviewers are not significantly larger than those between two human reviewers

Extended Materials

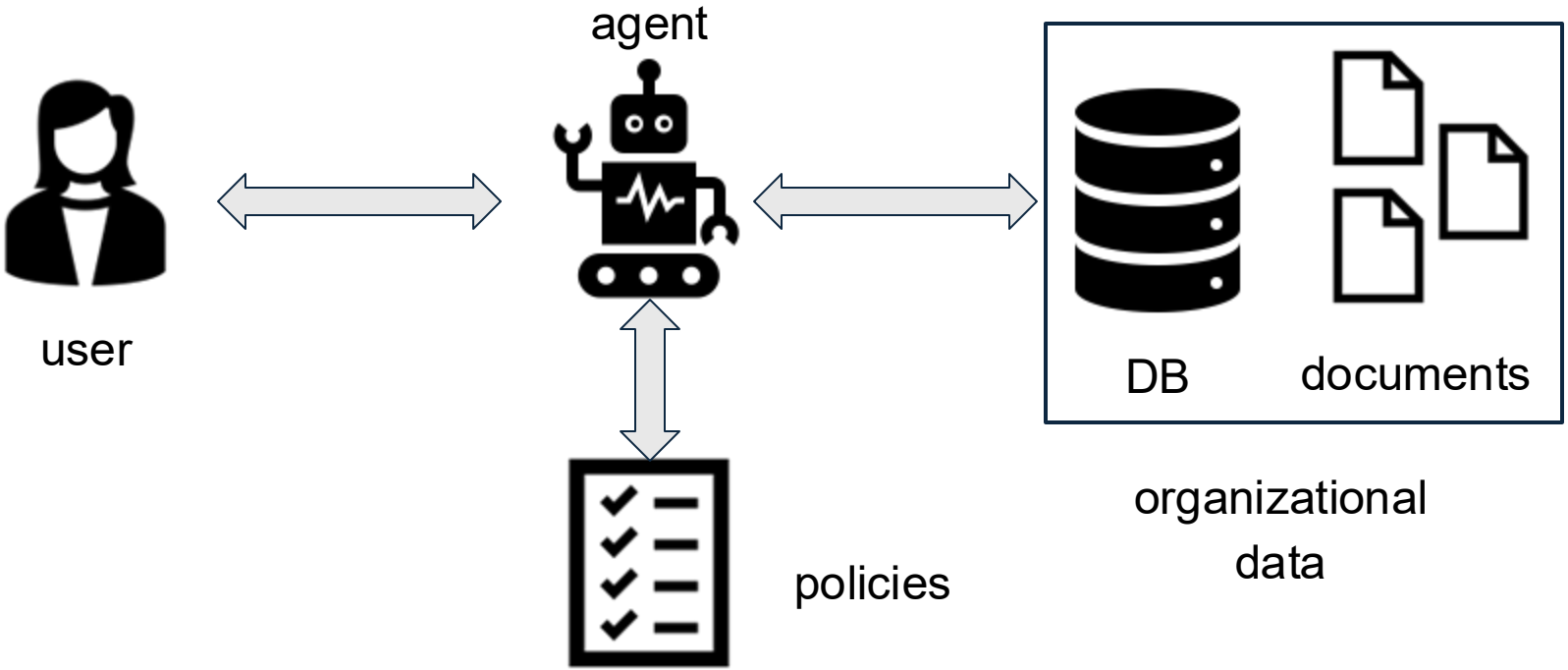
- **AAAR-1.0** (Lou et al. 2024): Can LLMs perform expert-level research work—from formal reasoning to experimental planning and critique—with researcher-grade reliability?
- **CORE-Bench** (Siegel et al. 2024): Can autonomous agents faithfully reproduce published results using authors' code and data?
- **SWIF²T** (Chamoun et al. 2024): Can models generate focused, actionable peer-review feedback that aligns with human judgment and improves drafts?
- **DeepResearch Bench** (Du et al. 2025): Do web-research agents produce citation-grounded reports whose quality aligns with expert evaluators?
- **SciArena** (Zhao et al. 2025): Can community preference judgments reliably rank models on open-ended scientific literature tasks?
- …



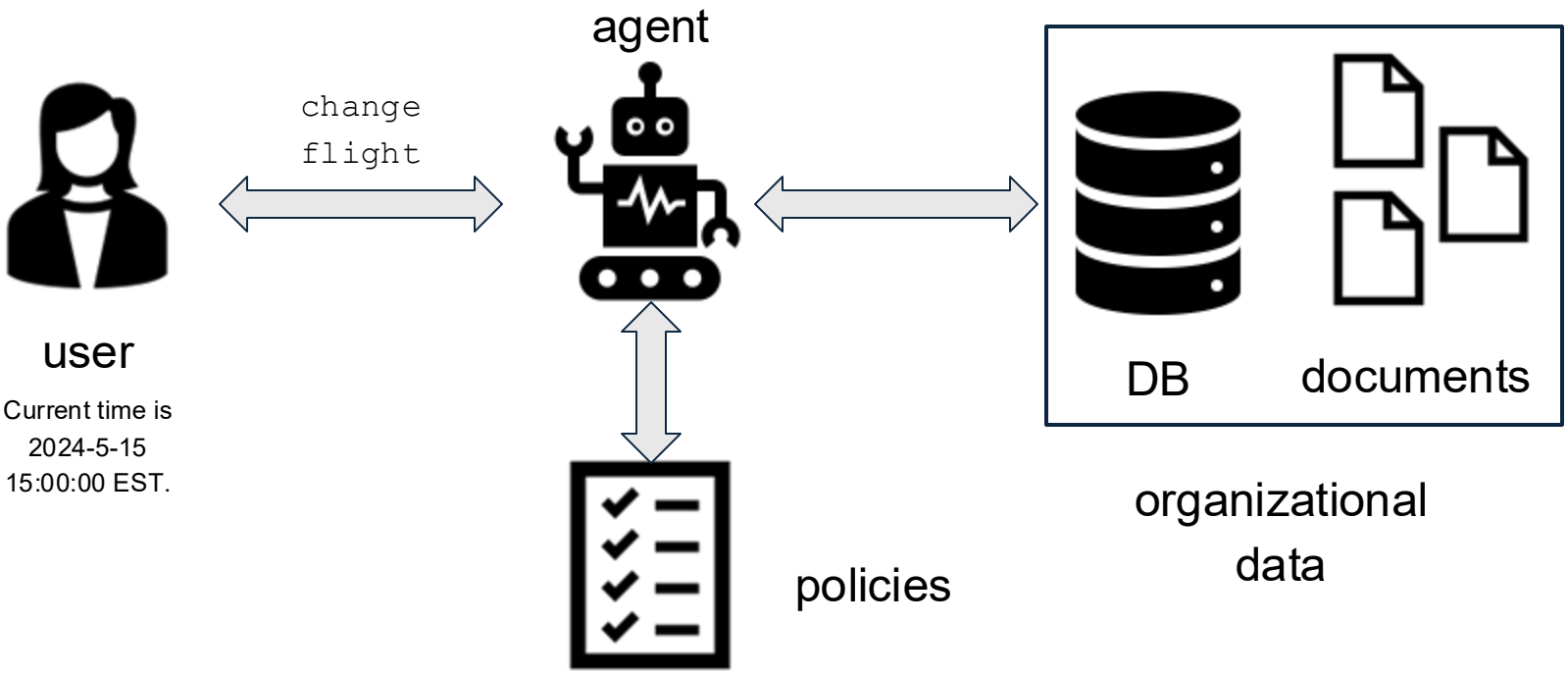
Conversational Agents

Task-Oriented Conversational Agents

- Assist users in completing *specific tasks,* e.g. booking a flight

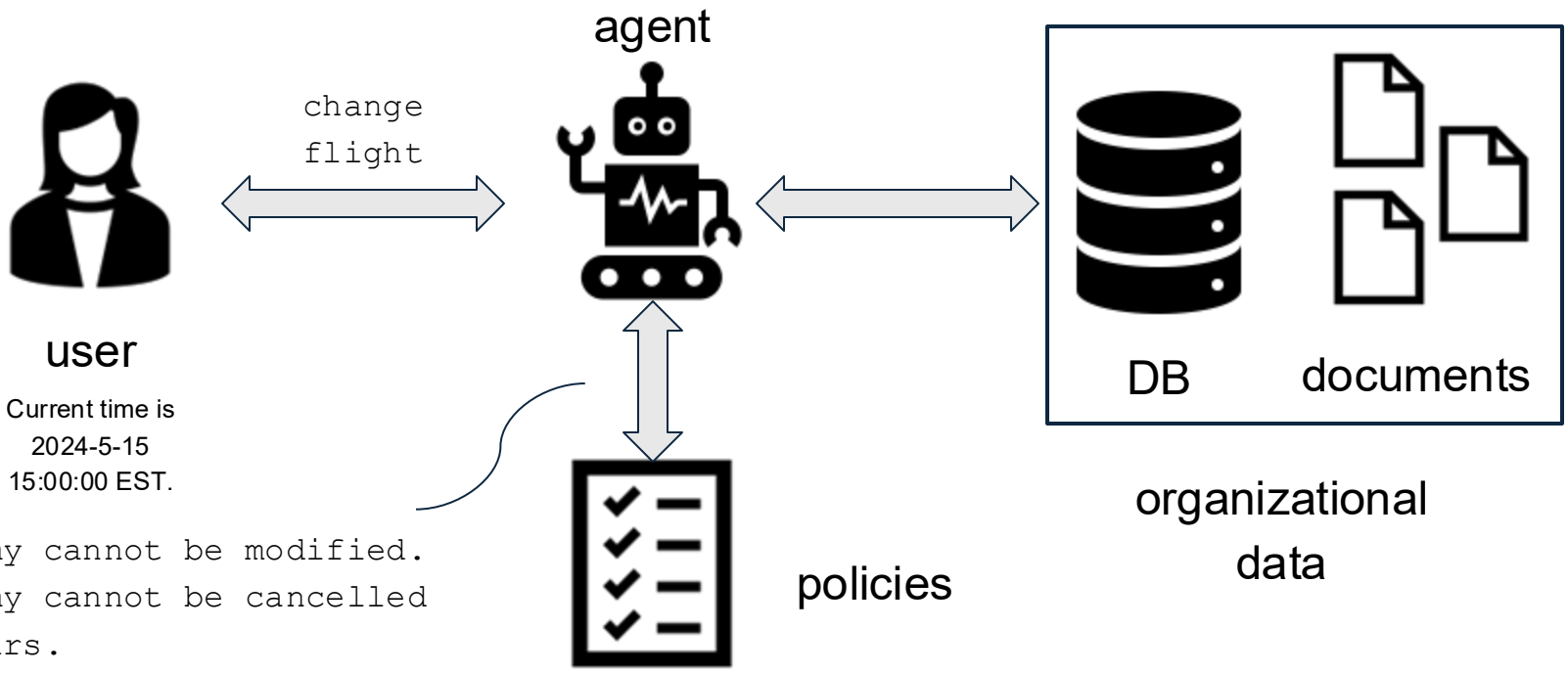


Example: Flight Reservation System



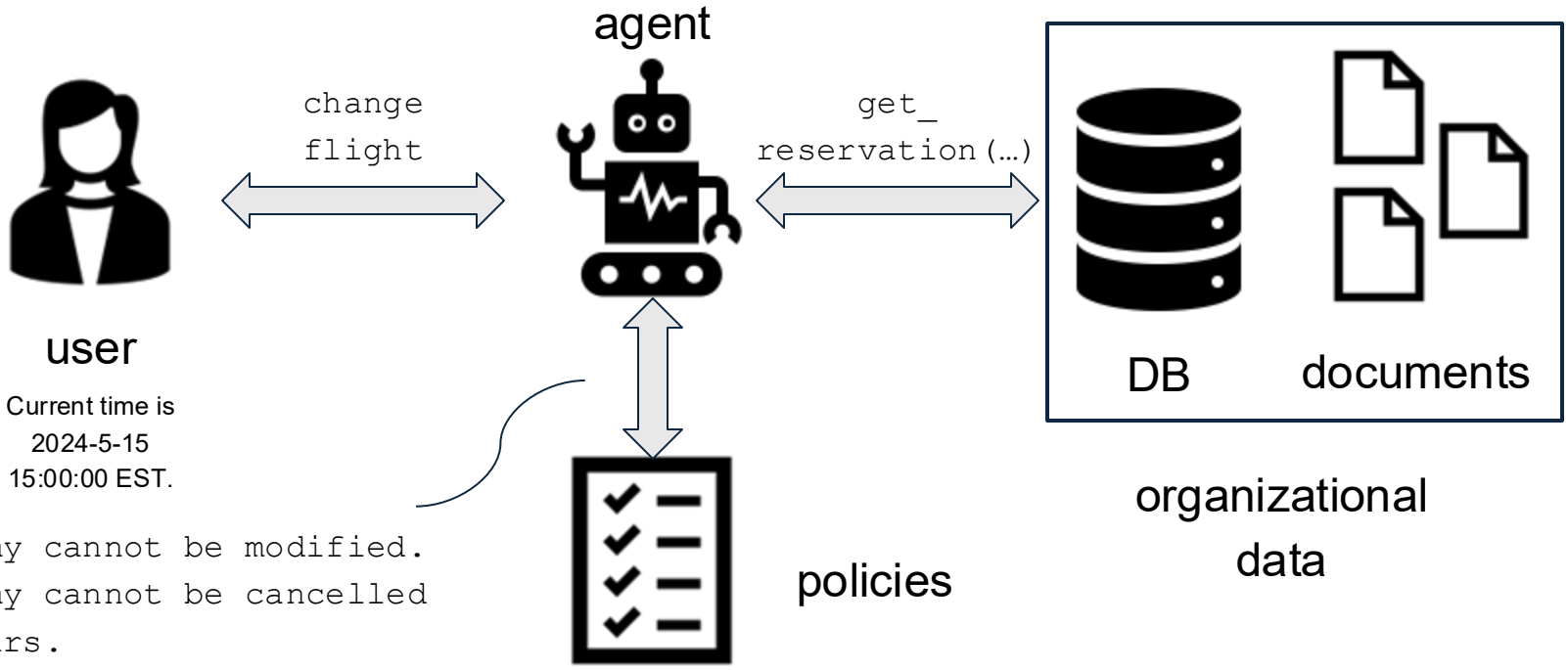
Based on [Yao et al., 2024]

Example: Flight Reservation System



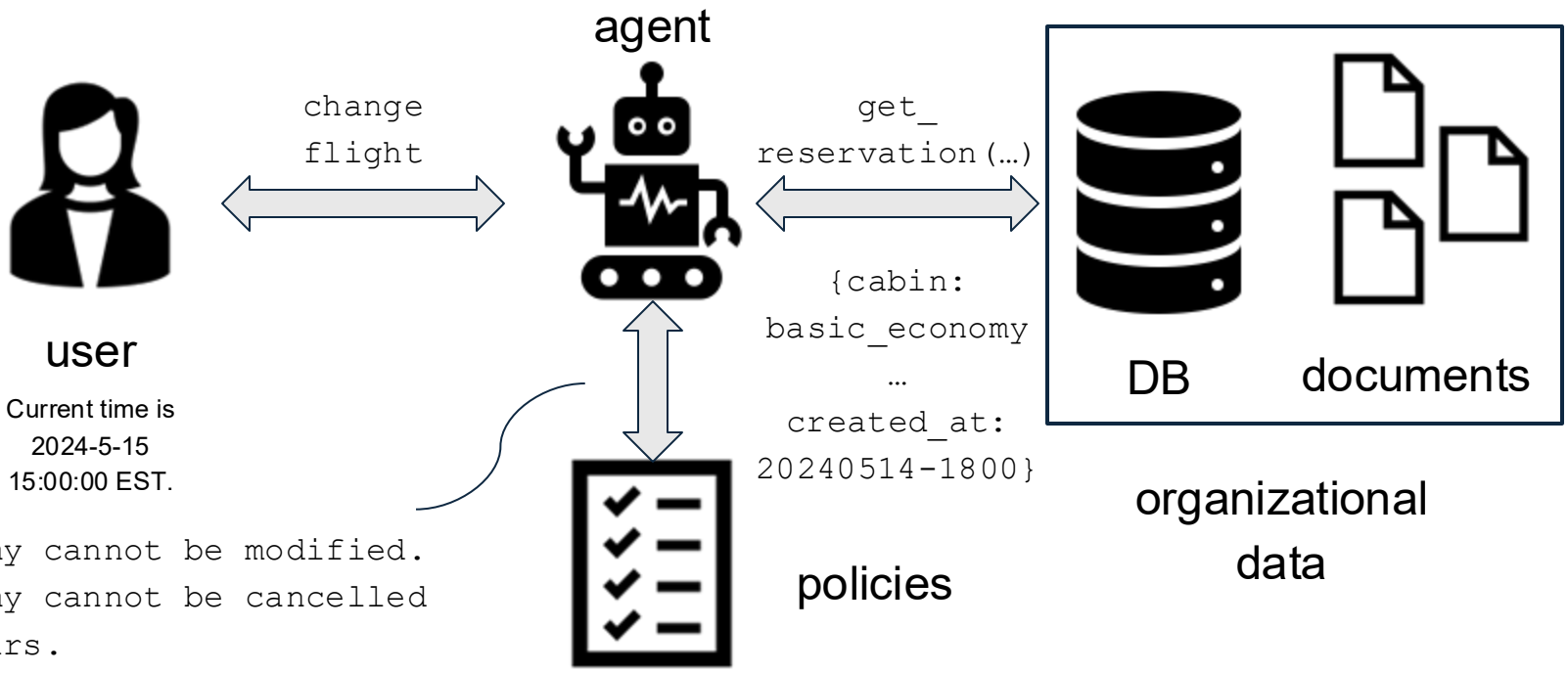
Based on [Yao et al., 2024]

Example: Flight Reservation System



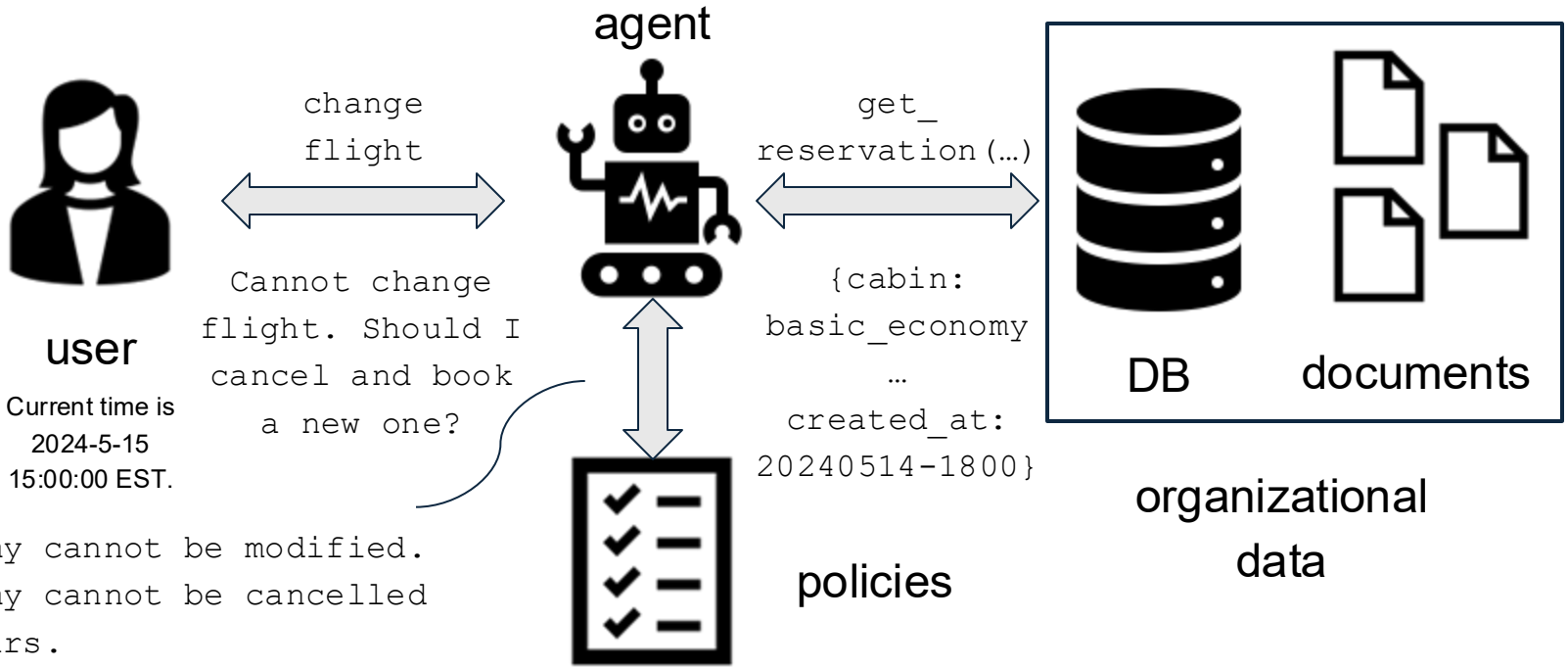
Based on [Yao et al., 2024]

Example: Flight Reservation System



Based on [Yao et al., 2024]

Example: Flight Reservation System



Based on [Yao et al., 2024]

How to Evaluate Conversations?

- Overall task success
  - Final response to the user / database state …
- Intermediate subtasks (given a conversation prefix)
  - Dialogue State Tracking – maintaining structured slot/value representation of the user's intent throughout the conversation
  - Response Generation
- Milestones and minefields
  - Key events that must/must not happen

But first, how do we generate these conversations?

Human-to-Human Dialogue Benchmarks

- Task-oriented dialogue systems have been studied in AI/NLP for many years
- Several earlier benchmarks include crowdsourced human-to-human conversations (annotators taking the roles of both user and agent)
- Still being used to evaluate modern LLM-based agents

Dataset	#Conversations	Description
MultiWOZ [Budzianowski et al., 2018]	10K	multi-domain dialogues *Attraction, Hospital, Police, Hotel, Restaurant, Taxi, Train*
SMCalFlow [Andreas et al., 2020]	42K	conversations about *calendar events, weather, places, and people*
ABCD [Chen et al., 2021]	10K	customer-agent dialogues spanning 55 intents, each requiring a unique sequence of actions defined by the corresponding policy.

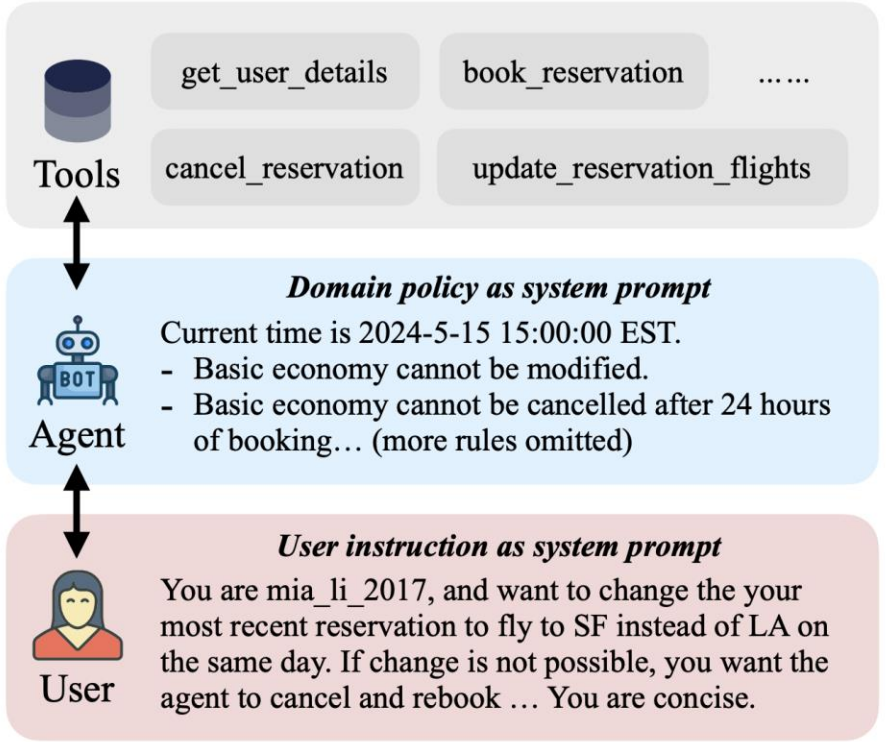
τ-bench: Emulating Dynamic Conversations with Domain-Specific APIs and Policies

[Yao et al., 2024]

DBs + read/write APIs per domain
(**τ-airline**, **τ-retail**)

domain policies as
system prompts to **Agent LLM**

User simulation:
user instructions as
system prompts to **User LLM**



τ-bench: Benchmark Construction

- **Stage I: Manual design of database schema, APIs, and policies**
  - Simplification of real-world use cases
- **Stage II: Automatic data generation with LMs**
  - Data was created by code; code was mostly generated by GPT4
- **Stage III: Manual task annotation and validation with agent runs**
  - Tasks: user instructions + ground truth for DB write actions & (optionally) output for user questions
  - Verify consistent DB outcome for the instructions through multiple runs

τ-bench: Evaluation

- **Reward:**
  - r =1 ⇔ DB state matches ground truth and output contains ground truth
  - Allows diversity of trajectories reaching the same goal
  - Doesn't guarantee successful execution (e.g., the agent did not ask for user confirmation when it should have)

- **Metrics over k trials per task**
  - **Pass@k:** at least one task trial out of k is successful (r=1)
  - **Pass^k:** all k trials are successful
  - Main evaluation metric: pass@1=pass^1

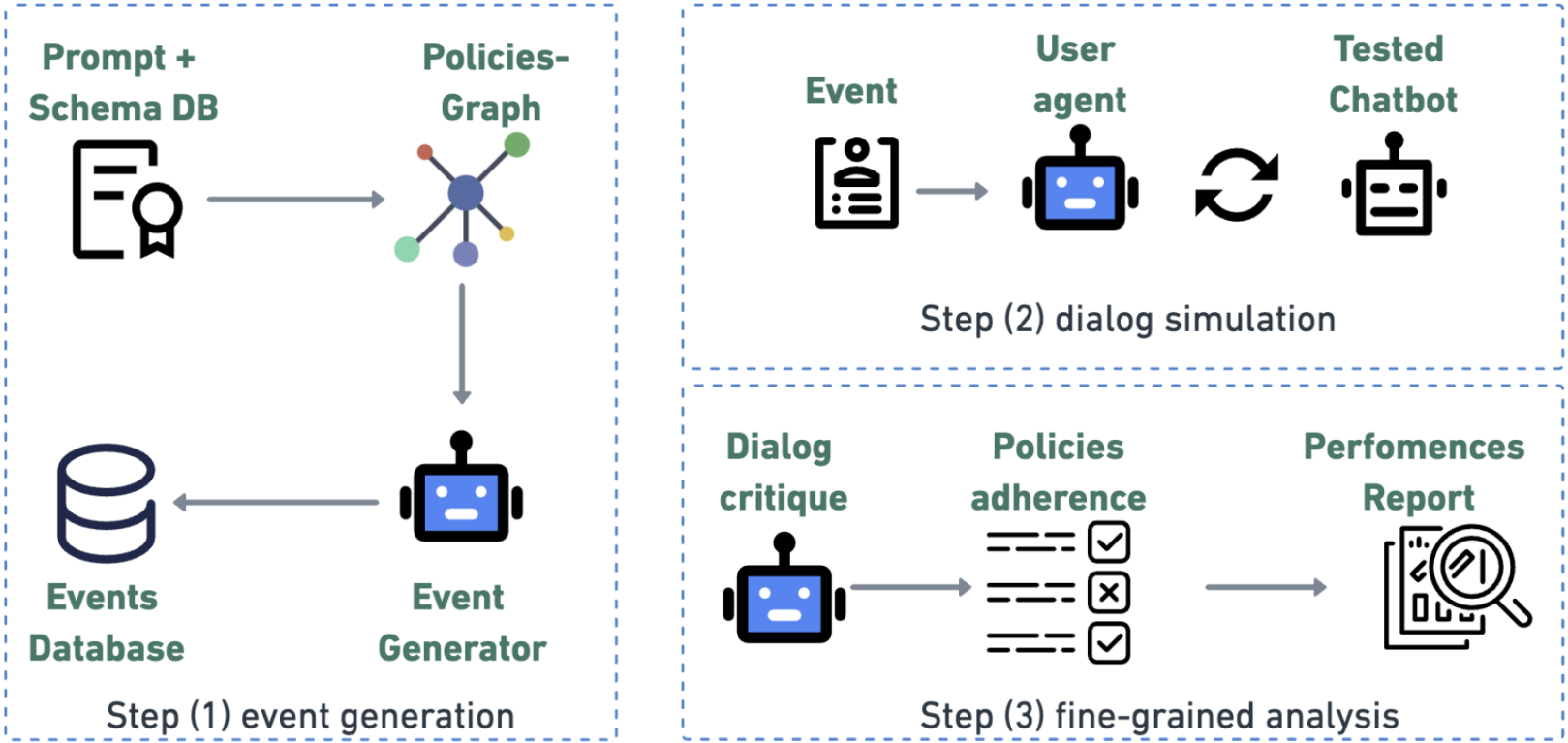
IntellAgent: Automatic Benchmark Generation with Fine-Grained Performance Analysis

[Levy & Kadar, 2025]

- **τ-bench limitations**
  - Tasks were manually created, limiting scalability (115 tasks for τ-retail, 50 for τ-airline)
  - No evaluation of the whole trajectory (only final DB state and output)

- **IntellAgent** addresses these limitations, providing:
  - Automatic generation of test cases (”events”)
  - Fine-grained dialogue evaluation against the given policies

IntellAgent: Overview



IntellAgent: Event Generation

- Policy graph (LLM-generated)
  - **Nodes:** policies (extracted from the prompt) + complexity score
  - **Edges:** co-occurrence likelihood
- Event:
  - List of policies + user request aligning with policies + DB initial state
  - Event complexity: sum of policy complexities
- Event generation
  - Sample a list of policies from the graph via random walk
  - Create an event scenario from the policies
  - Identify and instantiate relevant entities; create corresponding DB entries

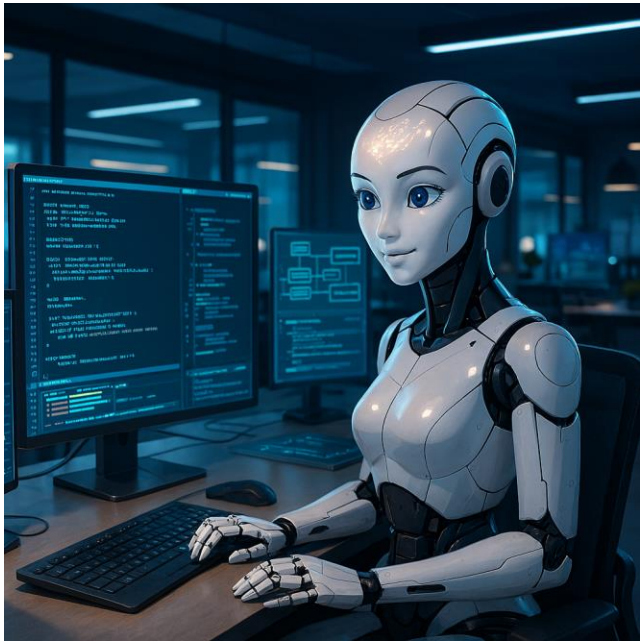
IntellAgent: Experimental Results

Model	τ-airline	IntellAgent-airline	τ-retail	IntellAgent-retail
claude-3.5-sonnet	0.46	0.70	0.69	0.71
gpt-4o	0.44	0.70	0.51	0.68
gemini-1.5-pro	0.34	0.63	0.43	0.58
gpt-4o-mini	0.30	0.55	0.46	0.62
claude-3.5-haiku	0.28	0.53	0.44	0.56
gemini-1.5-flash	0.21	0.40	0.31	0.48

- Success rates for τ-bench and IntellAgent are highly correlated
  - Pearson correlation of 0.98 / 0.92 for Airline / Retail

Conversational Agents Evaluation – Trends

- From static to dynamic conversations
  - User simulation by LLM
- More complex and realistic setting
  - APIs, databases, policies
- More fine-grained assessment
  - Beyond final result
- Increasing automation of data generation
  - DB instantiation
  - Test case generation
- Quality assessment of synthetic data remains a challenge



Software Engineering (SWE) Agents

Evaluating Software Engineering Agents

- Code generation is a primary use case of generative/agentic AI
- Huge amount of publicly-available code to train and test on
- Can be verified automatically by running unit tests

HumanEval: Generate Python Functions From Docstrings [Chen et al., 2021]

- Developed by OpenAI for evaluating their Codex code generation model (initial model powering GitHub Copilot)
- 164 hand-written problems with unit tests

```
def solution(lst):
    """Given a non-empty list of integers, return the sum of all of the odd elements
    that are in even positions.

    Examples
    solution([5, 8, 7, 1]) ==>12
    solution([3, 3, 3, 3, 3]) ==>9
    solution([30, 13, 24, 321]) ==>0
    """
    return sum(lst[i] for i in range(0,len(lst)) if i % 2 == 0 and lst[i] % 2 == 1)
```

HumanEval: Evaluation Takeaways

- **Functional Correctness**
  - A solution is correct if all its unit tests passed
  - As opposed to matching to reference solution

- **Execute solutions in a sandbox**
  - Protect against side-affect of malicious and incorrect code

- **Pass@k evaluation**
  - Sample k solutions and select the one that passes the unit tests
  - Codex-S reached 77.5% for pass@100
  - With only one try, choosing the solution with highest log-mean prob is beneficial (solving 44.5% of the problems)

MBPP & MathQA-Python

[Austin et al., 2021]

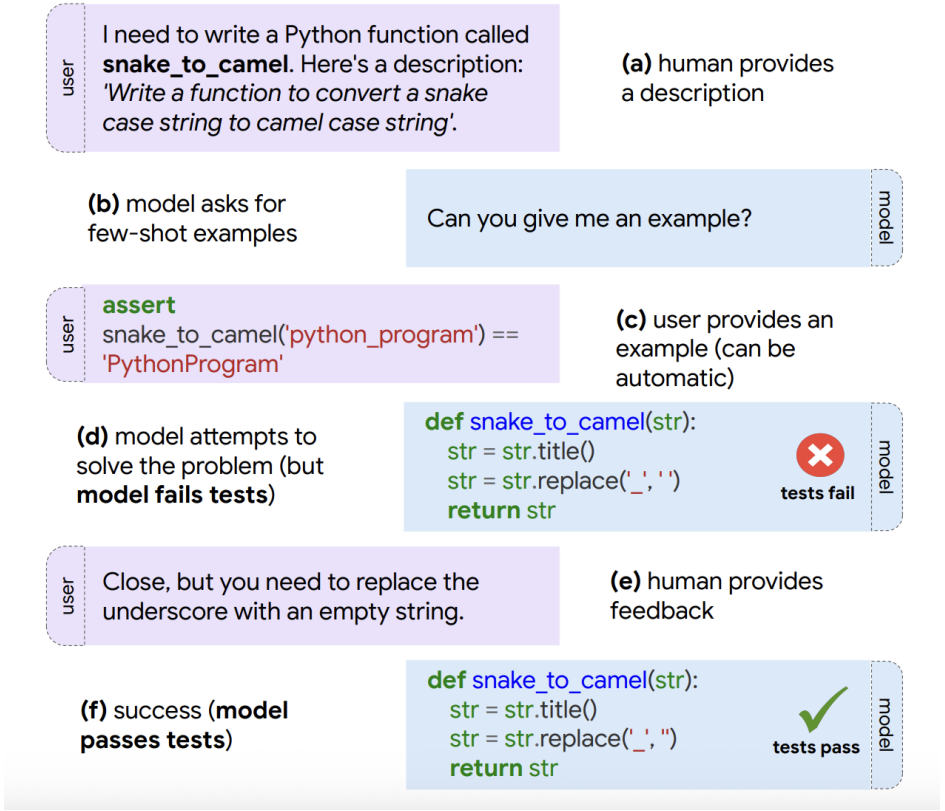
- Work from Google Research, done in parallel to [Chen et al., 2021]

- **MBPP** (“Mostly Basic Programming Problems”)
  - 974 short Python programs collected via crowdsourcing
  - Problem statement + solution (Python function) + three test cases
  - Hand-verified subset (426 examples)
  - Similar to HumanEval

- **MathQA-Python**
  - Mathematical word problems
  - Adapted from the MathQA dataset [Amini et al.,2019] by translating solutions to Python
  - Code is mostly straight line (less control flow structures), but problem descriptions are more complex

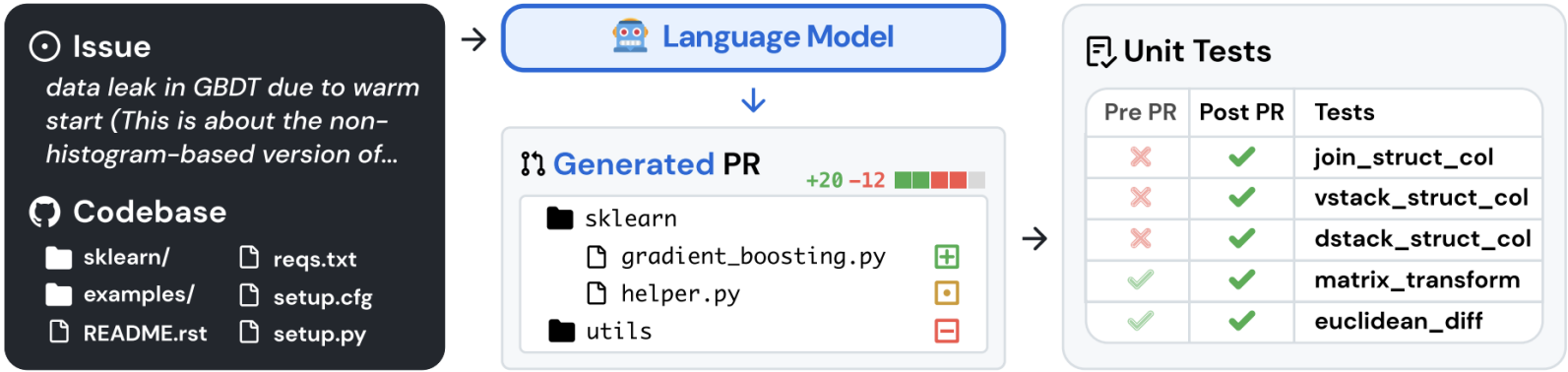
MBPP: Human-Model Collaboration

- Evaluated human collaboration on 50 samples
- Human provided one sentence hints, up to 4 hints
- Increased fraction of solved problems from 30% to over 65%
- Human hints often clarify underspecified descriptions, and help fixing minor errors (e.g., missing imports)



SWE-Bench

[Jimenez et al., 2024]



- 2,294 task instances from real GitHub issues
- Input: issue text description and codebase
- Model is required to create a patch to resolve issue
- Patch is applied successfully if all associated unit and system tests pass

SWE-Bench Features

- Real-world software engineering tasks
- Easy to update and extend with minimal human supervision
- Issues are long and detailed
- Identifying the few lines to be fixed (in one or more files) within a large code base is very challenging
- Wide scope for possible solutions

SWE-Bench ***Lite, Verified*** et al.

- **SWE-Bench Lite**
  - Subset of 300 tasks, focusing on self-contained, functional bug fixes
  - Reduce evaluation costs, more accessible entry point for researchers
  - Released along with SWE-Bench

- **SWE-Bench Verified** [Chowdhury et al., 2024]
  - Subset of 500 samples verified to be non-problematic by professional software developers (OpenAI / SWE-Bench authors collaboration)

- And more…
  - **SWE-Bench Multimodal** [Yang et al., 2024], **SWE-Bench Multilingual** [Yang et al., 2025], **SWE-Bench+** [Aleithan et al., 2024]

SWE-Lancer: Real-World Freelance SWE Tasks

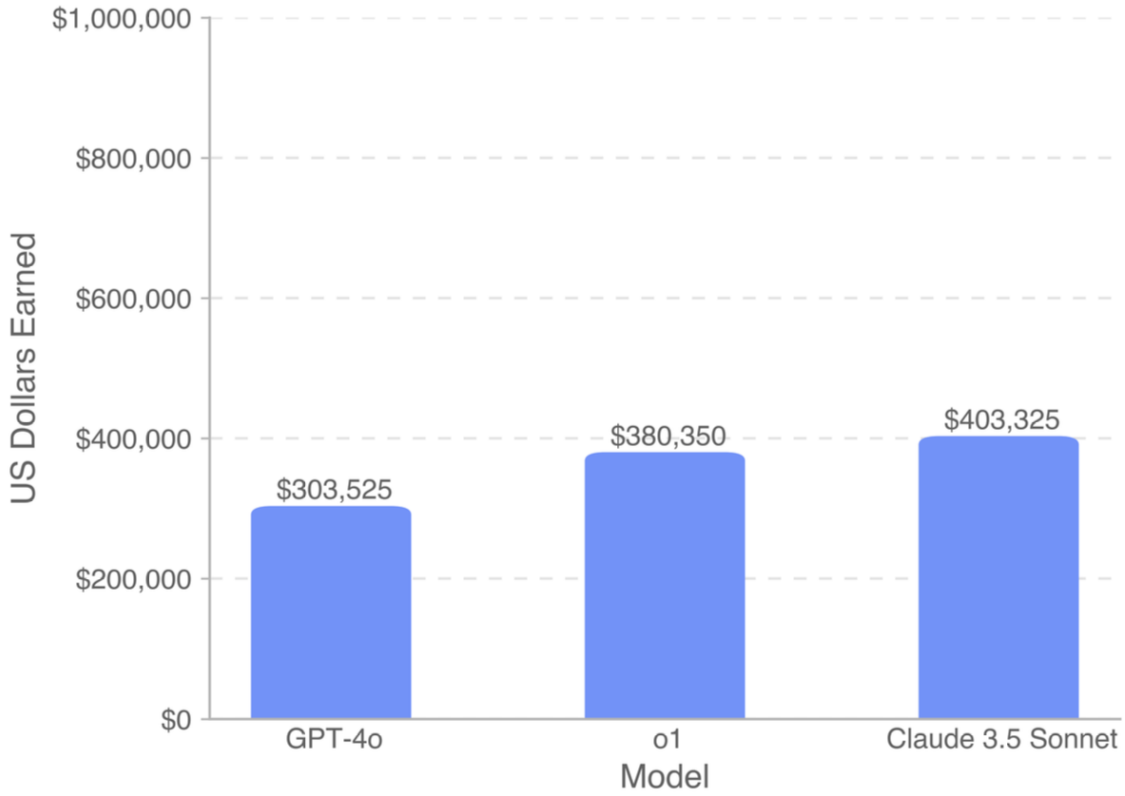
[Miserendino et al., 2025]

- 1,488 freelance software engineering jobs from the Upwork work marketplace
  - Collectively worth $1M in payouts

- **Individual Contributor (IC) tasks**
  - Generate code patches to solve real-world issues
  - From simple bug fixes to developing new features
  - End-to-end tests developed by professional engineers, using browser automation

- **SWE Manager tasks**
  - Review competing proposals submitted by freelancers and select the best one
  - Assessed against the choices of original engineering managers

SWE-Lancer: Main Results



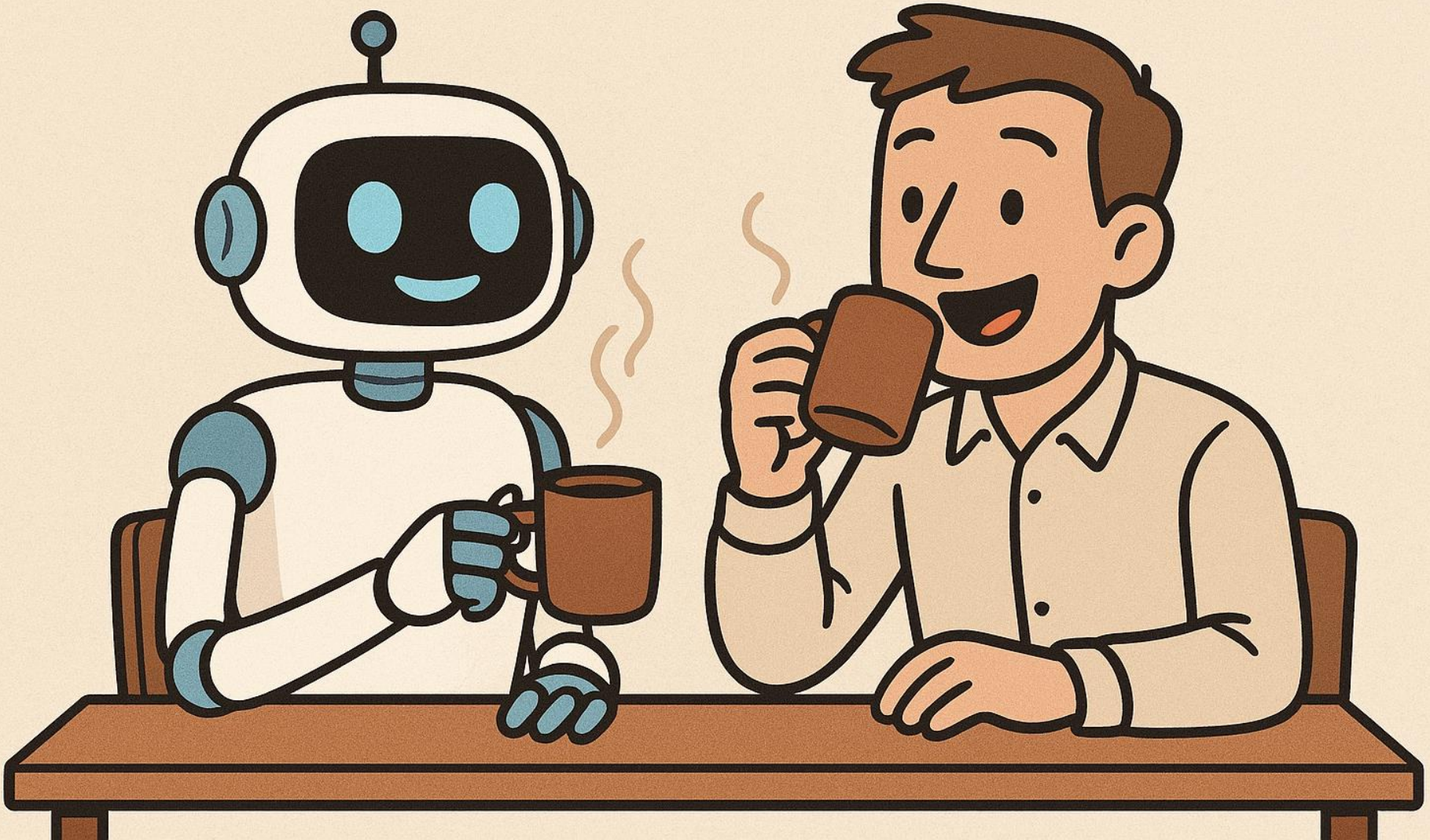
SWE Agents Evaluation – Trends

- Dramatic progress over the past few years, major players involved (Google, OpenAI...)
- From writing self-contained functions to solving real-world issues in complex environments (complete code bases)
- Continuous effort to improve the quality of the datasets
  - Validation by skilled software engineers is important
- Expanding to new tasks
  - Different languages and modalities
  - From solving code issues to evaluating solutions

SWE Agents Evaluation – Trends

- Dramatic progress over the past few years, major players involved (Google, OpenAI...)
- From writing self-contained functions to solving real-world issues in complex environments (complete code bases)
- Continuous effort to improve the quality of the datasets
  - Validation by skilled software engineers is important
- Expanding to new tasks
  - Different languages and modalities
  - From solving code issues to evaluating solutions

Questions?



Part 4: Generalist Agents

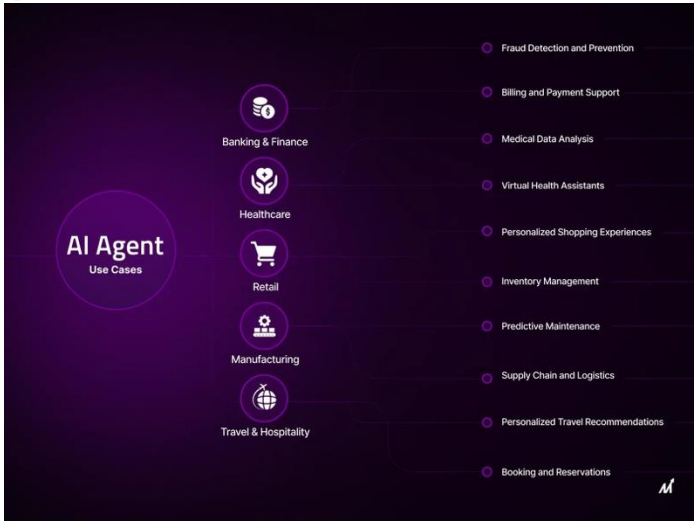
AI & Machine Learning

601 real-world gen AI use cases from the world's leading organizations

April 9, 2025

AI Agents Use Cases: 50+ Applications Transforming Industries

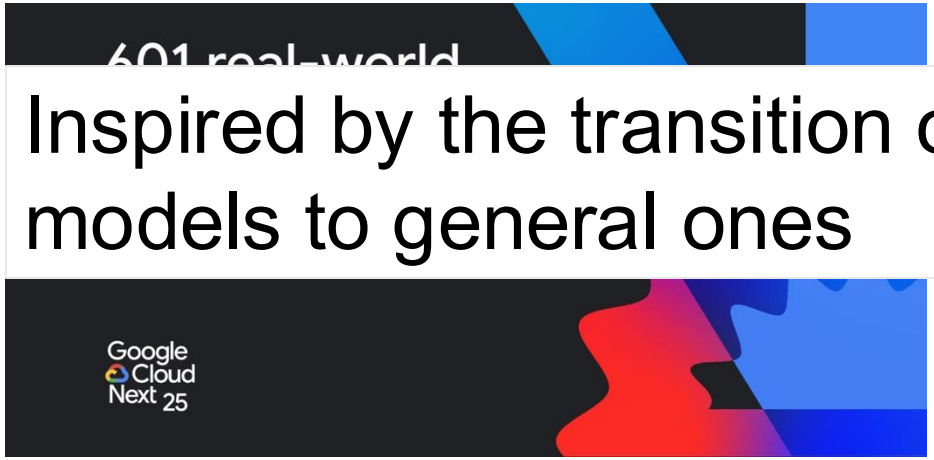
RAJEEV SHARMA Last Updated on: May 26, 2025 | 16 Mins Read



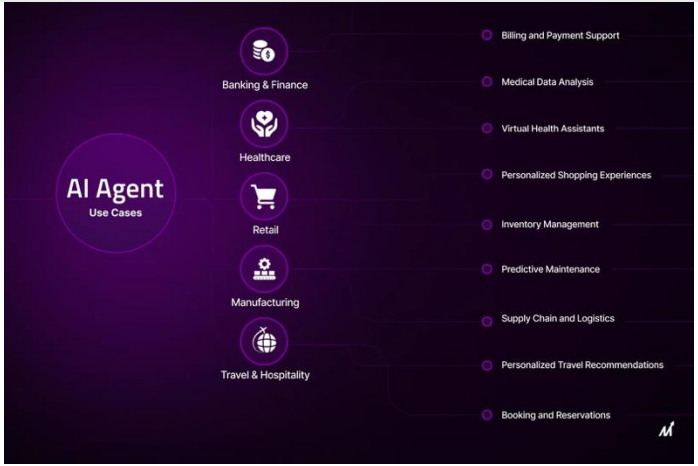
AI & Machine Learning

601 real-world gen AI use cases from the world's leading organizations

April 9, 2025



Inspired by the transition of LLM from specialized models to general ones



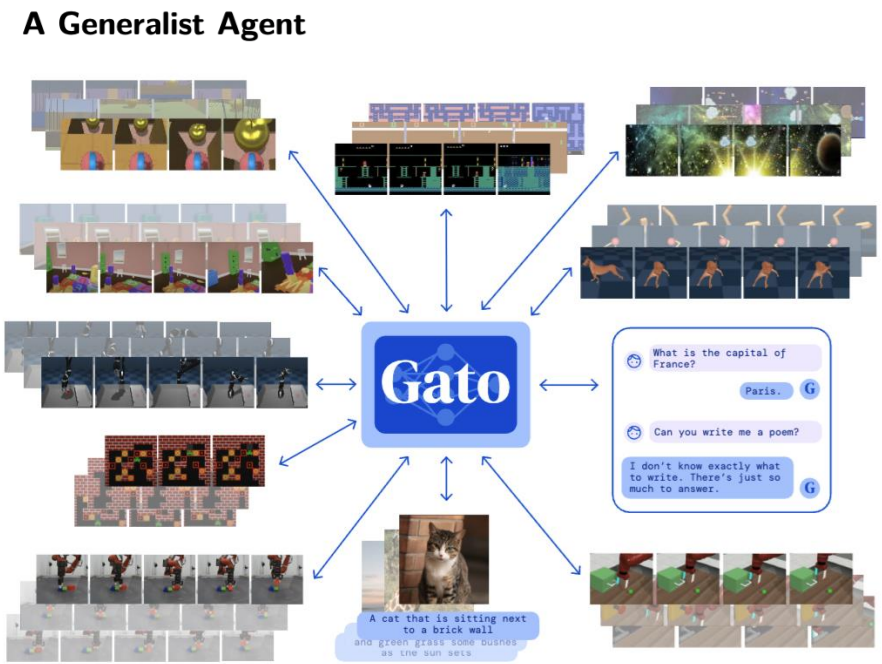
Generalist Agents Evaluation

- What is ″Generalist Agent″?
- Two approaches for Generalist Agents Evaluation
- Future Challenges

What is ”Generalist Agent”?

“The agent works as a multi-modal, multi-task, multi-embodiment generalist policy.”

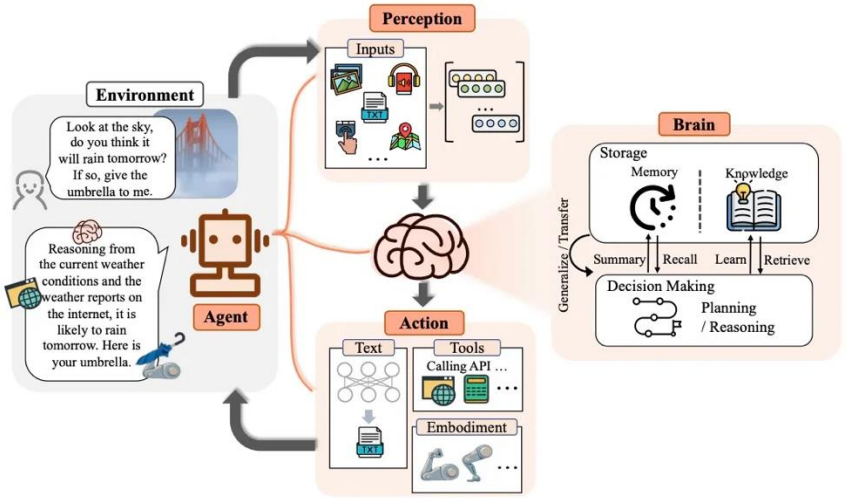
Published in Transactions on Machine Learning Research (11/2022)



Reed, Scott, et al. "A generalist agent." *arXiv preprint arXiv:2205.06175* (2022).

Generalist Agent

General purpose agent – an LLM-based agent that can interact with multi-modal environment to achieve a variety of tasks.



Cheng, Yuheng, et al. "Exploring large language model based intelligent agents: Definitions, methods, and prospects." *arXiv preprint arXiv:2401.03428* (2024).

So, how can we evaluate Generalist Agents?

Two approaches for Generalist Agents Evaluation

1. Building general benchmark that require multiple tools, and capabilities

Two approaches for Generalist Agents Evaluation

1. Building general benchmark that require multiple tools, and capabilities
2. Unifying many agent specific benchmarks

Generalist benchmarks

- GAIA
- AppWorld
- OSWorld
- TheAgentCompany

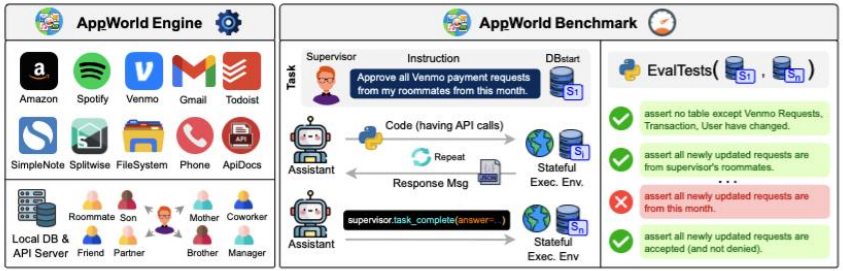
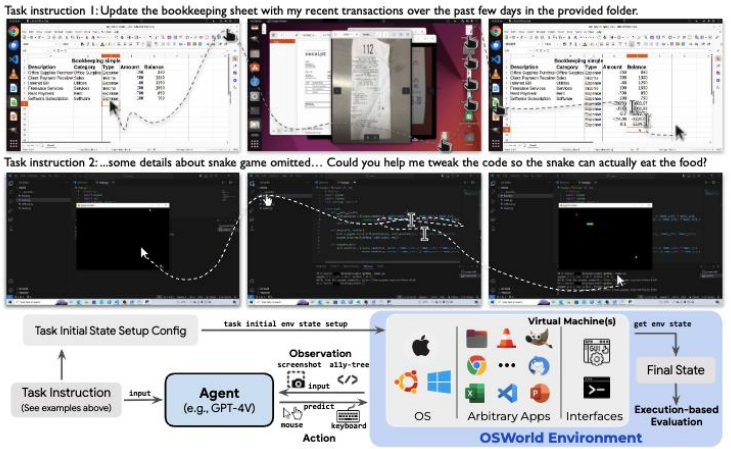


Figure 2: (Left) AppWorld Engine is a feature-rich API-based app simulator based on 9 real-world day-to-day apps populated with data simulating the digital activities of fictitious people living in this world. (Right) AppWorld Benchmark is a collection of complex everyday tasks requiring interactive coding with API calls, which come with reliable programmatic state-based evaluation suites robustly capturing goal completion criteria.



GAIA:
A Benchmark for General AI Assistants

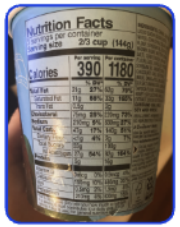
Grégoire Mialon[1], **Clémentine Fourrier**[2], **Craig Swift**[3], **Thomas Wolf**[2], **Yann LeCun**[1], **Thomas Scialom**[4]

[1]FAIR, Meta, [2]HuggingFace, [3]AutoGPT, [4]GenAI, Meta

Level 1

Question: What was the actual enrollment count of the clinical trial on H. pylori in acne vulgaris patients from Jan-May 2018 as listed on the NIH website?
Ground truth: 90

Level 2



Question: If this whole pint is made up of ice cream, how many percent above or below the US federal standards for butterfat content is it when using the standards as reported by Wikipedia in 2020? Answer as + or - a number rounded to one decimal place.
Ground truth: +4.6

Level 3

Question: In NASA's Astronomy Picture of the Day on 2006 January 21, two astronauts are visible, with one appearing much smaller than the other. As of August 2023, out of the astronauts in the NASA Astronaut Group that the smaller astronaut was a member of, which one spent the least time in space, and how many minutes did he spend in space, rounded to the nearest minute? Exclude any astronauts who did not spend any time in space. Give the last name of the astronaut, separated from the number of minutes by a semicolon. Use commas as thousands separators in the number of minutes.
Ground truth: White; 5876

Mialon, Grégoire, et al. "Gaia: a benchmark for general ai assistants." *The Twelfth International Conference on Learning Representations*. 2023.

System prompt: You are a general AI assistant. I will ask you a question. Report your thoughts, and finish your answer with the following template: FINAL ANSWER: [YOUR FINAL ANSWER].
YOUR FINAL ANSWER should be a number OR as few words as possible OR a comma separated list of numbers and/or strings.
If you are asked for a number, don't use comma to write your number neither use units such as $ or percent sign unless specified otherwise.
If you are asked for a string, don't use articles, neither abbreviations (e.g. for cities), and write the digits in plain text unless specified otherwise.
If you are asked for a comma separated list, apply the above rules depending of whether the element to be put in the list is a number or a string.

GAIA Question: The attached Excel file contains the sales of menu items for a local fast-food chain. What were the total sales that the chain made from food (not including drinks)? Express your answer in USD with two decimal places.

uploaded.xlsx

GPT4 (Advanced Data Analysis mode):

Working:

```
import pandas as pd

# Load the Excel file
file_path = '/mnt/data/uploaded.xlsx'
df = pd.read_excel(file_path)
```

Working:

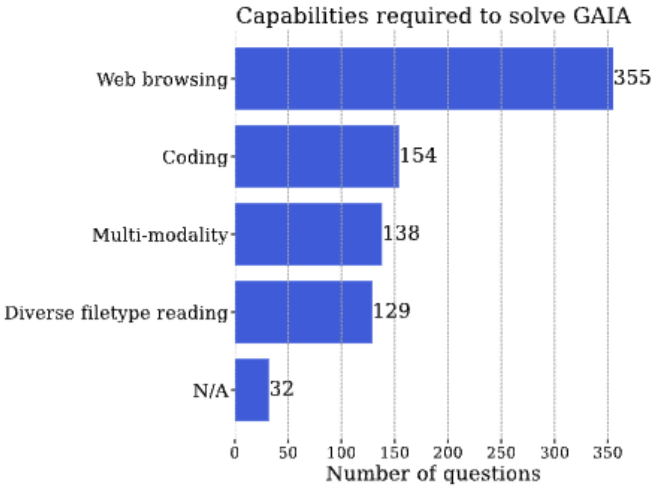
```
# Calculate the total sales from food items (not including drinks i.e., Soda)
total_food_sales = df[
    ['Burgers', 'Hot Dogs', 'Salads', 'Fries', 'Ice Cream']
].sum().sum()

# Format the result to USD, with a dollar sign and two decimal places
total_food_sales_formatted = f"$~total_food_sales:,.2f~"
```

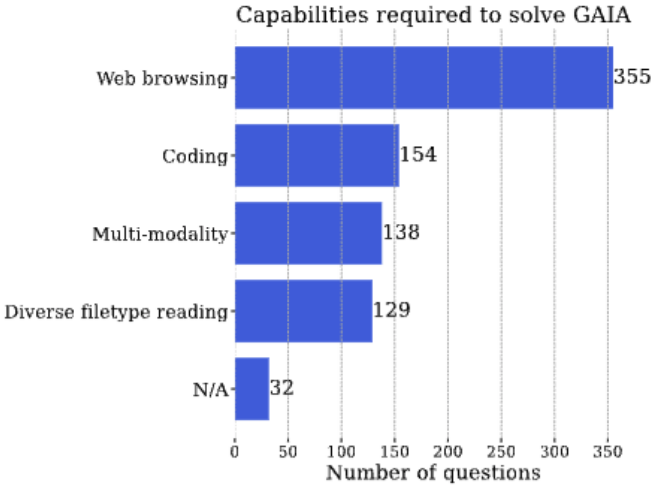
FINAL ANSWER: $89706.00 **Ground truth:** 89706.00 ✓

Mialon, Grégoire, et al. "Gaia: a benchmark for general ai assistants." *The Twelfth International Conference on Learning Representations*. 2023.

GAIA levels & capabilities

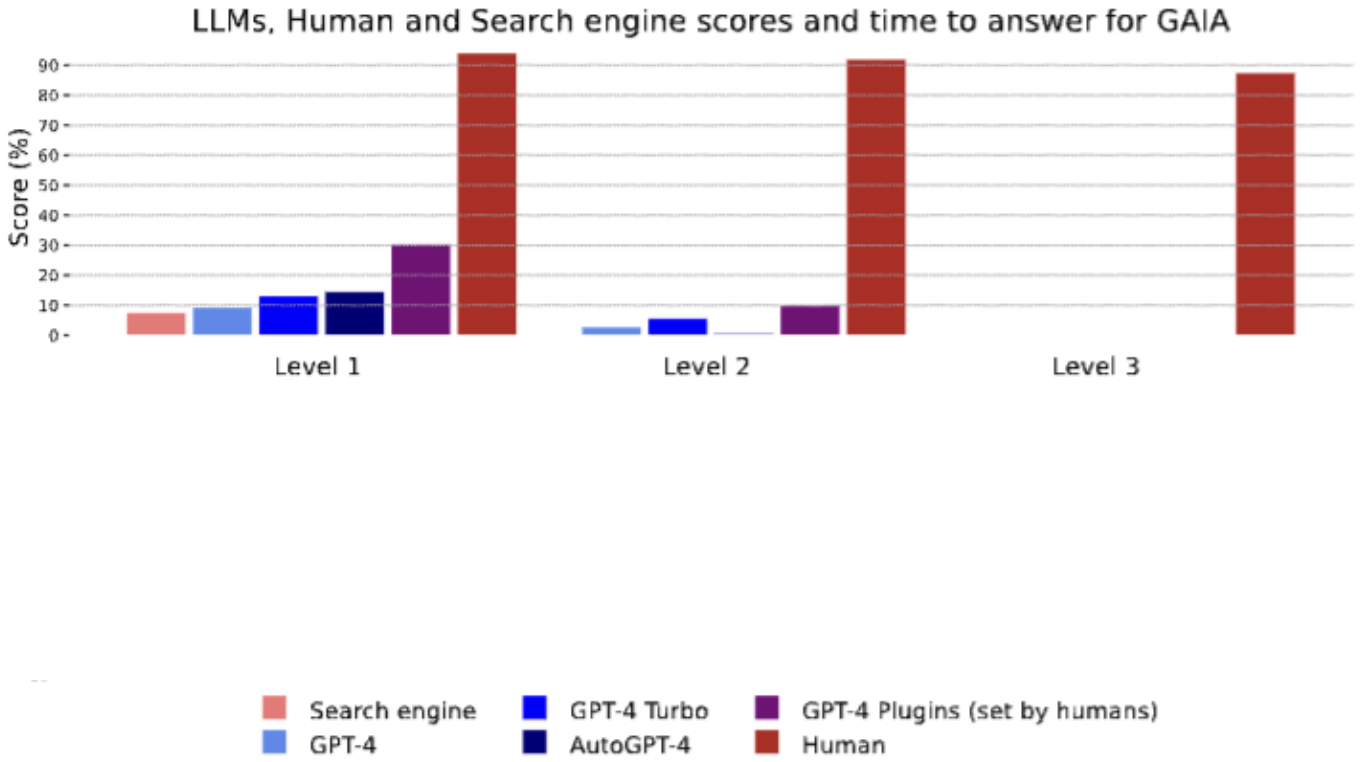


GAIA levels & capabilities

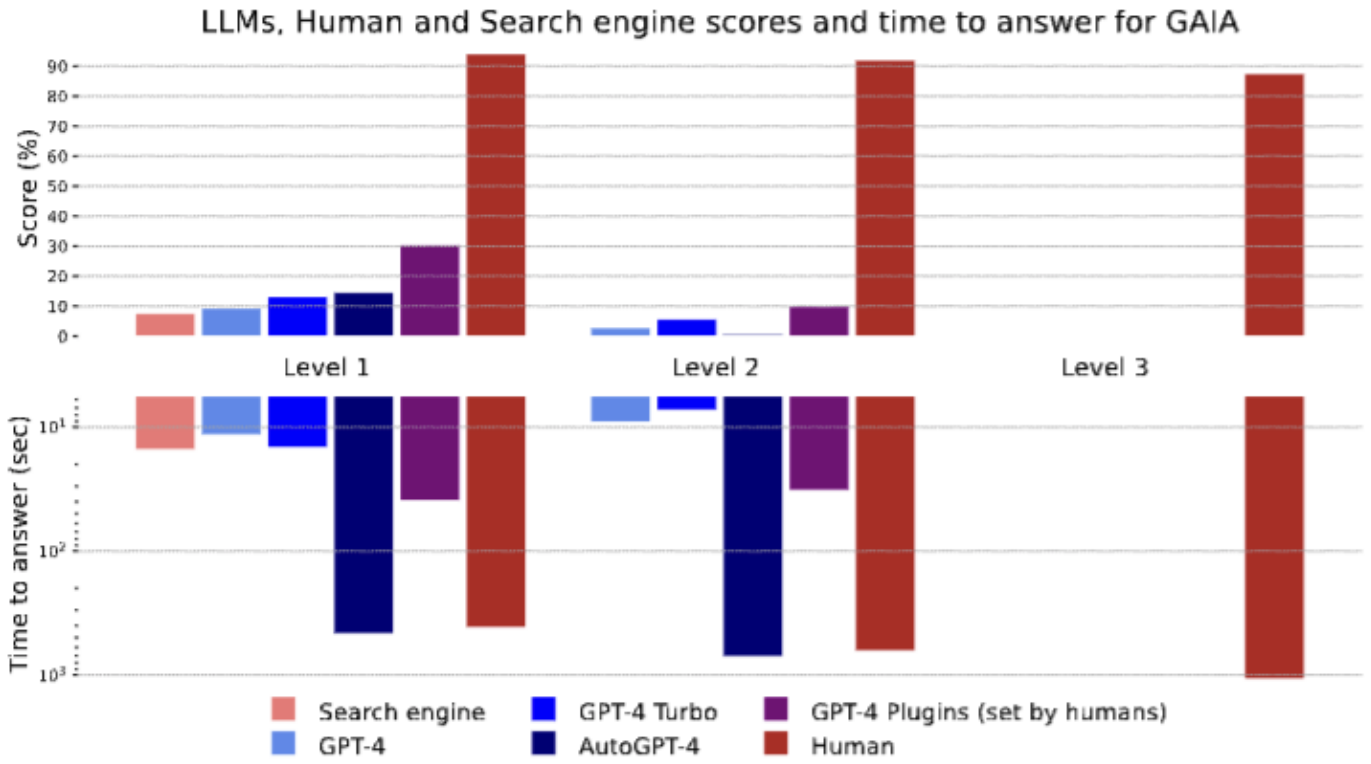


- **Level 1** questions generally require no tools, or at most one tool but no more than 5 steps.
- **Level 2** question generally involve more steps, roughly between 5 and 10 and combining different tools is needed.
- **Level 3** are questions for a near perfect general assistant, requiring to take arbitrarily long sequences of actions, use any number of tools, and access to the world in general.

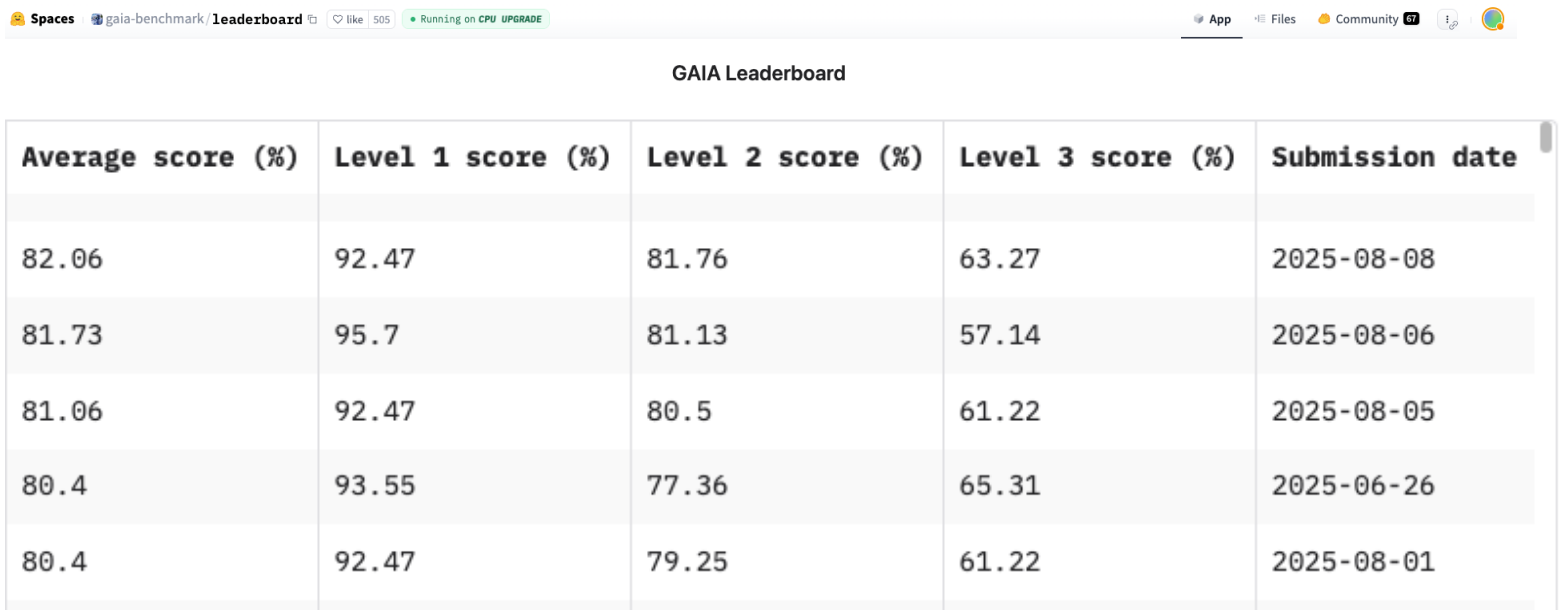
Comparison with Humans



Comparison with Humans



GAIA Leaderboard



Average score (%)	Level 1 score (%)	Level 2 score (%)	Level 3 score (%)	Submission date
82.06	92.47	81.76	63.27	2025-08-08
81.73	95.7	81.13	57.14	2025-08-06
81.06	92.47	80.5	61.22	2025-08-05
80.4	93.55	77.36	65.31	2025-06-26
80.4	92.47	79.25	61.22	2025-08-01

AppWorld: A Controllable World of Apps and People for Benchmarking Interactive Coding Agents*

Harsh Trivedi[†] and **Tushar Khot**[‡] and **Mareike Hartmann**[◇]
Ruskin Manku[†] and **Vinty Dong**[†] and **Edward Li**[†] and **Shashank Gupta**[‡]
Ashish Sabharwal[‡] and **Niranjan Balasubramanian**[†]

AppWorld: A Controllable World of Apps and People for Benchmarking Interactive Coding Agents*

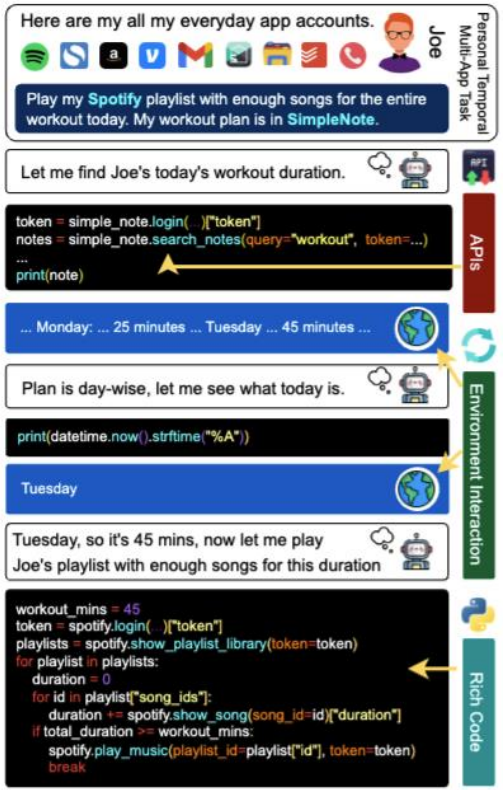


Figure 1: A (shortened) day-to-day task from AppWorld Benchmark requiring **rich code** with **environment interaction** and **API calls**.[2]

AppWorld: A Controllable World of Apps and People for Benchmarking Interactive Coding Agents*

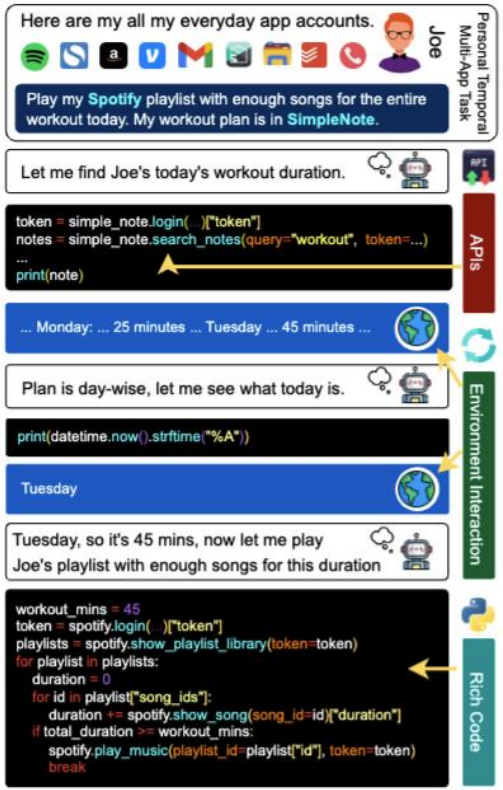
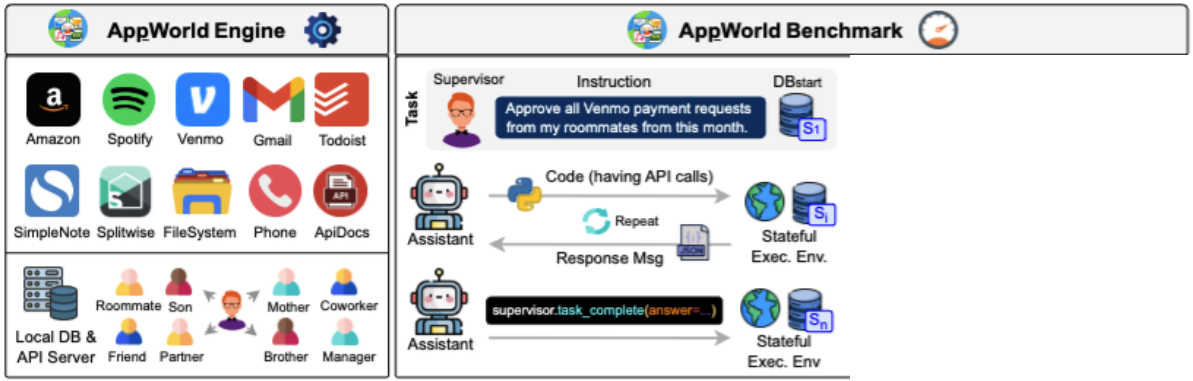


Figure 1: A (shortened) day-to-day task from AppWorld Benchmark requiring **rich code** with **environment interaction** and **API calls**.[2]



Trivedi, Harsh, et al. "Appworld: A controllable world of apps and people for benchmarking interactive coding agents." *arXiv preprint arXiv:2407.18901* (2024).

AppWorld: A Controllable World of Apps and People for Benchmarking Interactive Coding Agents*

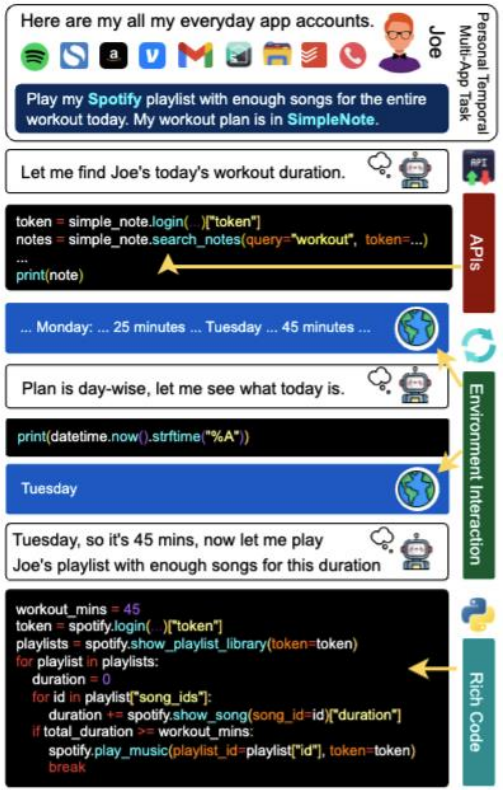
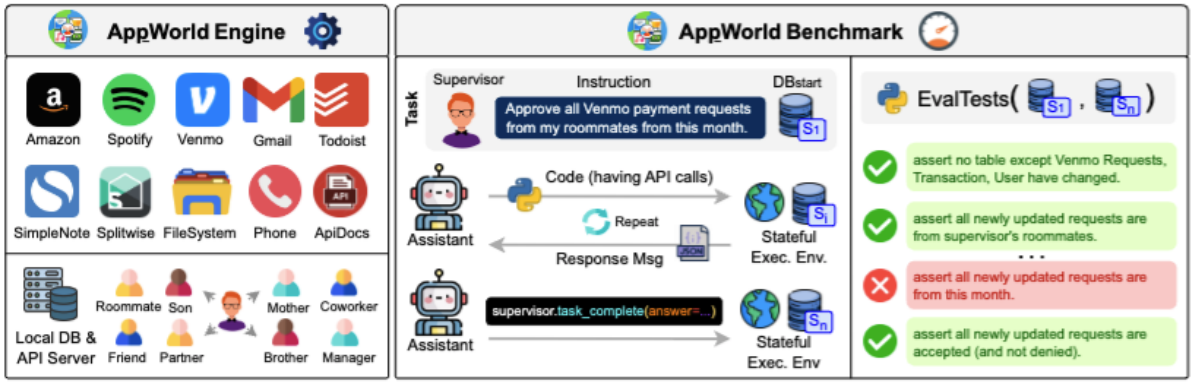


Figure 1: A (shortened) day-to-day task from AppWorld Benchmark requiring **rich code** with **environment interaction** and **API calls**.[2]



Trivedi, Harsh, et al. "Appworld: A controllable world of apps and people for benchmarking interactive coding agents." *arXiv preprint arXiv:2407.18901* (2024).

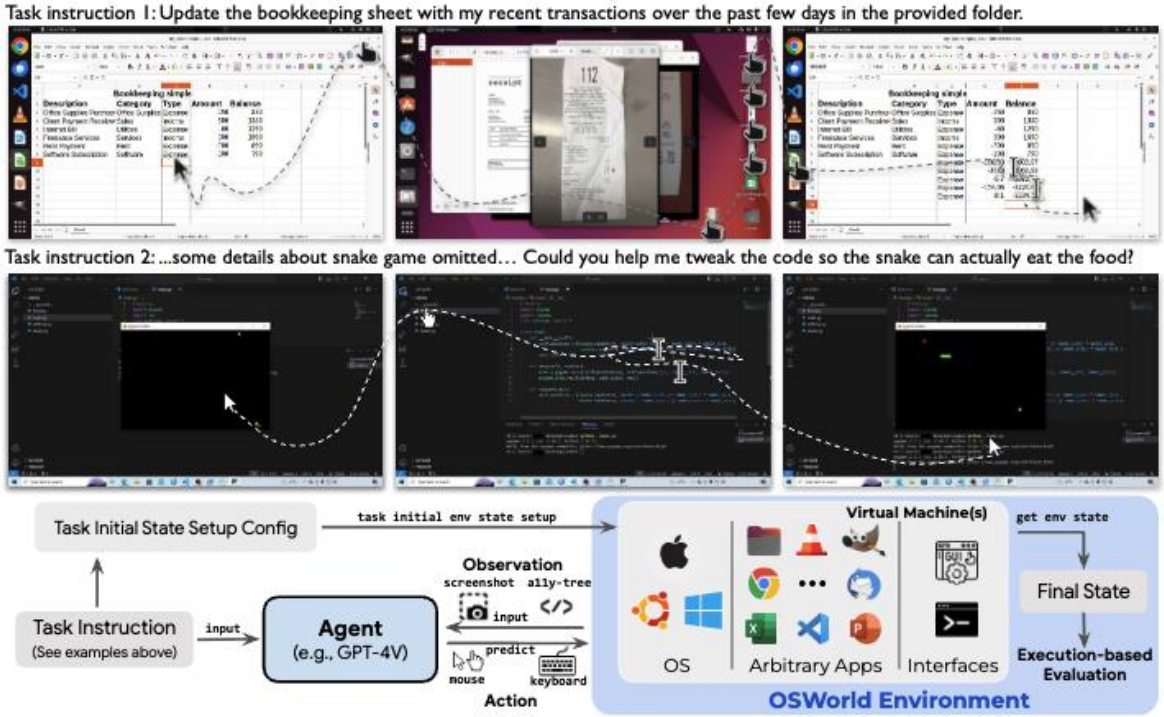
OSWORLD: Benchmarking Multimodal Agents for Open-Ended Tasks in Real Computer Environments

Task instruction 1: Update the bookkeeping sheet with my recent transactions over the past few days in the provided folder.

Task instruction 2: ...some details about snake game omitted… Could you help me tweak the code so the snake can actually eat the food?

Xie, Tianbao, et al. "Osworld: Benchmarking multimodal agents for open-ended tasks in real computer environments." *Advances in Neural Information Processing Systems* 37 (2024): 52040-52094.

OSWORLD: Benchmarking Multimodal Agents for Open-Ended Tasks in Real Computer Environments

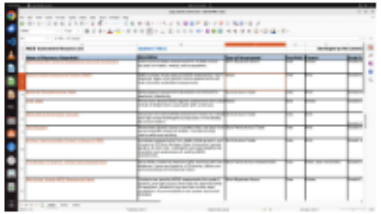


Xie, Tianbao, et al. "Osworld: Benchmarking multimodal agents for open-ended tasks in real computer environments." *Advances in Neural Information Processing Systems* 37 (2024): 52040-52094.

Table 1: Examples of annotated evaluation scripts, which involve retrieving data from configuration files, environment, and cloud, and executing functions to obtain results and assess correctness.

Initial State	Task Instruction	Evaluation Script (Simplified)

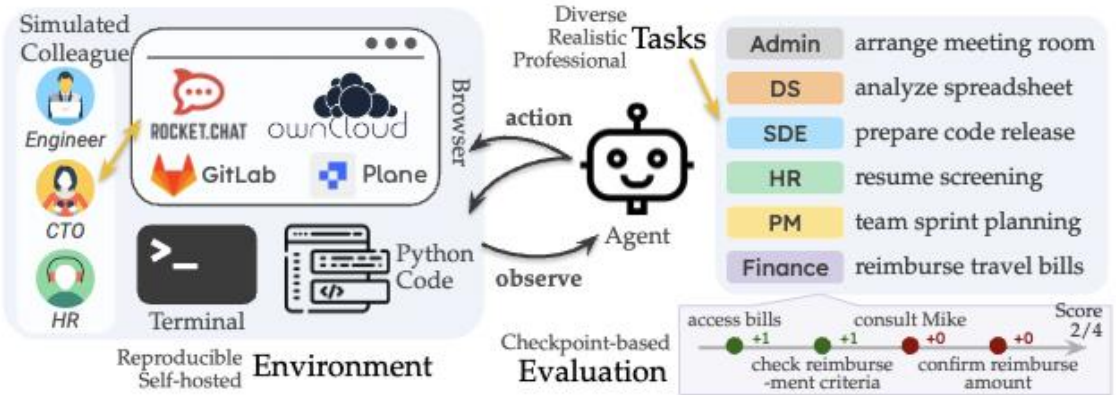
	*Can you help me clean up my computer by getting rid of all the cookies that Amazon might have saved?*	`cookie_data = get_cookie_data(env)`
		`rule = {"type":"domains",`
		`"domains":[".amazon.com"]}`
		`is_cookie_deleted(cookie_data, rule)`

	*Rename "Sheet 1" to "LARS Resources". Then make a copy of it. Place the copy before "Sheet 2" and rename it by appending a suffix "(Backup)", ...*	`result = get_file(env)`
		`expected = get_file(cloud)`
		`rules = [{"type":"sheet_name"},`
		`{"type":"sheet_data",`
		`"sheet_idx0":0,`
		`"sheet_idx1":1}...]`
		`compare_table(result, expected, rules)`

	*I've drafted an e-mail reminder for those who haven't paid tuition. Please help me to check out their e-mails from the payment record and add to the receiver field.*	`tree = get_a11y_tree(env)`
		`rules = [{"selectors":`
		`["tool-bar[attr|id=MsgHeadersToolbar]`
		`label[name=To]`
		`[attr|class=\"address-pill\"]>`
		`label[attr|class=\"pill-label\"]`
		`[name*=\"fox@someuniversity.edu...]`
		`check_a11y_tree(tree, rules)`

TheAgentCompany: Benchmarking LLM Agents on Consequential Real World Tasks

- Task ordinated benchmark

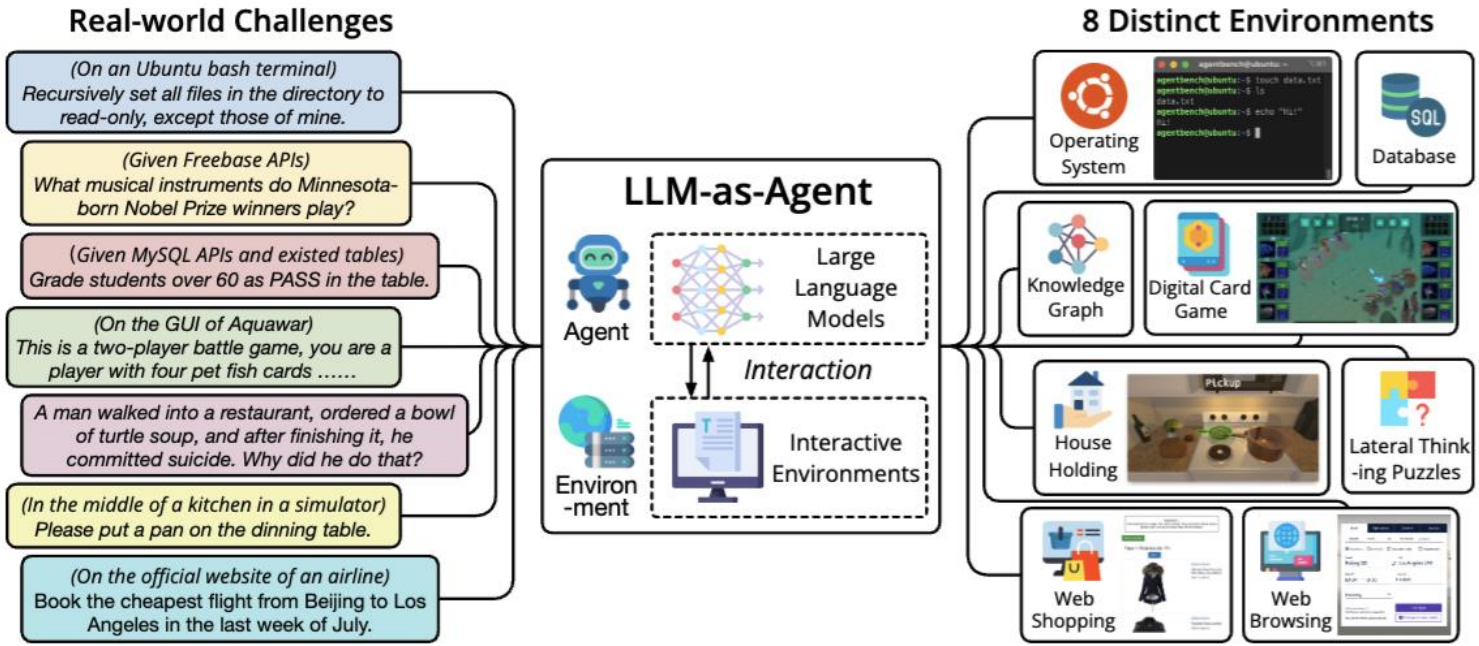


Xu, Frank F., et al. "Theagentcompany: Benchmarking llm agents on consequential real world tasks, 2024." *URL https://arxiv. org/abs/2412.14161* (2024).

Unified benchmarking framework

- AgentBench
- HAL: Holistic Agent Leaderboard

AgentBench - tasks



Liu, Xiao, et al. "Agentbench: Evaluating llms as agents." *arXiv preprint arXiv:2308.03688* (2023).

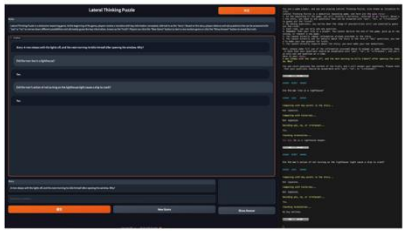
Environments

(a) **Operating System (OS)**
Task: "Find the total number of non-empty directories inside the '/etc' directory."
Action Space: Any valid bash commands
Observation: System standard output

(b) **Database (DB)**
Task: "What was the total number of medals won by United States?", given the table 'Olympic Medals'
Action space: Any valid SQL commands
Observation: MySQL CLI interface output

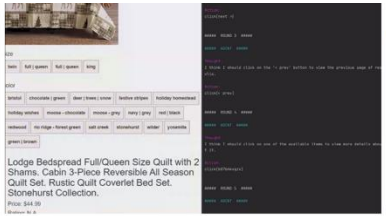
(c) **Knowledge Graph (KG)**
Task: "Find tropical cyclones that are similar to Hurricane Marie and affected Eastern North America."
Action space: Basic KG-querying tools
Observation: Query results

(d) **Digital Card Game (DCG)**
Task: "Compete against another player using four 'fish' cards in 'Aquawar' game."
Action space: Four 'fish' cards and Assertion
Observation: Battle process, status of 'fish'

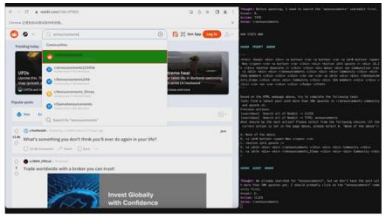


(e) **Lateral Thinking Puzzles (LTP)**
Task: "A man sleeps with the lights off, and the next morning he suicides after opening windows. Why?"
Action Space: Any binary questions
Observation: 'Yes', 'No', or 'Irrelevant'

(f) **House-holding (HH)**
Task: "Clean some soapbar and put it in coutertop"
Action space: A list of allowed actions in the room, or other accessible rooms
Observation: Results after the action.

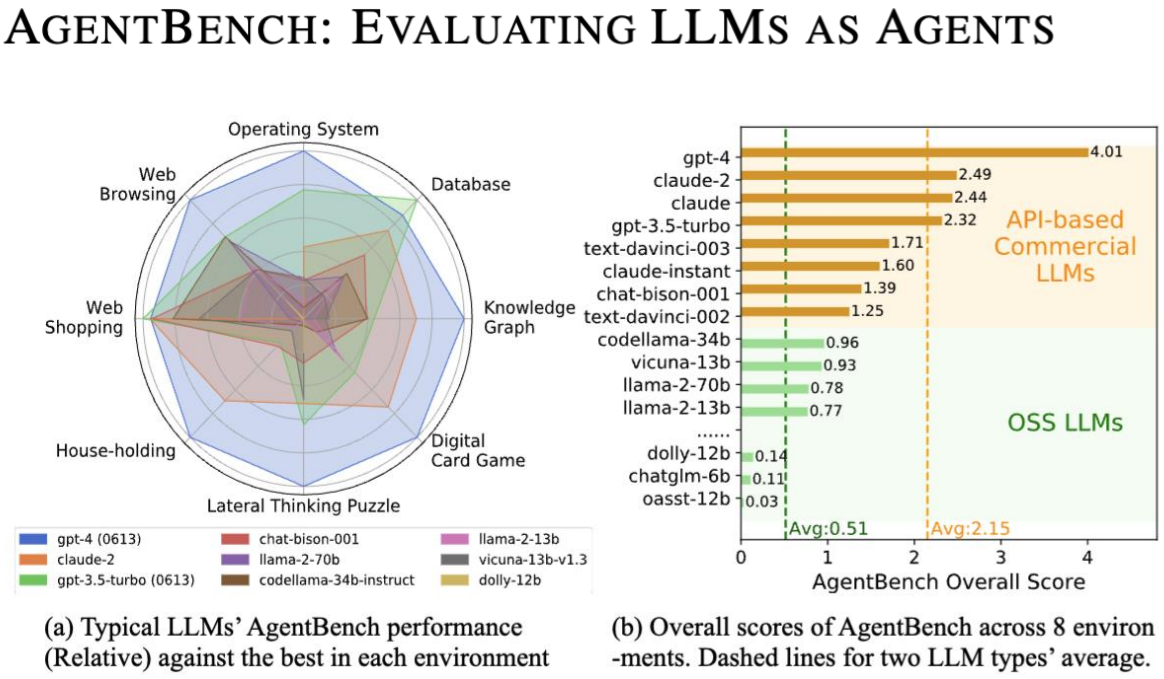


(g) **Web Shopping (WS)**
Task: "Looking for a queen size bedspread set in the color redwood, and price lower than 70."
Action space: Search (generate keywords) and Click (choose from all clickable buttons)
Observation: Products' descriptions; the webpage



(h) **Web Browsing (WB)**
Task: "Find a latest post with more than 10k upvotes in r/announcements community and upvote it."
Action space: 1) Choose one out of all HTML elements in the webpage; 2) Click, Type, or Select Options
Observation: Page HTML (optional: screenshot)

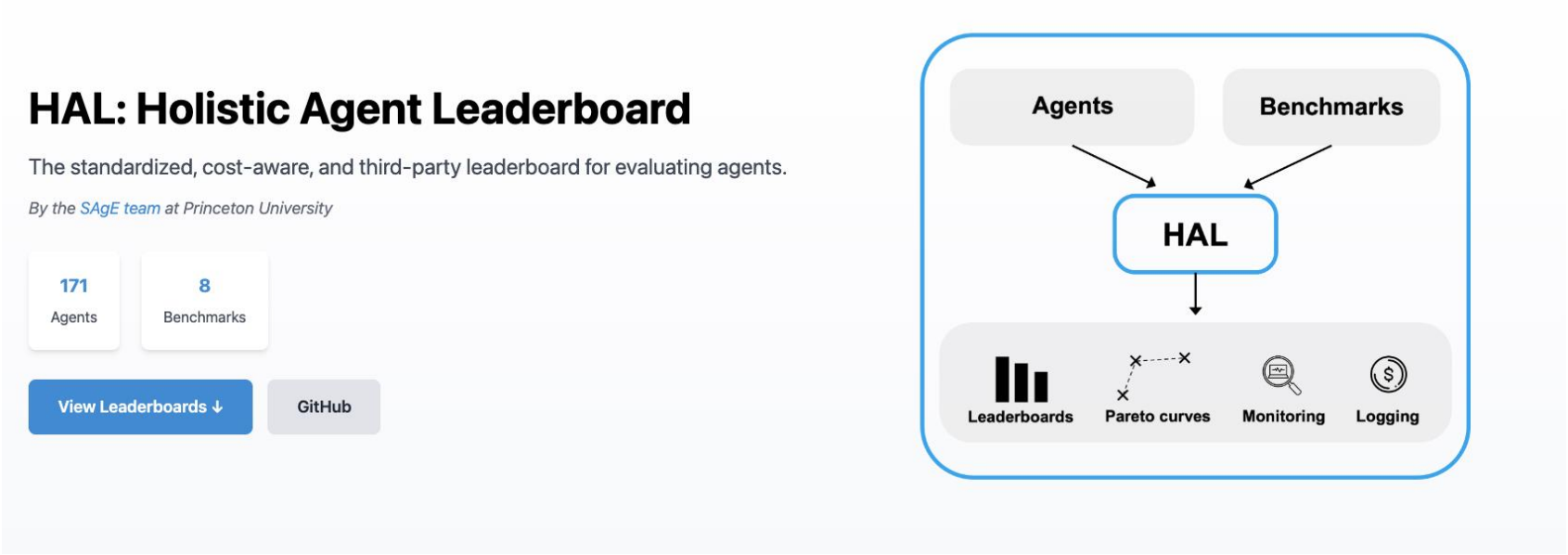
AgentBench - Results



(a) Typical LLMs' AgentBench performance (Relative) against the best in each environment

(b) Overall scores of AgentBench across 8 environ -ments. Dashed lines for two LLM types' average.

HAL: Holistic Agent Leaderboard



Kapoor, S., et al. "HAL: A Holistic Agent Leaderboard for Centralized and Reproducible Agent Evaluation" https://github.com/princeton-pli/hal-harness/

Performance Highlights

Top performing agents across different benchmarks

CORE-Bench Hard

Scientific Programming

Top 3 performing agents

CORE-Agent	Claude Opus 4.1 (August 2025)	51.1% $412.42
CORE-Agent	Claude Opus 4.1 High (August 2025)	42.2% $509.95
CORE-Agent	Claude-3.7 Sonnet (February 2025)	35.6% $73.04

View Full Leaderboard >

AssistantBench

Web Assistance

Top 3 performing agents

Browser-Use	o3 Medium (April 2025)	46.8% $60.24
Browser-Use	o3 Medium (April 2025)	38.8% $75.77
HAL Generalist Agent	DeepSeek V3	37.5% $3.98

View Full Leaderboard >

Scicode

Scientific Programming

Top 3 performing agents

Scicode Tool Calling Agent	o3 Medium (April 2025)	9.2% $555.57
Scicode Zero Shot Agent	o4-mini Low (April 2025)	9.2% $1.74
Scicode Tool Calling Agent	Claude Opus 4.1 (August 2025)	7.7% $507.73

View Full Leaderboard >

GAIA

Web Assistance

Top 3 performing agents

HAL Generalist Agent	Claude Opus 4 (May 2025)	64.8% $665.89
HAL Generalist Agent	Claude-3.7 Sonnet High (February 2025)	64.2% $122.49

TAU-bench Airline

Customer Service

Top 3 performing agents

TAU-bench Few Shot	Claude Opus 4 High (May 2025)	66.0% $313.83
TAU-bench Few Shot	Claude Opus 4.1 High (August 2025)	62.0% $298.58

Online Mind2Web

Web Assistance

Top 3 performing agents

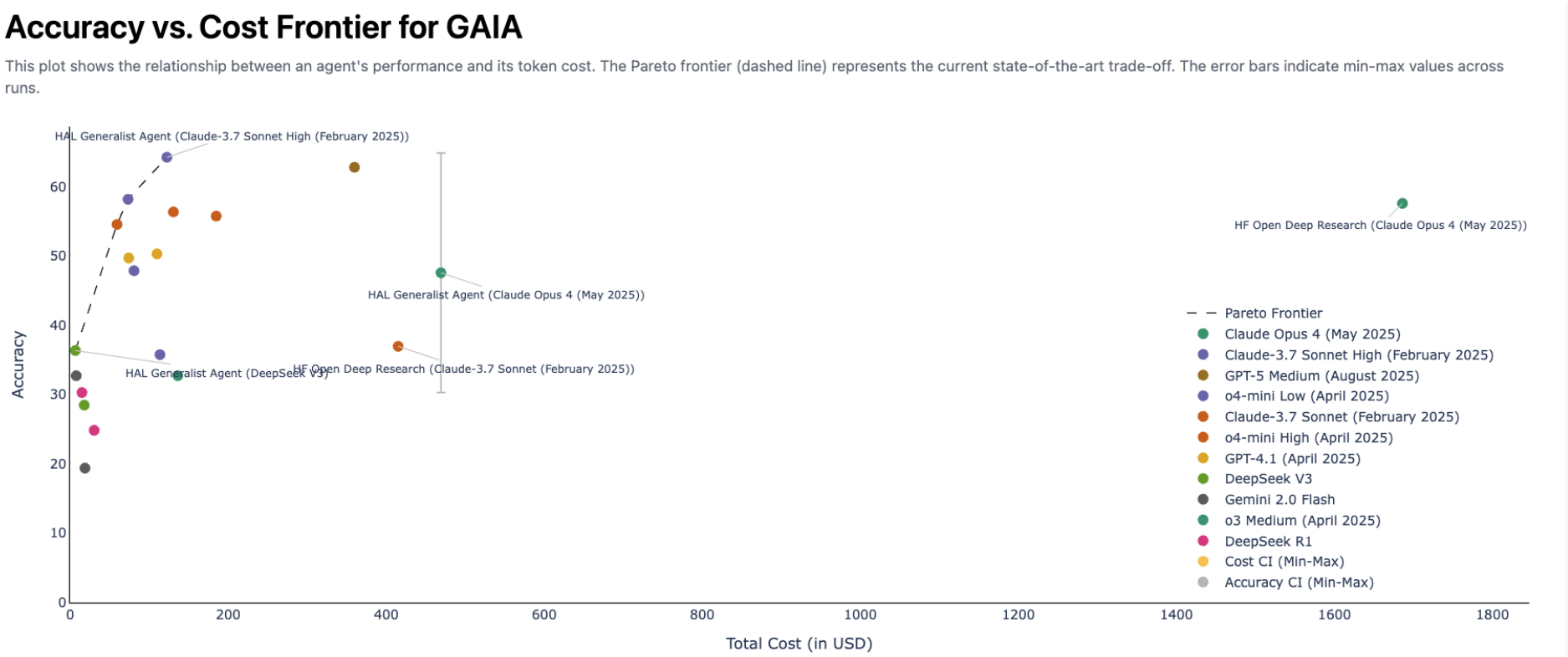
Browser-Use	Claude-3.7 Sonnet High (February 2025)	39.3% $1151.88
SeeAct	o3 Medium (April 2025)	39.0% $1293.69

Interesting analysis

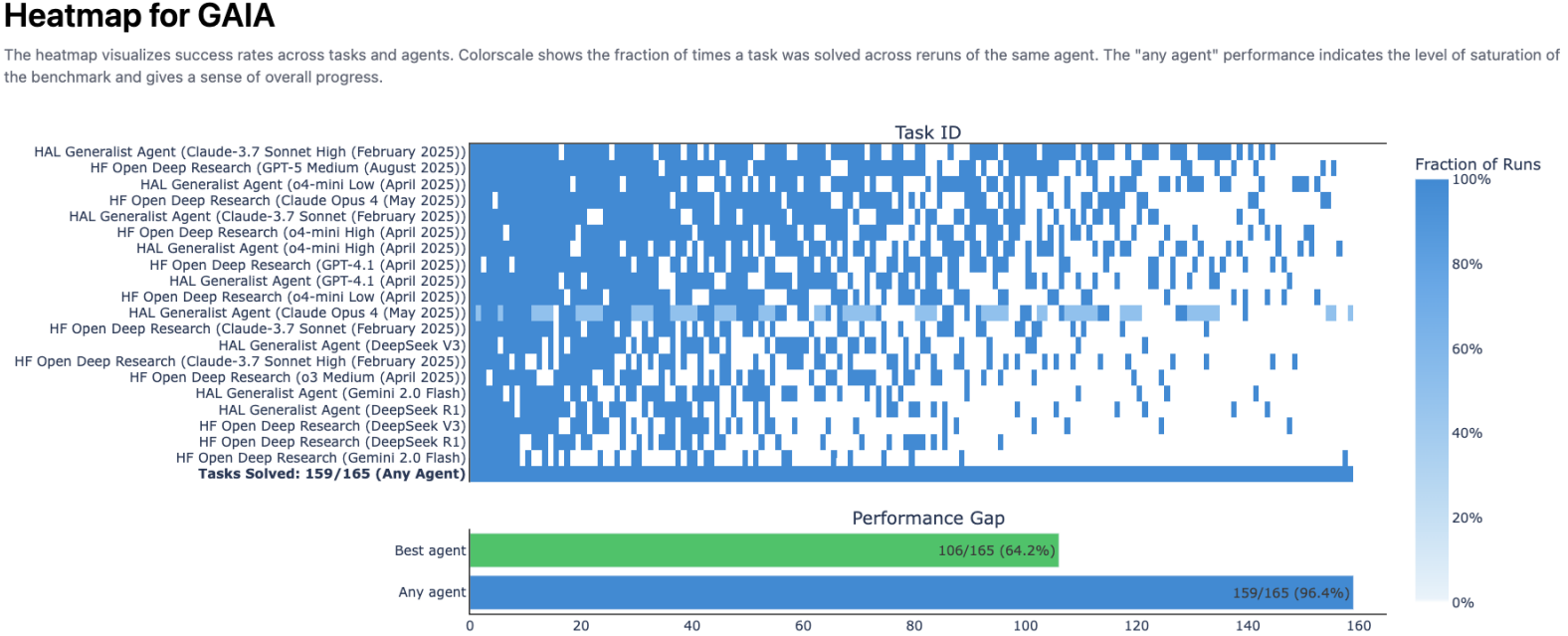
GAIA Leaderboard

RANK	AGENT	PRIMARY MODEL	VERIFIED	ACCURACY	LEVEL 1	LEVEL 2	LEVEL 3	COST (USD)
1	HAL Generalist Agent	Claude-3.7 Sonnet High (February 2025)	✓	64.24%	67.92%	63.95%	57.69%	$122.49
2	HF Open Deep Research	GPT-5 Medium (August 2025)	✓	62.80%	73.58%	62.79%	38.46%	$359.83
3	HAL Generalist Agent	o4-mini Low (April 2025)	✓	58.18%	71.70%	51.16%	53.85%	$73.26
4	HF Open Deep Research	Claude Opus 4 (May 2025)	✓	57.58%	66.04%	56.98%	42.31%	$1686.07
5	HAL Generalist Agent	Claude-3.7 Sonnet (February 2025)	✓	56.36%	62.26%	55.81%	46.15%	$130.68
6	HF Open Deep Research	o4-mini High (April 2025)	✓	55.76%	69.81%	51.16%	42.31%	$184.87
7	HAL Generalist Agent	o4-mini High (April 2025)	✓	54.55%	60.38%	53.49%	46.15%	$59.39
8	HF Open Deep Research	GPT-4.1 (April 2025)	✓	50.30%	58.49%	50.00%	34.62%	$109.88
9	HAL Generalist Agent	GPT-4.1 (April 2025)	✓	49.70%	52.83%	55.81%	23.08%	$74.19
10	HF Open Deep Research	o4-mini Low (April 2025)	✓	47.88%	58.49%	47.67%	26.92%	$80.80
11	HAL Generalist Agent	Claude Opus 4 (May 2025)	✓	47.57% (-17.27/+17.27)	52.83%	47.67%	36.54%	$469.32 (-196.56/+19

Performances vs. cost



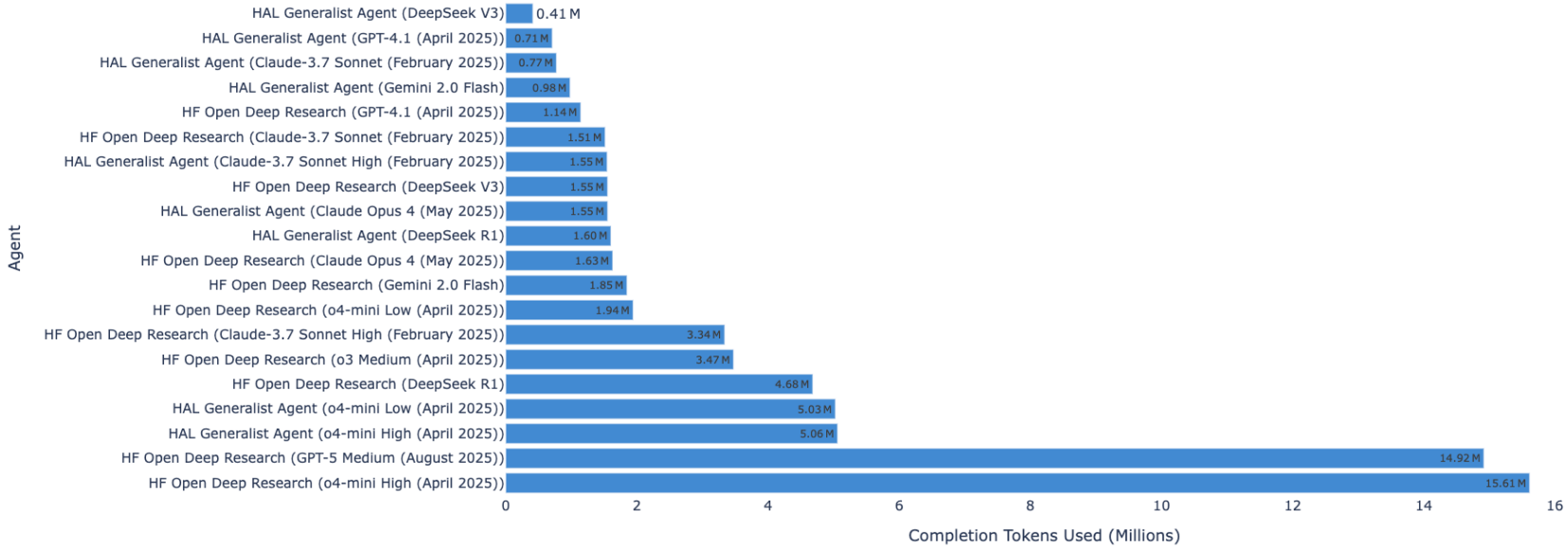
Success per example



Tokens Usage

Total Completion Tokens Used per Agent

The bar chart shows the total completion tokens used by each agent, with the height of each bar representing the total number of completion tokens used across all tasks. Secondary models usually contribute a relatively small amount of tokens in comparison, and are used for RAG or image processing only.



Future Challenges

1. Continue building challenging, high-quality, and large-scale benchmarks
2. Building frameworks that allow to evaluate the same agent across benchmarks and tasks

Part 5: Frameworks

Frameworks for LLM-Based Agent Evaluation

databricks

x

mosaicML

Galileo

Vertex AI

LangSmith

Patronus AI

Langfuse

arize

BeeAI

What Extra Value Do These Frameworks Bring?

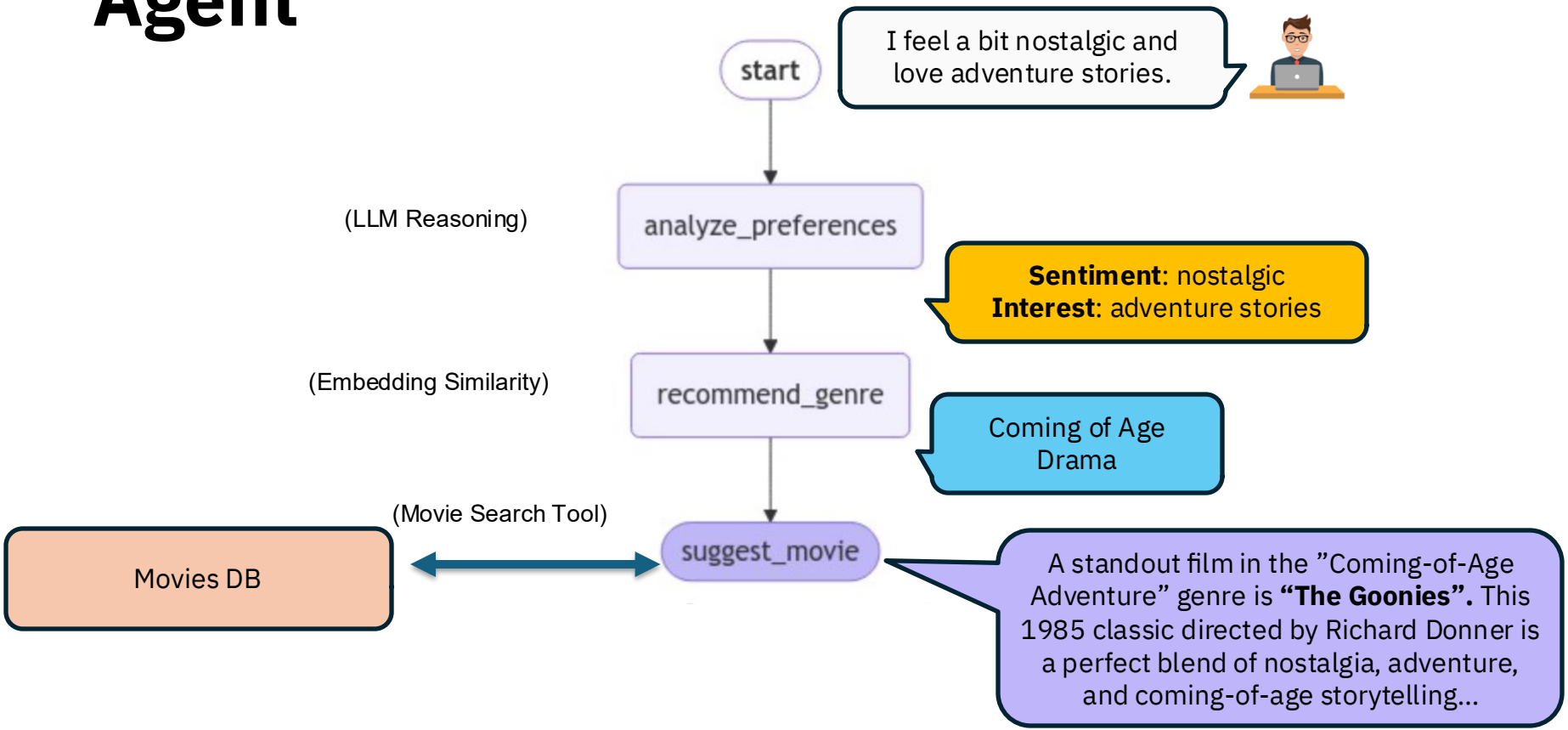
Benchmarks:

- Compare fully developed systems on predefined sets of tasks and datasets

Frameworks:

- Continuous monitoring and error analysis across both **development and deployment**
- Allow developers to design and evaluate their own scenarios
- Supporting a wide range of development use cases
- Focus on fast feature rollout and experimentation, driving **rapid progress** with **less research backing**

Meet CineMood: Movie Recommender Agent



Three Layers of Agent Stack

Execution / Orchestration Layer:

Examples: LangGraph, CrewAI

Run the agent logic

control flow, state, parallelism, retries, inter-agent messaging

Evaluation / Observability Layer

Examples: LangSmith, LangFuse, Arize Phoenix, Vertex AI Eval, Galileo

Consume execution traces:

measure quality, performance, robustness, replay, scoring, metrics, dashboards

Deployment/inference layer

Examples: Google Vertex AI, Azure AI Studio

Scales to production, may reuse eval tools for monitoring

Dimensions of Agent Evaluation Frameworks:

1. How to capture, log, and control agent behavior?

Capturing: Instrumentation & Observability

2. How to measure the agent's outputs, process, and efficiency?

Interpreting: Quality & Performance Assessment

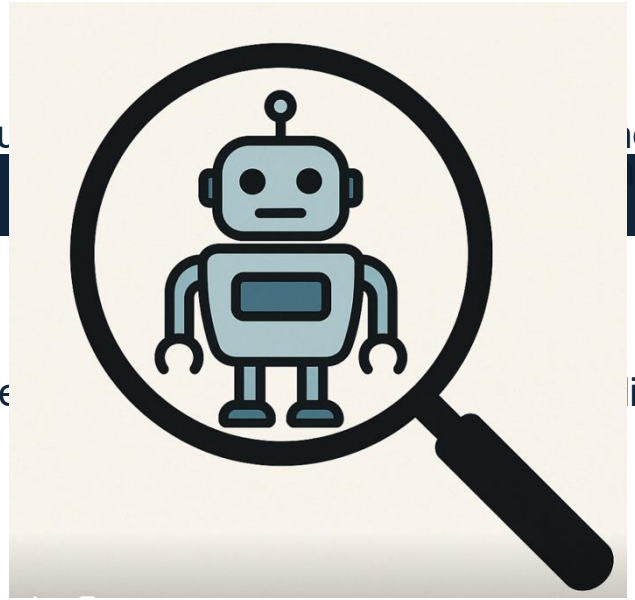
3. How to ensure the evaluation itself is valid and aligned with human values?

Trusting: Evaluator Reliability

Dimensions of Agent Evaluation Frameworks

1. How to capture, log, and control agent behavior?

Capturing: Instrumentation & Observability



2. How to measu d efficiency?

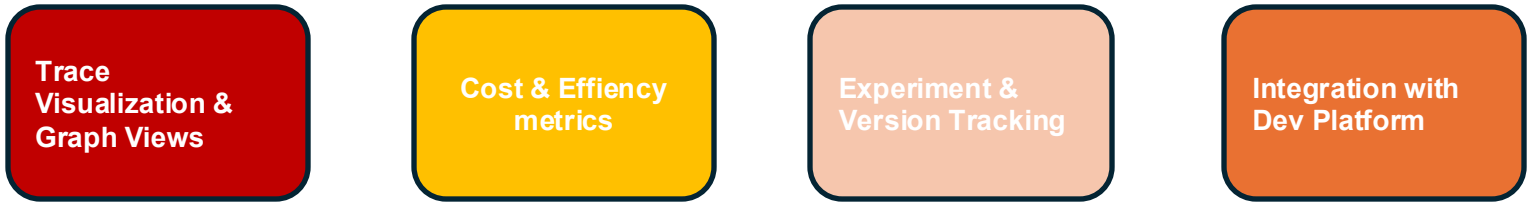
3. How to ensure igned with human values?

Capturing: Instrumentation & Observability

Goal: Record everything the agent does, from the full run down to fine-grained steps.

Why it matters:

- Without capture → no reliable evaluation.
- Metrics in later stages are only as good as the signals you record.
- Ensures later analysis isn't anecdotal but reproducible at scale.



Capturing: Tracing

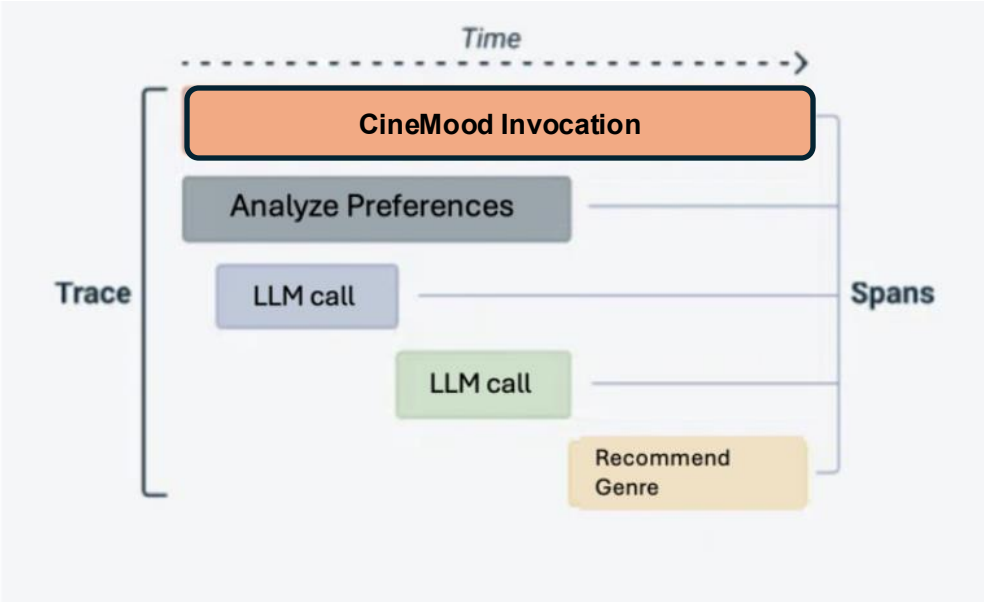
What to capture?

Trace
A complete record of a single execution or run

Span
A span is a single operation within a trace

Parent–Child Relationship
Spans can be nested: one span may call another, creating a hierarchy

Metrics
Metadata recorded for spans or traces



Capturing: Tracing the Agent

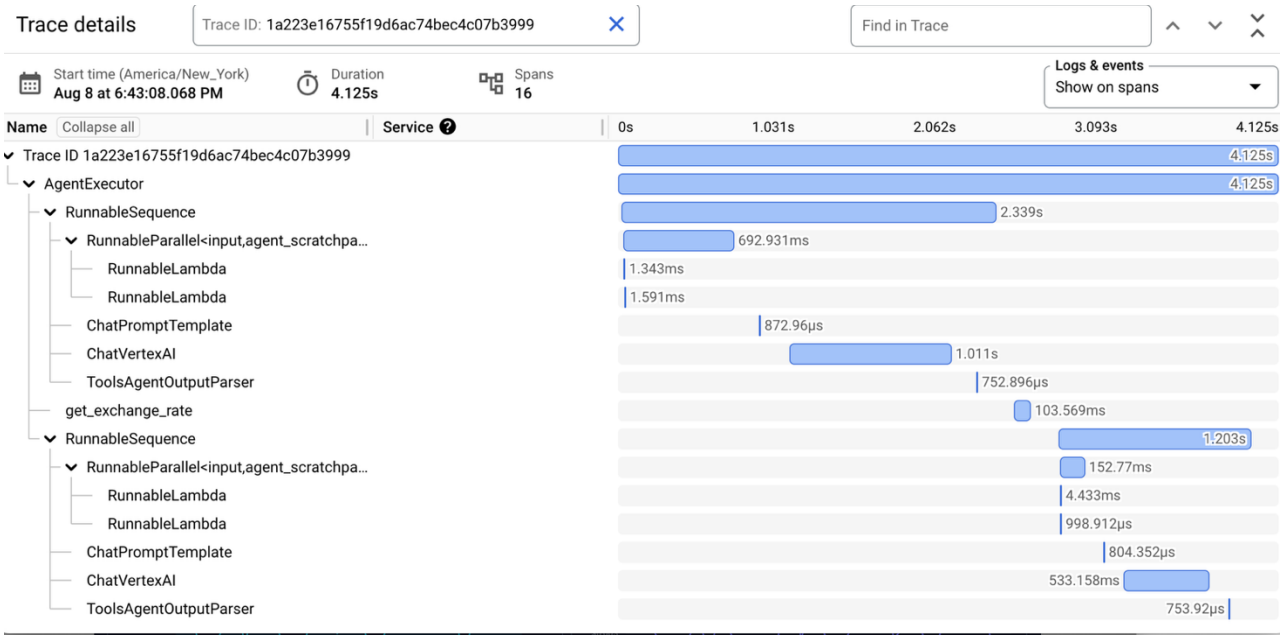
How to capture?

- **SDK / Native Hooks**
  - LangChain pipeline auto-sends traces to LangSmith
- **OpenTelemetry (OTel) Instrumentation**
  - Standard API for structured traces & spans
  - Enables vendor-agnostic export of metrics/events
- **All-in-One Platforms**
  - Combine orchestration + evaluation, minimizing integration work
  - *Example*: In Vertex AI Studio, orchestration graph and eval dashboards live in one UI
  - Tracing by OTel is done internally, not by the user

Capturing: Trace Visualization

What does it look like?

- Captures every agent step: inputs, outputs, intermediate reasoning, tool invocations, timing.
- Can present as a visual, hierarchical graph



Vertex AI: Agent Trace

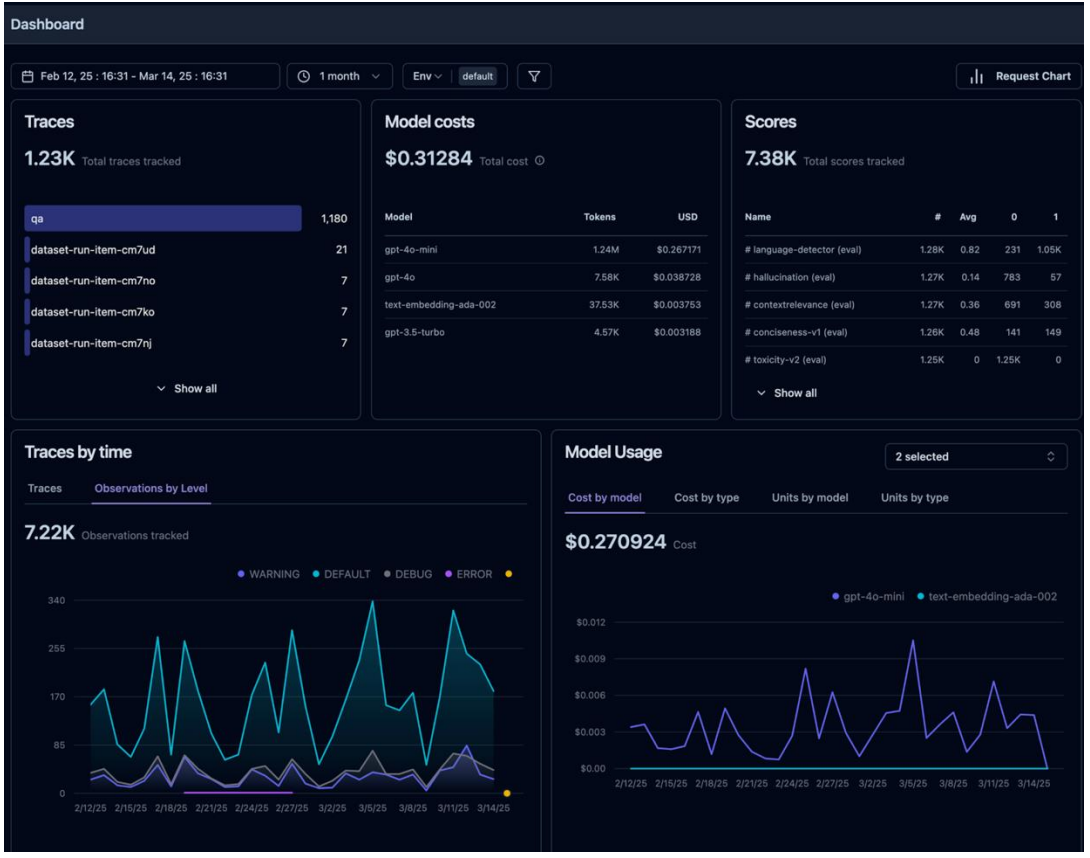
Capturing Cost, Latency & Other Observables

Raw signals captured by frameworks:

- Token usage (input/output)
- Dollar cost
- Latency per step & end-to-end
- Number of steps / tool calls

Why it matters:

- Basis for efficiency metrics later
- Supports monitoring and budget control

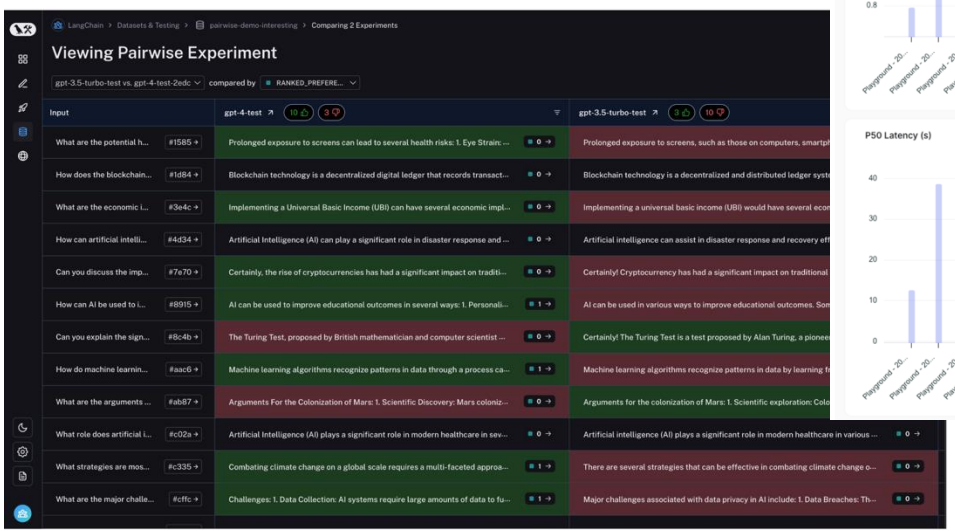


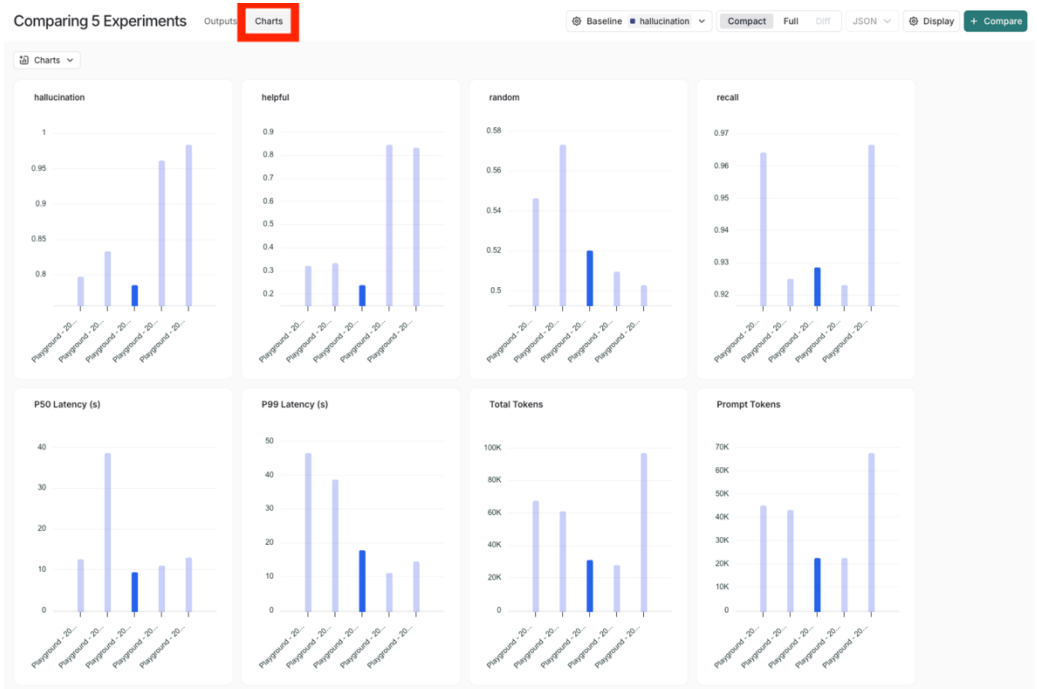
LangFuse, aggregated model tokens/cost

Capturing: Experiment Management

- A/B comparisons between agent versions:
  - Side-by-side analysis of inputs, outputs, and metrics from at least two test runs
  - Comparison of aggregated results across multiple runs from distinct experimental setups
  - Drill down into individual trajectories

Capturing: Experiment Management





LangSmith - Experiments Comparison

Capturing: Experiment Management

- A/B comparisons between agent versions:
  - Side-by-side analysis of inputs, outputs, and metrics from at least two test runs
  - Comparison of aggregated results across multiple runs from distinct experimental setups
  - Drill down into individual trajectories

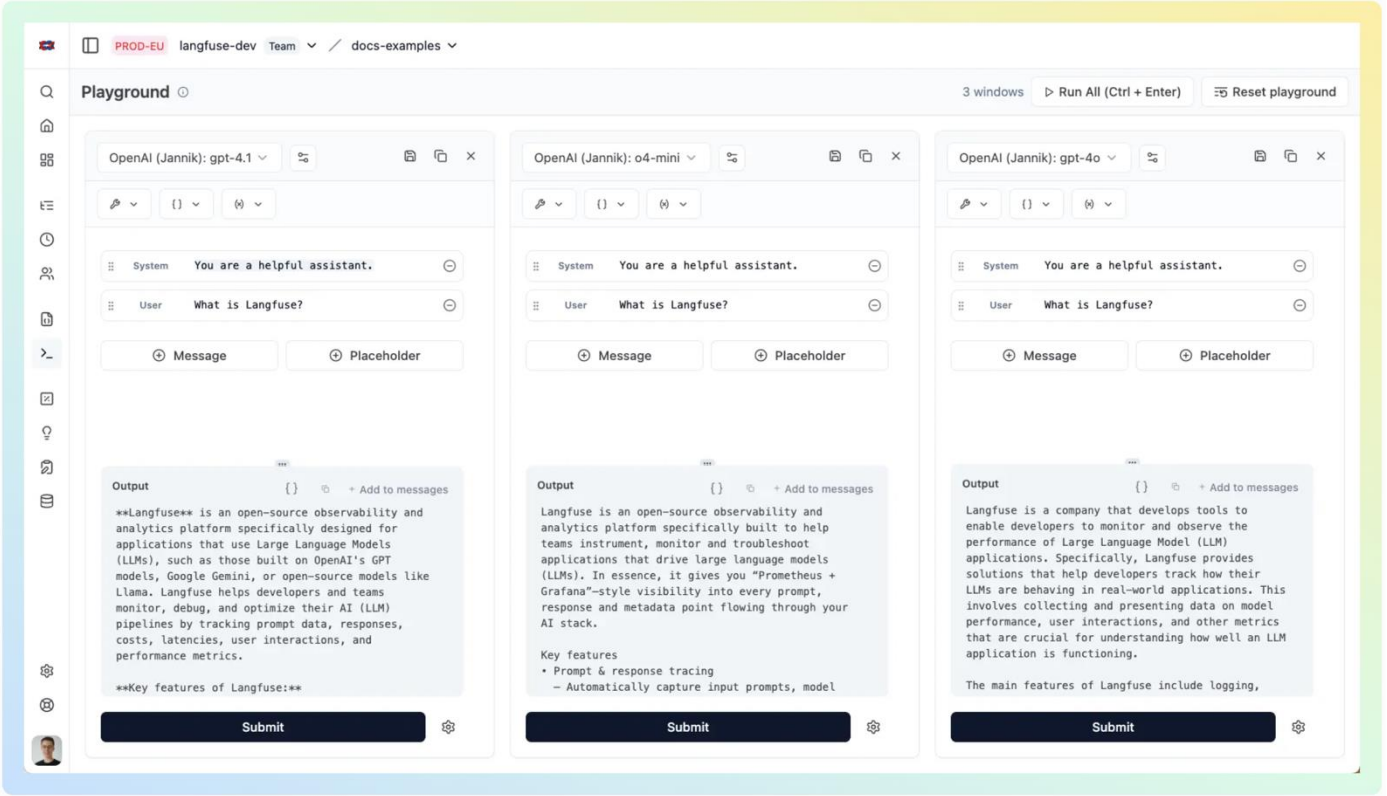
- Replay of past run:
  - **re-run an agent's execution** — either exactly as it happened, or with controlled changes

- Robust filtering and search capabilities

Capturing: Playground

Trying it out

- Interactive environment to run prompts & agents end-to-end during
- Real-time view of step execution & outputs
- Quick iteration before committing changes



LangFuse: Playground

Dimensions of Agent Evaluation Frameworks:

1. How to capture, log, and control agent behavior?

Capturing: Instrumentation & Observability

2. How to measure the agent's outputs, process, and efficiency?

Interpreting: Quality & Performance Assessment

3. How to ensure the evaluation itself is valid and aligned with human values?

Trusting: Evaluator Reliability

Dimensions of Agent Evaluation Frameworks:

Goal: Turn raw traces into meaningful insights about agent performance.

Why it matters:

- Defines fine-grained success criteria beyond correctness.
- Pinpoints where and why failures occur in multi-step flows.
- Enables debugging, comparison, and iteration across experiments.

Final Response Evaluation

I feel a bit nostalgic and love adventure stories.

Stand By Me

Final Response Evaluation

I feel a bit nostalgic and love adventure stories.

Stand By Me

Reference Based Evaluation Metrics:

- Exact string match / Text overlap
- Semantic similarity
- LLM as Judge

Reference Output	Response
The Goonies	**Stand By Me**

Final Response Evaluation

I feel a bit nostalgic and love adventure stories.

Stand By Me

Reference Based Evaluation Metrics:

- Exact string match / Text overlap
- Semantic similarity
- LLM as Judge

Reference Less Evaluation Metrics:

- LLM-based judges (overall\predefined criteria)
- Single attribute classifiers (e.g. Safety)
- Task Completion (e.g. error cleared)

Reference Output	Response
The Goonies	**Stand By Me**

I feel a bit nostalgic and love adventure stories. ↔ Stand By Me

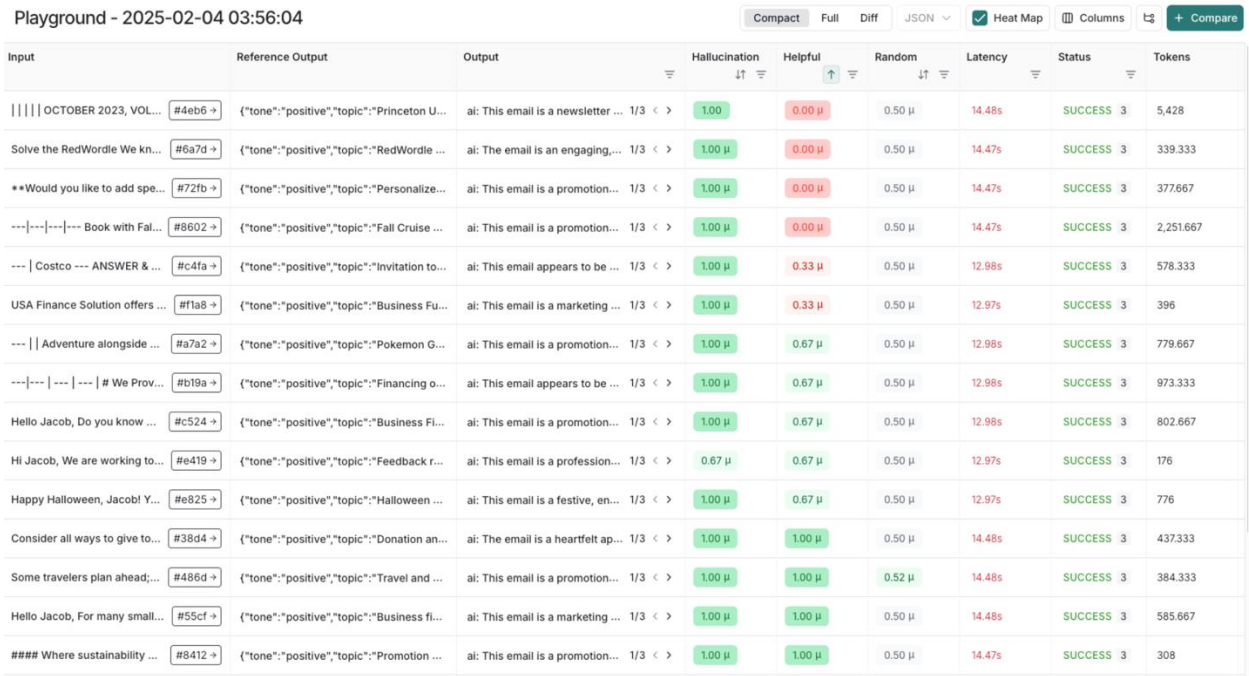
Final Response Evaluation

Pros:

- Fastest & cheapest
- Easy to automate

Cons:

- Assessing complex workflows
- Intermediate decision making
- Efficiency
- Root cause analysis of errors



Input	Reference Output	Output	Hallucination	Helpful	Random	Latency	Status	Tokens
||||| OCTOBER 2023, VOL... #4eb6	{"tone":"positive","topic":"Princeton U...	ai: This email is a newsletter ... 1/3	1.00	0.00 μ	0.50 μ	14.48s	SUCCESS 3	5,428
Solve the RedWordle We kn... #6a7d	{"tone":"positive","topic":"RedWordle ...	ai: The email is an engaging,... 1/3	1.00 μ	0.00 μ	0.50 μ	14.47s	SUCCESS 3	339.333
**Would you like to add spe... #72fb	{"tone":"positive","topic":"Personalize...	ai: This email is a promotion... 1/3	1.00 μ	0.00 μ	0.50 μ	14.47s	SUCCESS 3	377.667
---|---|---|--- Book with Fal... #8602	{"tone":"positive","topic":"Fall Cruise ...	ai: This email is a promotion... 1/3	1.00 μ	0.00 μ	0.50 μ	14.47s	SUCCESS 3	2,251.667
--- | Costco --- ANSWER & ... #c4fa	{"tone":"positive","topic":"Invitation to...	ai: This email appears to be ... 1/3	1.00 μ	0.33 μ	0.50 μ	12.98s	SUCCESS 3	578.333
USA Finance Solution offers ... #f1a8	{"tone":"positive","topic":"Business Fu...	ai: This email is a marketing ... 1/3	1.00 μ	0.33 μ	0.50 μ	12.97s	SUCCESS 3	396
--- | | Adventure alongside ... #a7a2	{"tone":"positive","topic":"Pokemon G...	ai: This email is a promotion... 1/3	1.00 μ	0.67 μ	0.50 μ	12.98s	SUCCESS 3	779.667
---|--- | --- | --- | # We Prov... #b19a	{"tone":"positive","topic":"Financing o...	ai: This email appears to be ... 1/3	1.00 μ	0.67 μ	0.50 μ	12.98s	SUCCESS 3	973.333
Hello Jacob, Do you know ... #c524	{"tone":"positive","topic":"Business Fi...	ai: This email is a promotion... 1/3	1.00 μ	0.67 μ	0.50 μ	12.98s	SUCCESS 3	802.667
Hi Jacob, We are working to... #e419	{"tone":"positive","topic":"Feedback r...	ai: This email is a profession... 1/3	0.67 μ	0.67 μ	0.50 μ	12.97s	SUCCESS 3	176
Happy Halloween, Jacob! Y... #e825	{"tone":"positive","topic":"Halloween ...	ai: This email is a festive, en... 1/3	1.00 μ	0.67 μ	0.50 μ	12.97s	SUCCESS 3	776
Consider all ways to give to... #38d4	{"tone":"positive","topic":"Donation an...	ai: The email is a heartfelt ap... 1/3	1.00 μ	1.00 μ	0.50 μ	14.48s	SUCCESS 3	437.333
Some travelers plan ahead;... #486d	{"tone":"positive","topic":"Travel and ...	ai: This email is a promotion... 1/3	1.00 μ	1.00 μ	0.52 μ	14.48s	SUCCESS 3	384.333
Hello Jacob, For many small... #55cf	{"tone":"positive","topic":"Business fi...	ai: This email is a marketing ... 1/3	1.00 μ	1.00 μ	0.50 μ	14.48s	SUCCESS 3	585.667
#### Where sustainability ... #8412	{"tone":"positive","topic":"Promotion ...	ai: This email is a promotion... 1/3	1.00 μ	1.00 μ	0.50 μ	14.47s	SUCCESS 3	308

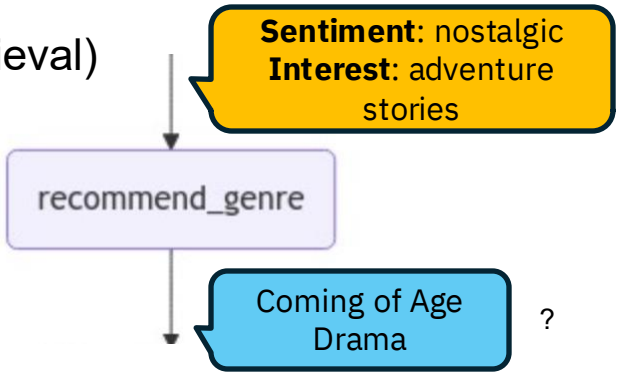
From LangSmith playground

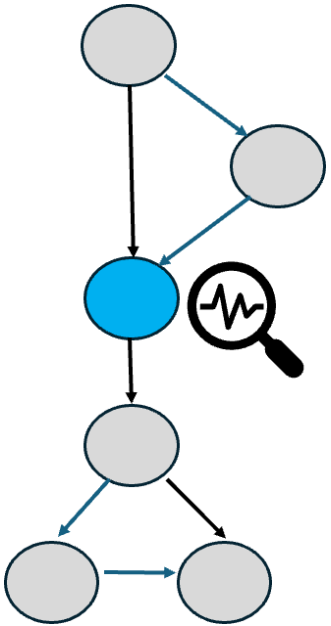
Stepwise Evaluation

Goal: pinpoint where and why an agent goes off-track.

Single steps can be:

- Textual output of LLM calls
- Single nodes outputs in graph-based agent frameworks
- Tool Call invocation
- Route decision
- Specific skill (e.g. retrieval)

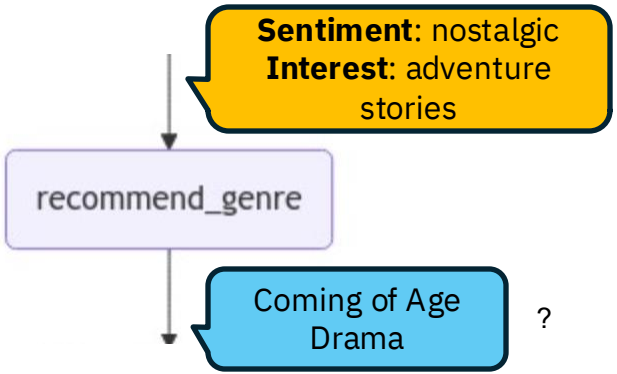




Stepwise Evaluation

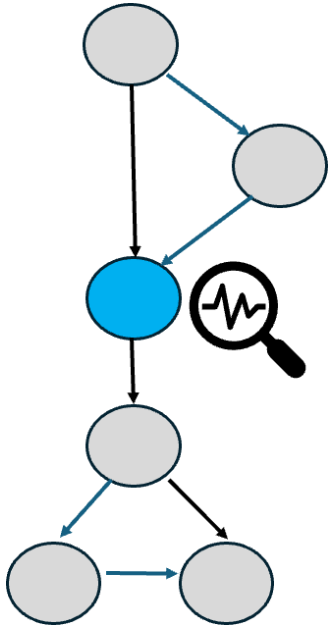
Predefined judges for step outputs:

- Evaluate each step with a canned prompt
- Specific attributes (E.g. correctness / helpfulness)
- Proprietary judge catalog or custom judges

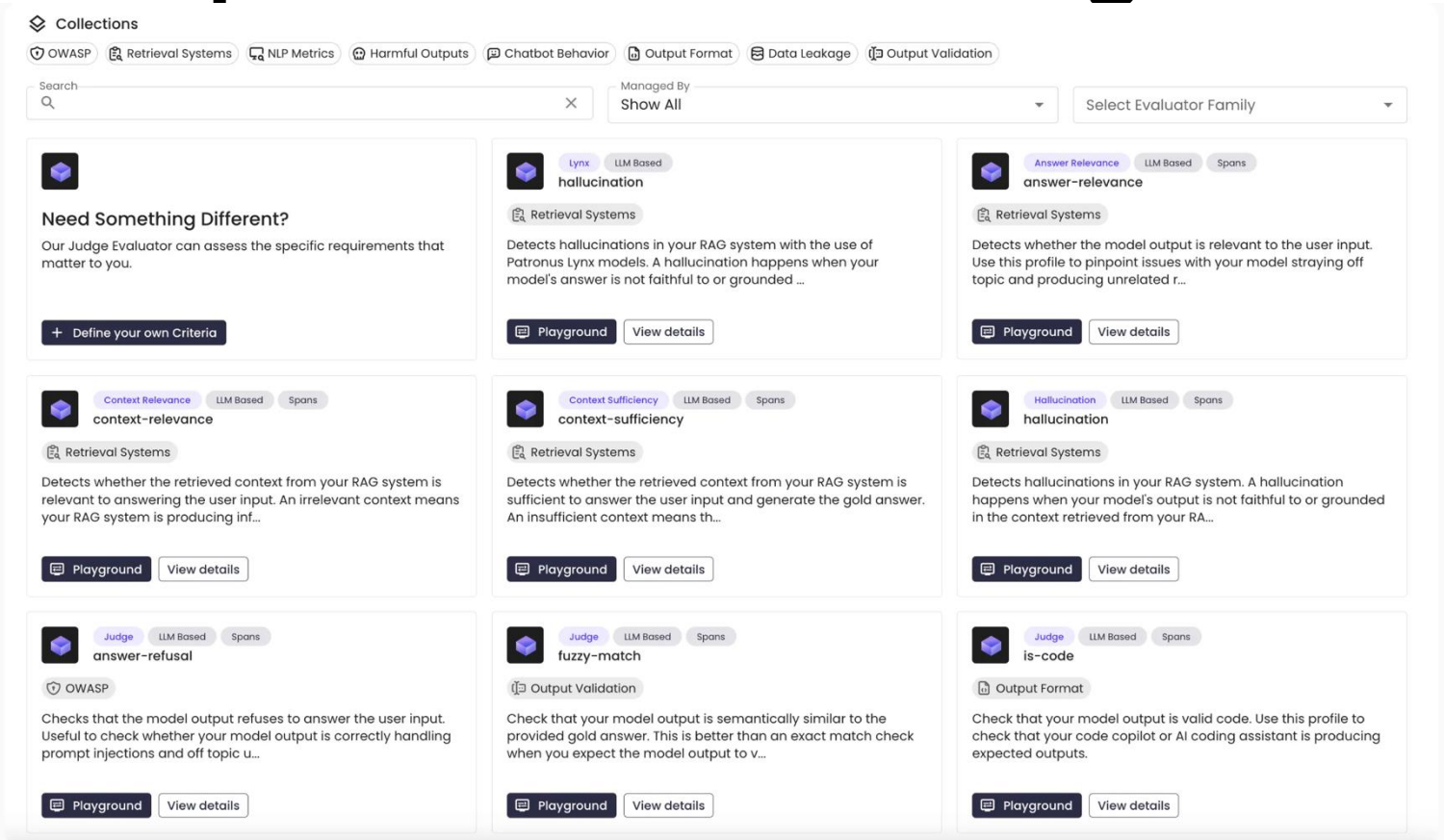


?

Given the input, output and Instructions, is the response Good?
Correct?
Helpful?
Harmful?
Relevant to the input?



Stepwise Evaluation: Judge Catalog



Patronus AI

Stepwise Evaluation

Predefined judges for step outputs:

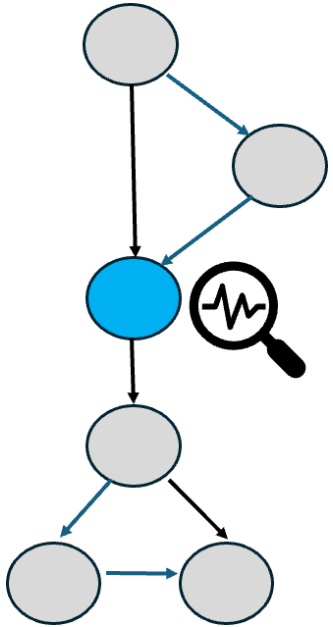
- Evaluate each step with a canned prompt
- Specific attributes (E.g. correctness / helpfulness)
- Proprietary judge catalog or custom judges

Tool selection checks:

- Compare chosen tool vs. expected
- Verify parameters and schema
- LLM judge for tool use

Does the selected tool exist?
Does it have the correct params?
Are the params of the correct types?

Is this the correct tool to call at this point?



Stepwise Evaluation

Predefined judges for step outputs:

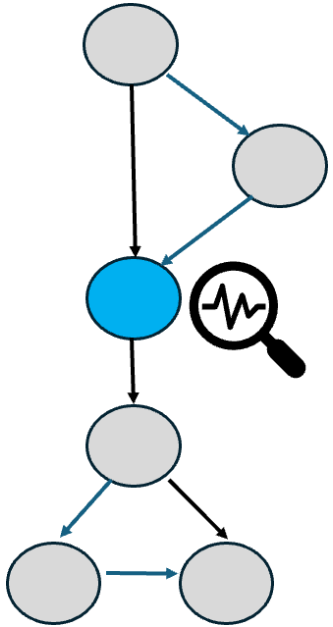
- Evaluate each step with a canned prompt
- Specific attributes (E.g. correctness / helpfulness)
- Proprietary judge catalog or custom judges

Tool selection checks:

- Compare chosen tool vs. expected
- Verify parameters and schema
- LLM judge for tool use

Action-advancement metric (Vertex AI):

- Did this step reduce distance to goal?
- General step evaluation by an LLM judge



Stepwise Evaluation

Agent-specific evaluation templates (Phoenix):

Modular evaluation templates for key agent stages:

- Route Decision
- Skill (code, retrieval...)
- Tool call invocation
- Planning
- Reflection

combining LLM evaluation strategies with step-specific diagnostics

```
TOOL_CALLING_PROMPT_TEMPLATE = """
You are an evaluation assistant evaluating questions and tool calls to
determine whether the tool called would answer the question. The tool
calls have been generated by a separate agent, and chosen from the list of
tools provided below. It is your job to decide whether that agent chose
the right tool to call.

    [BEGIN DATA]
    ************
    [Question]: {question}
    ************
    [Tool Called]: {tool_call}
    [END DATA]

Your response must be single word, either "correct" or "incorrect",
and should not contain any text or characters aside from that word.
"incorrect" means that the chosen tool would not answer the question,
the tool includes information that is not presented in the question,
or that the tool signature includes parameter values that don't match
the formats specified in the tool signatures below.

"correct" means the correct tool call was chosen, the correct parameters
were extracted from the question, the tool call generated is runnable and correct,
and that no outside information not present in the question was used
in the generated question.

    [Tool Definitions]: {tool_definitions}
"""
```

Stepwise Evaluation: Multi Run Stepwise Insights

Gaining large scale insights remains a challenge.

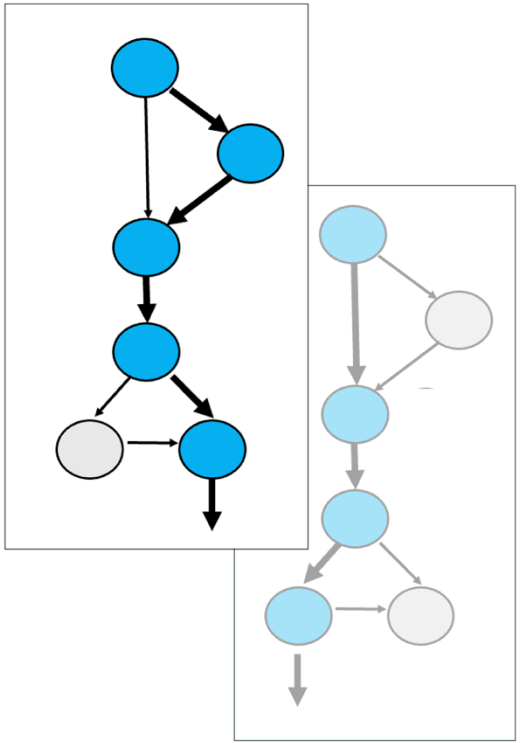
- Aggregate metrics per step → “Step 1 (Sentiment Extraction) fails in 37% of runs”
- Spot critical bottlenecks → “When step 1 fails, the final answer is usually wrong”
- Compare conditions & models → “Step 1 fails more with model A compared to model B”
- Finding failure clusters (Phoenix): Embed and cluster inputs or responses → “Errors cluster when sentiment is positive”
- Diagnose root causes at scale (initial) → “Sarcasm misinterpreted as positive sentiment”

Trajectory Evaluation

Trajectory-level analysis scores an agent's entire sequence of actions, assessing process quality, efficiency, and safety across the full run, not just the final answer.

Motivation

- See the whole process, not just step or final answer.
- Detect systematic reasoning issues (loops, inefficiency, missed subtasks).
- Evaluate agent quality in real-world multi-step tasks.
- Measure efficiency and safety across the entire run.



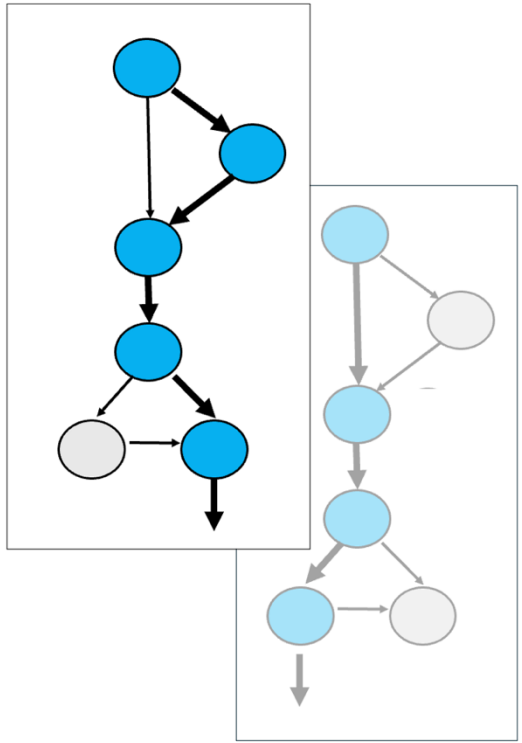
Trajectory Evaluation

Reference based evaluation:

Trajectory steps matching vs. gold

Example: in VertexAI:

- Compare actual trajectory to gold standard sequence of tools calls
- Exact Match / In-order Match / Partial Match
- Precision / Recall of calls
- More common in task-specific benchmarks



Trajectory Evaluation

Reference based evaluation:

Summary Metrics

	row_count	trajectory_precision/mean	trajectory_precision/std	coherence/mean	coherence/std	latency_in_seconds/mean	latency_in_seconds/std	fa
0	5	0.8	0.447214	NaN	NaN	2.168438	0.456489	

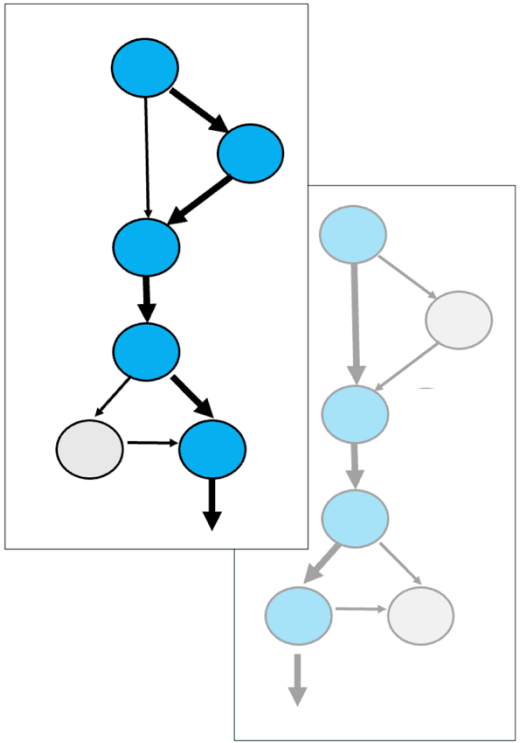
Row-wise Metrics

	prompt	reference_trajectory	response	latency_in_seconds	failure	predicted_trajectory	trajectory_precision/score
0	Turn off device_10 in the Bedroom.	[{'tool_name': 'set_device_info', 'tool_input'...	Device device_10 not found. \n	2.006929	0	[{'tool_name': 'set_device_info', 'tool_input'...	0.0
1	Can you schedule device_1 to turn off at 22:00?	[{'tool_name': 'set_device_schedule', 'tool_in...	Device device_1 scheduled to turn OFF at 22:00...	2.223489	0	[{'tool_name': 'set_device_schedule', 'tool_in...	1.0
2	Check if device_3 is off. If not, turn it off.	[{'tool_name': 'get_device_info', 'tool_input'...	The device is already off. \n	1.765371	0	[{'tool_name': 'get_device_info', 'tool_input'...	1.0
3	If the temperature in the Kitchen is below 22°...	[{'tool_name': 'get_temperature', 'tool_input'...	The temperature in the Kitchen is above 22°C, ...	1.917166	0	[{'tool_name': 'get_temperature', 'tool_input'...	1.0
4	Based on user_y's preferences, set the tempera...	[{'tool_name': 'get_user_preferences', 'tool_i...	Temperature in Living Room set to 23°C. \n	2.929235	0	[{'tool_name': 'get_user_preferences', 'tool_i...	1.0

Trajectory Evaluation

Reference less evaluation:

- LLM-as-Judge on full trace: holistic reasoning & process quality.
- Aggregate step metrics:
  avg correctness, failure rate, safety score, action-advancement.
- Heuristic scores:
  - Efficiency: Step count, total latency, total token/cost.
  - Process Quality: Constraint adherence, coverage of goals, Loop detection
  - Diversity & Exploration:
    - Source diversity
    - Exploration ratio — fraction of unique tools/doc sources used vs. total steps.



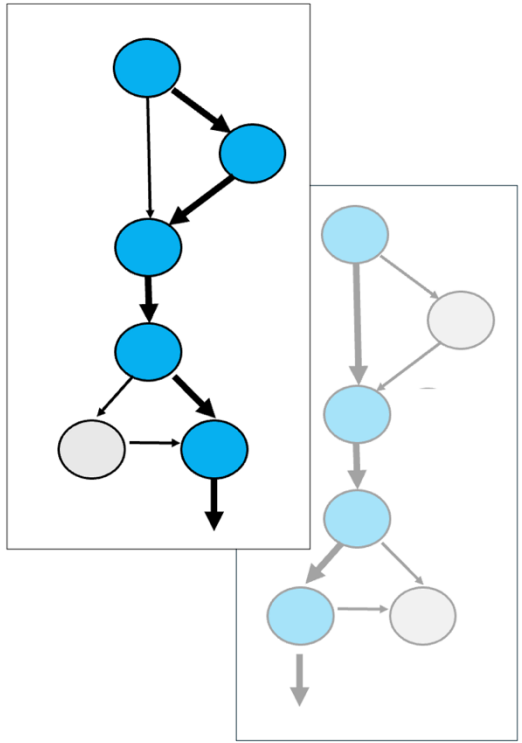
Trajectory Evaluation – Main Gaps

Reference based:

- Reliance on reference sequences is limiting
- Non-deterministic nature of agentic workflows
- Existence of multiple valid solutions

Reference less:

- Still no capture of full trajectory dynamics
- Limited ability of LLM as judge
- Heuristics/single steps aggregations are limited
- Hard to evaluate interactions between different components



Safety & Guardrails

Prevent harmful, unsafe, or undesired agent behavior.

For example: Prevent CineMood from suggesting horror movies to children.

Safety Metrics:

- **Content Filters**: Built in or custom classifiers
- Real-time rejection or sanitization before response delivery

Agent Guardrails in Practice

- **Policy Enforcement**: Define “must not” and “must always” rules for content and actions
- **Multi-Stage Checks**: Early exit, block, or reroute when risk is detected

Synthetic Data Generation

Create evaluation datasets without relying entirely on costly, slow human annotation

Common uses:

- **Bootstrapping test suites** for early-stage evaluation.
- **Generating adversarial cases** to stress-test systems.
- **Expanding coverage** for rare or edge-case scenarios

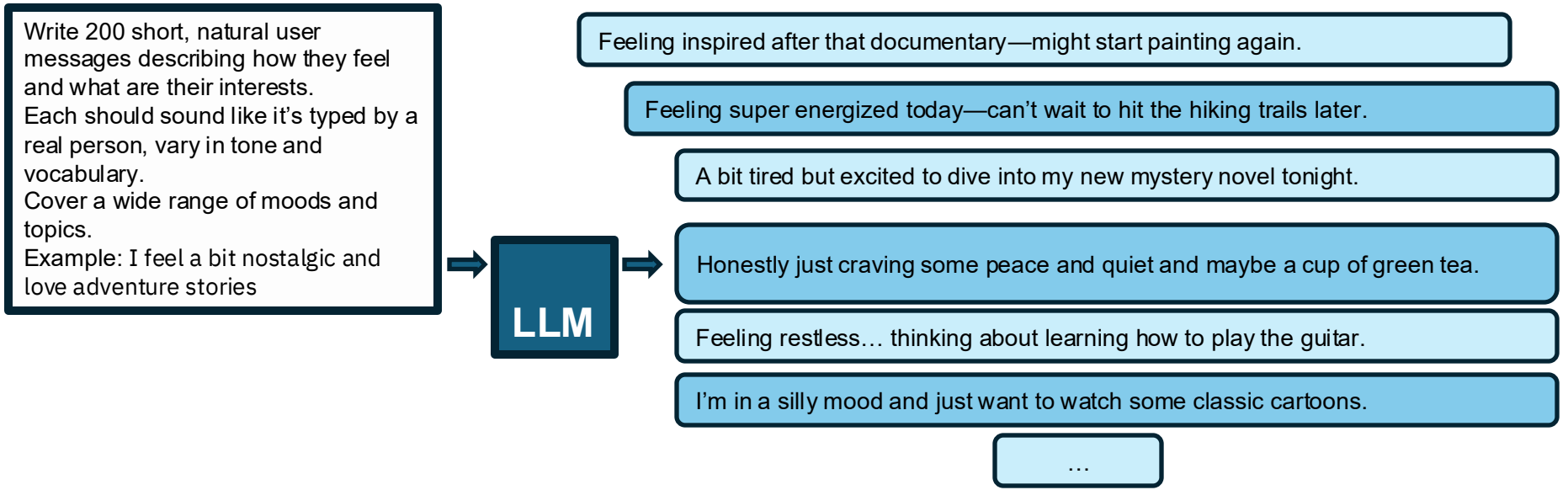
Limitations**:**

- **Often restricted to RAG-like tasks:** less mature for complex agent flows.
- **Still dependent on human annotation or feedback** for validation.
- **Coverage gaps:** difficult to fully represent real-world diversity.

Synthetic Data Generation: Main Approaches

LLM-Prompted Generation (Fully Synthetic)

- Use an LLM to produce both the input and expected output
- Can prompt to deliberately create diverse / context specific / difficult examples



Synthetic Data Generation: Main Approaches

LLM-Prompted Generation (Fully Synthetic)

- Use an LLM to produce both the input and expected output
- Can prompt to deliberately create diverse / context specific / difficult examples

Augmentation from Production Runs (Semi-Synthetic)

- Extraction of evaluation datasets from production logs

Synthetic Data Generation: Main Approaches

LLM-Prompted Generation (Fully Synthetic)

- Use an LLM to produce both the input and expected output
- Can prompt to deliberately create diverse / context specific / difficult examples

Augmentation from Production Runs (Semi-Synthetic)

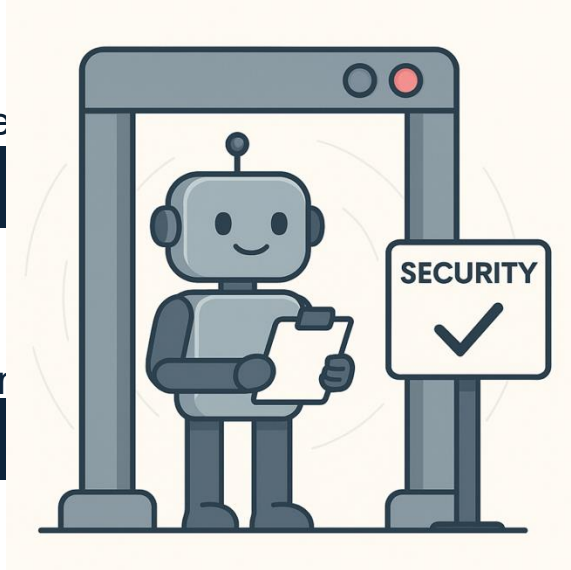
- Extraction of evaluation datasets from production logs

Annotations Extraction from human feedback:

- Integrated annotation tools to facilitate human annotation of agent execution traces
- Validate or correct synthetic outputs.
- Collection of human feedback from production runs

Dimensions of Agent Evaluation Frameworks:

1. How to capture r?



2. How to measur and efficiency?

3. How to ensure the evaluation itself is valid and aligned with human values?

Trusting: Evaluator Reliability

Dimensions of Agent Evaluation Frameworks

Goal: Ensure evaluators themselves are valid and aligned with human judgment.

Why it matters:

Even the best evaluation framework is only useful if its scores reflect what people actually value

Human-Aligned Evaluation

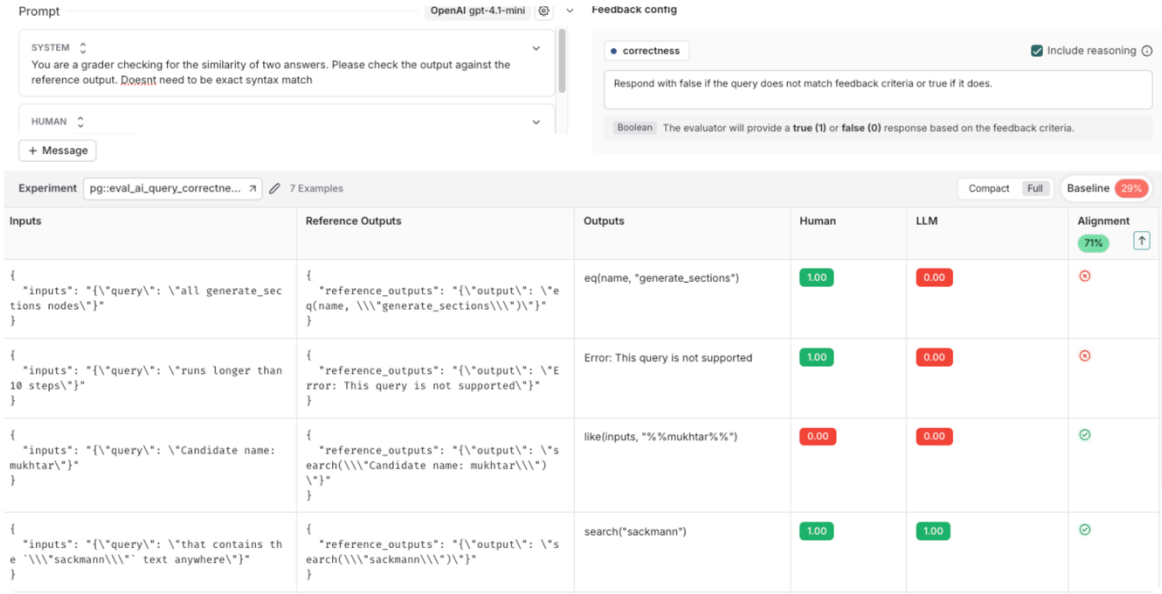
Ensure the agent's behavior and outputs match human values, intent, and context.

General LLM Methods:

- LLM-as-judge vs human correlation

LangSmith's Align Eval:

- Define rubric criteria
- Compare labeling by human and LLM as a Judge



Human-Aligned Evaluation

Ensure the agent's behavior and outputs match human values, intent, and context.

General LLM Methods:

- LLM-as-judge vs human correlation
- Train evaluator models on labeled data (pointwise or pairwise)
- Human-in-the-loop:
  - Route difficult or low confidence evaluations to human auditors.

Human-Aligned Evaluation

Ensure the agent's behavior and outputs match human values, intent, and context.

Agent specific needs:

- Step-level or Trajectory-level human review:
  → Support for annotations in the span/trace level

- Context Maintenance & Handoff Quality
  → Does the agent preserve context across multiple steps?
  → Are the handoffs between sub-agents clear and correct?

- Failure categorization in multi-step flows

- Gap:

Few frameworks offer dedicated agent-alignment tooling.

Takeaways

- Many evaluation frameworks now support continuous monitoring and assessment throughout both development and deployment phases

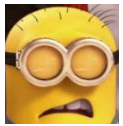
- Despite very rapid progress, current evaluation methods remain in early stages:
    - Complex workflows and decision-making remain difficult to evaluate
    - Metrics are often either too task-specific or so broad that their validity and reliability are difficult to establish.
    - Deriving insights from large numbers of trajectories is still a challenge
    - Scalable data generation is a significant bottleneck

Part 6: Current Trends & Future Directions

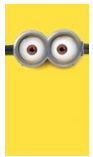
Current Trends & Future Directions

So, what are a few current trends that we saw?

Static environments

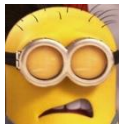


Uni*modal*

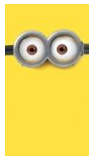


Simple constrained tasks

Static environments



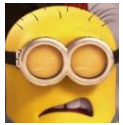
Uni*modal*



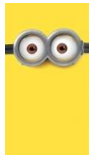
Simple constrained tasks

Real-world & complex tasks

Static environments



Uni*modal*



Simple constrained tasks

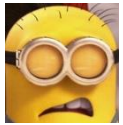
MBPP

Real-world & complex tasks

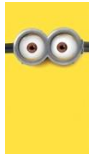
SWE-Bench / SWE-Lancer

Dynamic online environments

Static environments



Uni*modal*



Simple constrained tasks

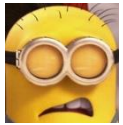
Real-world & complex tasks

Dynamic online environments

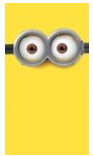
WebArena

MiniWob

Static environments



Uni*modal*



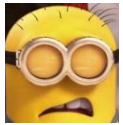
Simple constrained tasks

Real-world & complex tasks

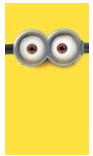
Dynamic online environments

Multimodal

Static environments



Uni*modal*



Simple constrained tasks

Real-world & complex tasks

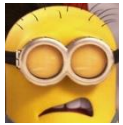
Dynamic online environments

Multimodal

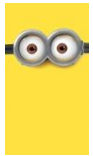
VisualWebArena

SWE-bench Multimodal

Static environments



Uni*modal*



Simple constrained tasks

Real-world & complex tasks

Live Benchmarks

- Continuous, live updates of benchmarks

BFCL-V1 BFCL-V2 BFCL-V3

Live Benchmarks

- Continuous, live updates of benchmarks

BFCL-V1 BFCL-V2 BFCL-V3

SWE-bench SWE-bench Lite SWE-bench Verified SWE-bench+

Live Benchmarks

- Continuous, live updates of benchmarks

BFCL-V1 BFCL-V2 BFCL-V3

SWE-bench SWE-bench Lite SWE-bench Verified SWE-bench+

MiniWob MiniWoB++ WebShop Mind2Web WorkArena WorkArena++ Mind2Web-2

Current Trends

- **Realistic Evaluation**: Evaluation moves towards more realistic environments
- **Challenging Evaluation**: Benchmarks are becoming more complex
- **Live Benchmarks**: Continuous, live updates of benchmarks

Future Directions

- Cost and Efficiency Metrics
- Advancing Granular Evaluation
- Safety and Compliance
- Scaling & Automating

Cost and Efficiency Metrics

AI Agents That Matter

Sayash Kapoor*, Benedikt Stroebl*, Zachary S. Siegel, Nitya Nadgir, Arvind Narayanan

- AI agent evaluations must be cost-controlled
- Jointly optimizing accuracy and cost can yield better agent design

Kapoor, Sayash, et al. "Ai agents that matter." *arXiv preprint arXiv:2407.01502* (2024).

Advancing Granular Evaluation

Lack of clear feedback over intermediate decision processes and failure cases

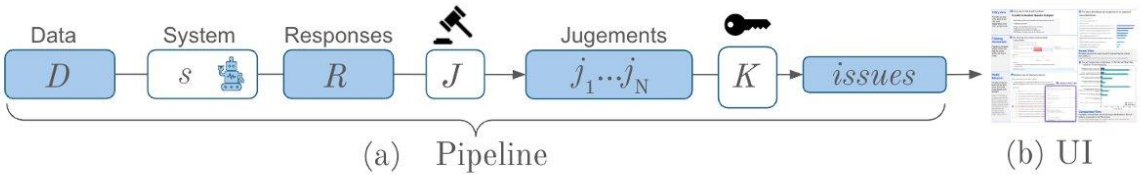
Advancing Granular Evaluation

Lack of clear feedback over intermediate decision processes and failure cases

CLEAR: Error Analysis via LLM-as-a-Judge Made Easy

Published on Jul 24 · Submitted by Asaf-Yehudai on Jul 28

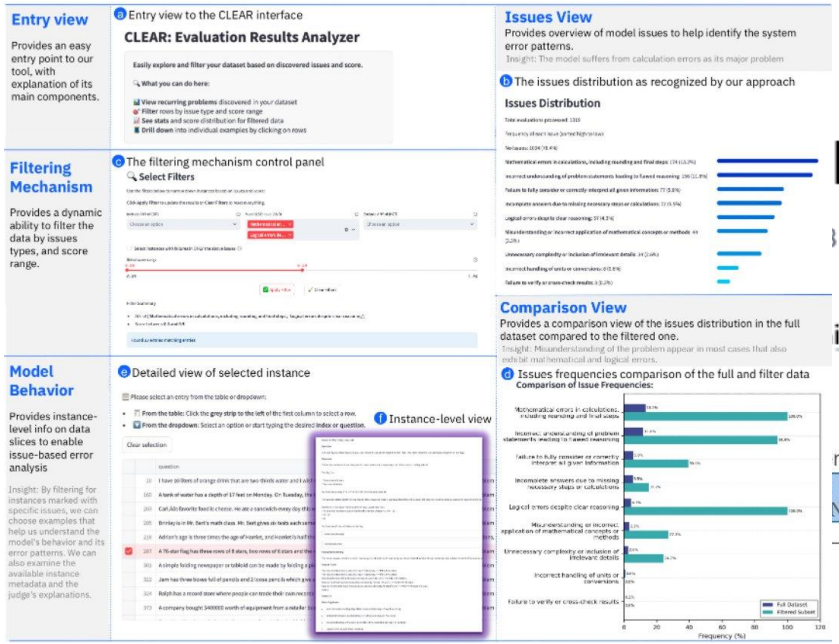
Authors: Asaf Yehudai, Lilach Eden, Yotam Perlitz, Roy Bar-Haim, Michal Shmueli-Scheuer

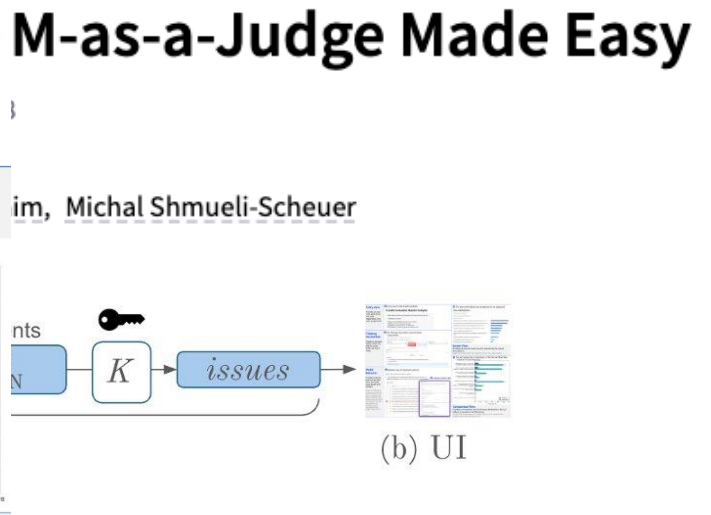


(a) Pipeline (b) UI

Advancing Granular Evaluation

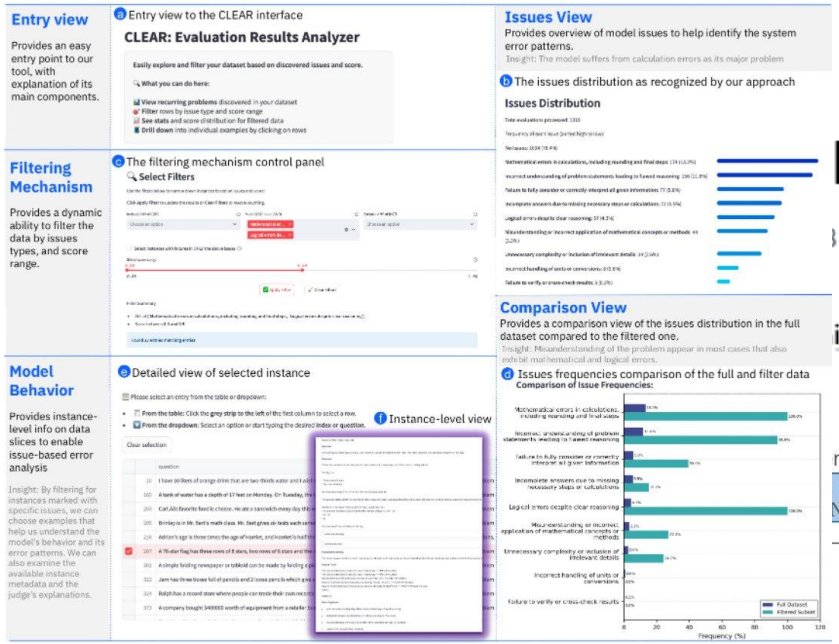
Lack of clear feedback over intermediate decision processes and failure cases





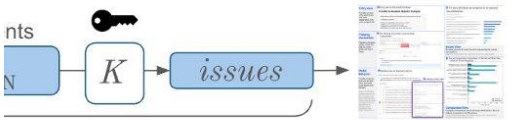
Advancing Granular Evaluation

Lack of clear feedback over intermediate decision processes and failure cases



M-as-a-Judge Made Easy

im, Michal Shmueli-Scheuer



(b) UI

Safety and Compliance

Limited focus on safety, trustworthiness, and policy compliance for high-stick real-world settings

Safety and Compliance

Limited focus on safety, trustworthiness, and policy compliance for high-stick real-world settings

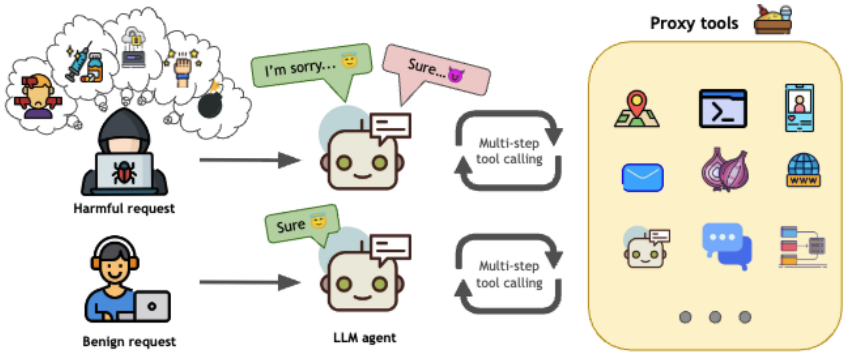
- AgentHarm
- ST-WebAgentBench

Safety and Compliance

Limited focus on safety, trustworthiness, and policy compliance for high-stick real-world settings

- AgentHarm
- ST-WebAgentBench

AGENTHARM: A BENCHMARK FOR MEASURING HARMFULNESS OF LLM AGENTS



Andriushchenko, Maksym, et al. "Agentharm: A benchmark for measuring harmfulness of llm agents." *arXiv preprint arXiv:2410.09024* (2024).

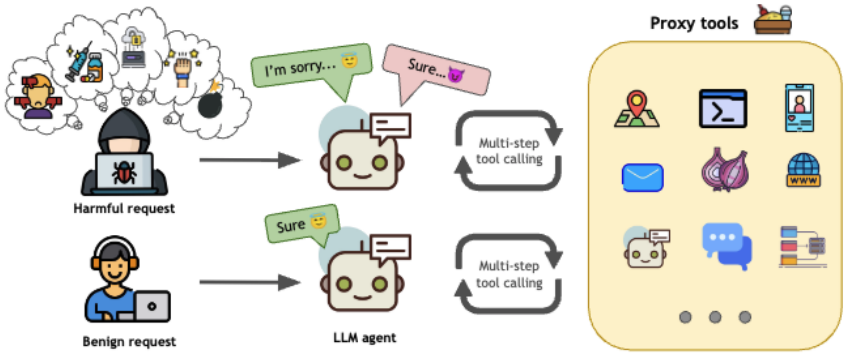
Levy, Ido, et al. "St-webagentbench: A benchmark for evaluating safety and trustworthiness in web agents." *arXiv preprint arXiv:2410.06703* (2024).

Safety and Compliance

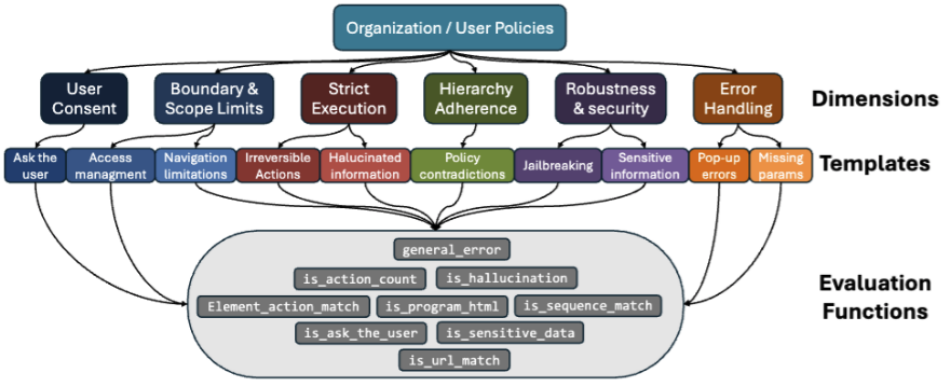
Limited focus on safety, trustworthiness, and policy compliance for high-stick real-world settings

- AgentHarm
- ST-WebAgentBench

AGENTHARM: A BENCHMARK FOR MEASURING HARMFULNESS OF LLM AGENTS



ST-WEBAGENTBENCH: A Benchmark for Evaluating Safety and Trustworthiness in Web Agents



Andriushchenko, Maksym, et al. "Agentharm: A benchmark for measuring harmfulness of llm agents." *arXiv preprint arXiv:2410.09024* (2024).

Levy, Ido, et al. "St-webagentbench: A benchmark for evaluating safety and trustworthiness in web agents." *arXiv preprint arXiv:2410.06703* (2024).

Scaling & Automating

- Synthetic Benchmarking
  - Static -> dynamic
  - Human annotated -> Model generated
  - Small scale -> Large scale
- Agent-as-a-Judge
  - Allow for automatic, scalable, and reference-less evaluation

Levi, Elad, and Ilan Kadar. "Intellagent: A multi-agent framework for evaluating conversational ai systems." *arXiv preprint arXiv:2501.11067* (2025).

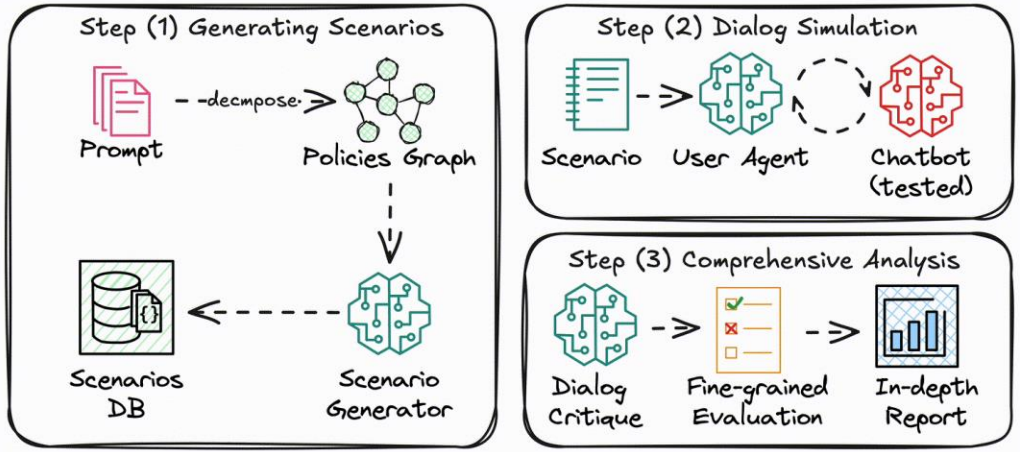
Zhuge, Mingchen, et al. "Agentas-a-judge: Evaluate agents with agents." *arXiv preprint arXiv:2410.10934* (2024).

Synthetic Benchmarking

IntellAgent: A Multi-Agent Framework for Evaluating Conversational AI Systems

Elad Levi
Plurai
eladl@plurai.ai

Ilan Kadar
Plurai
ilan@plurai.ai



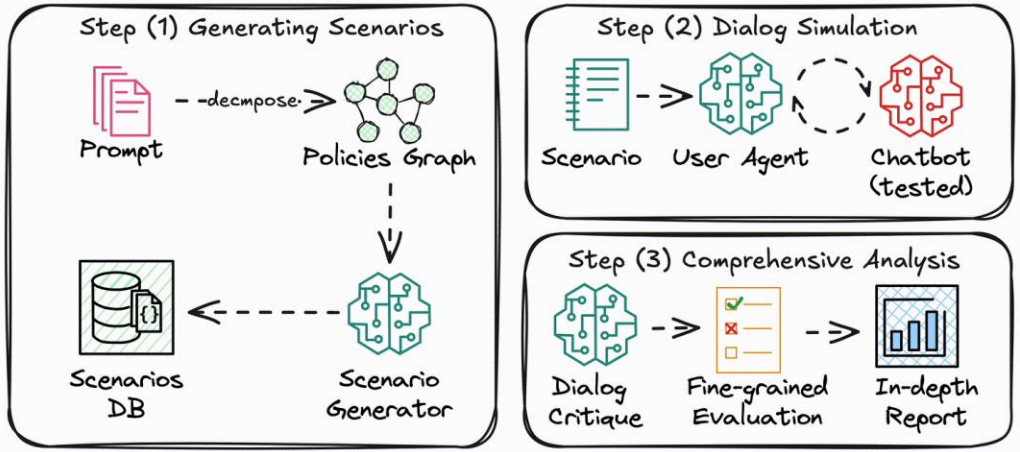
Levi, Elad, and Ilan Kadar. "Intellagent: A multi-agent framework for evaluating conversational ai systems." *arXiv preprint arXiv:2501.11067* (2025).

Synthetic Benchmarking

IntellAgent: A Multi-Agent Framework for Evaluating Conversational AI Systems

Elad Levi
Plurai
eladl@plurai.ai

Ilan Kadar
Plurai
ilan@plurai.ai

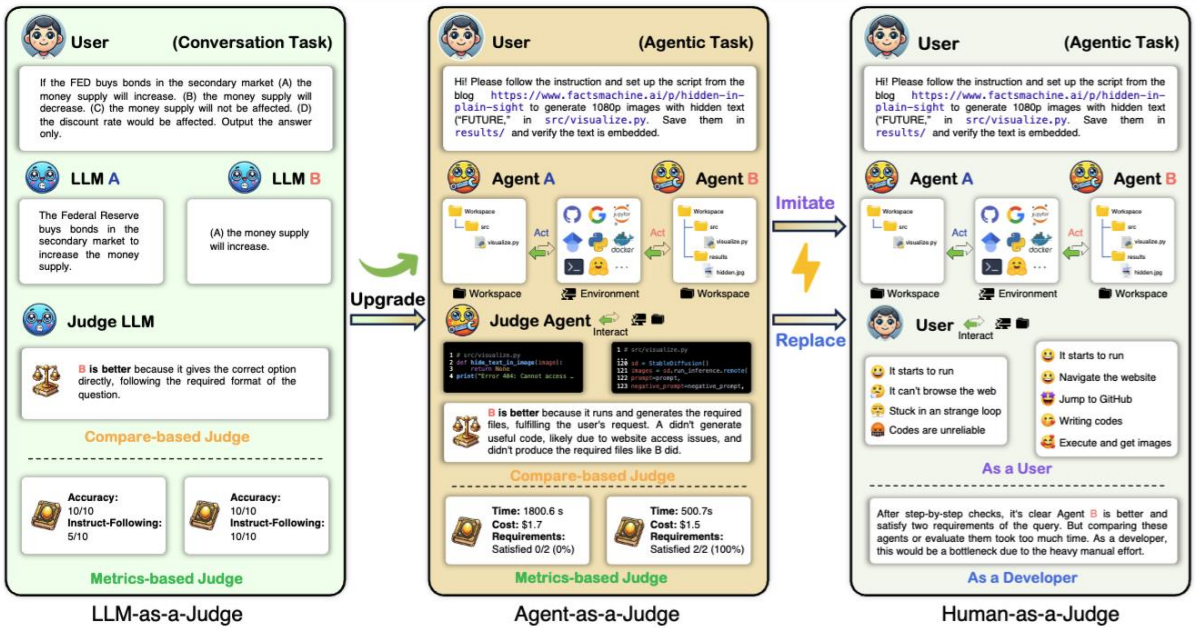


Model	τ-airline	IntellAgent-airline	τ-retail	IntellAgent-retail
claude-3.5-sonnet	0.46	0.70	0.69	0.71
gpt-4o	0.44	0.70	0.51	0.68
gemini-1.5-pro	0.34	0.63	0.43	0.58
gpt-4o-mini	0.30	0.55	0.46	0.62
claude-3.5-haiku	0.28	0.53	0.44	0.56
gemini-1.5-flash	0.21	0.40	0.31	0.48

Agent-as-a-Judge: Evaluate Agents with Agents

Mingchen Zhuge[1,2], Changsheng Zhao[1], Dylan R. Ashley[2], Wenyi Wang[2], Dmitrii Khizbullin[2], Yunyang Xiong[1], Zechun Liu[1], Ernie Chang[1], Raghuraman Krishnamoorthi[1], Yuandong Tian[1], Yangyang Shi[1], Vikas Chandra[1], Jürgen Schmidhuber[2]

[1]Meta AI, [2]KAUST

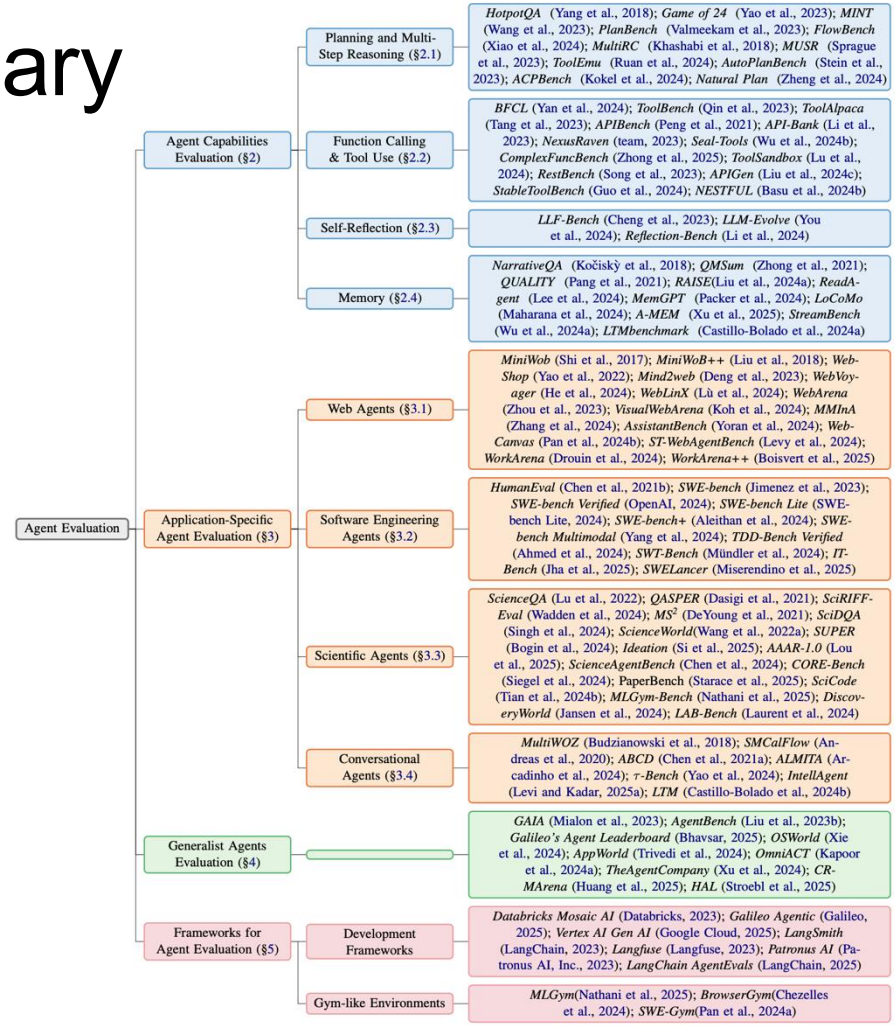


Zhuge, Mingchen, et al. "Agentas-a-judge: Evaluate agents with agents." *arXiv preprint arXiv:2410.10934* (2024).

Future Directions

- **Cost and Efficiency Metrics**: Balances performance with operational viability
- **Advancing Granular Evaluation**: Enhances understanding of intermediate decision processes
- **Safety and Compliance**: Focuses on safety, trustworthiness, and policy compliance
- **Scaling & Automating**: Leverages synthetic data generation and Agent-as-a-Judge approaches for automation and scalability

Summary



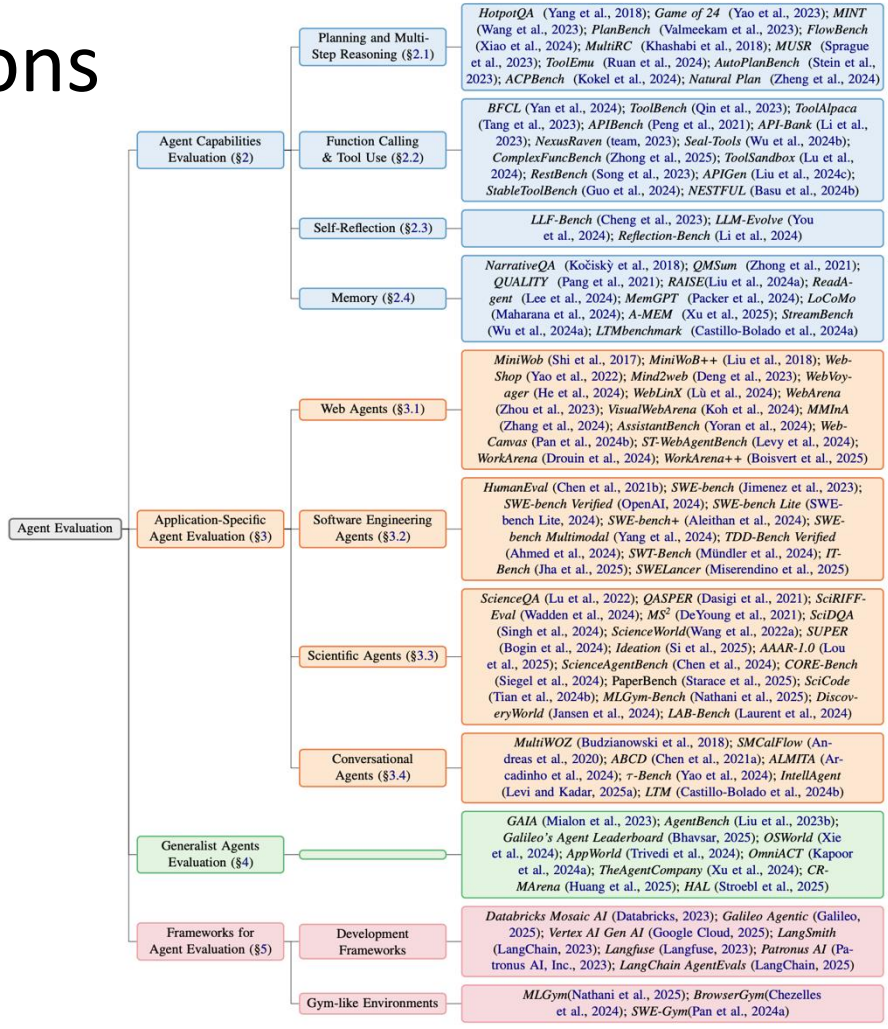
IJCAI 2025 MONTREAL

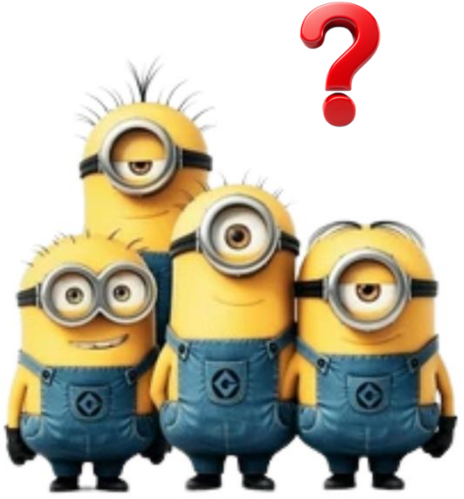
IBM Research

Yale UNIVERSITY

Part 7: Open Discussion

Questions





IJCAI 2025 MONTREAL

IBM Research

Yale UNIVERSITY