Section 7.2: Agent Benchmarking : A survey

2026 Spring

LLM Agents Foundation & Applications

Student Team

20260331

We have covered

- S1: LLM Basic Alignment
- S2: LLM Alignments for Reasoning
- S3: Agent applications
- S4: LLM Data synthesis
- S5: Agent Memory
- S6: LLM model serving
- S7: Agent Evaluation and Attack/Defense Landscape ➔ This lecture!
- S8: Agent Planning
- S9: World Modeling for GenAI Agents
- S10: Multi-Agents

[Submitted on 20 Mar 2025]

Survey on Evaluation of LLM-based Agents

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

The emergence of LLM-based agents represents a paradigm shift in AI, enabling autonomous systems to plan, reason, use tools, and maintain memory while interacting with dynamic environments. This paper provides the first comprehensive survey of evaluation methodologies for these increasingly capable agents. We systematically analyze evaluation benchmarks and frameworks across four critical dimensions: (1) fundamental agent capabilities, including planning, tool use, self-reflection, and memory; (2) application-specific benchmarks for web, software engineering, scientific, and conversational agents; (3) benchmarks for generalist agents; and (4) frameworks for evaluating agents. Our analysis reveals emerging trends, including a shift toward more realistic, challenging evaluations with continuously updated benchmarks. We also identify critical gaps that future research must address-particularly in assessing cost-efficiency, safety, and robustness, and in developing fine-grained, and scalable evaluation methods. This survey maps the rapidly evolving landscape of agent evaluation, reveals the emerging trends in the field, identifies current limitations, and proposes directions for future research.

Evaluation of LLM-based Agents

CS 6501: GenAI Foundation and LLM Agents

Presented by Mengmeng Ma

March 30, 2026

From a horse to the utility vehicle

The LLM

Raw power, no direction

Parameters | Pretraining | Capability

The System & Tools

Structure around the model

Tool use | Guardrails | Memory

The Agent

Directed, reliable, useful

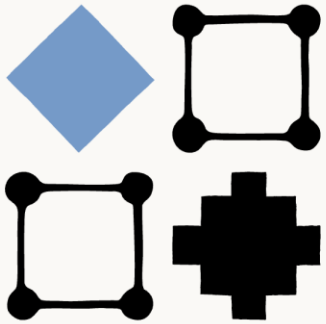
Goal-directed | Observable | Reliable

From ***model*** engineering to ***harness*** engineering

February 11, 2026 Engineering

Harness engineering: leveraging Codex in an agent-first world

Engineering at Anthropic



Harness design for long-running application development

Published Mar 24, 2026

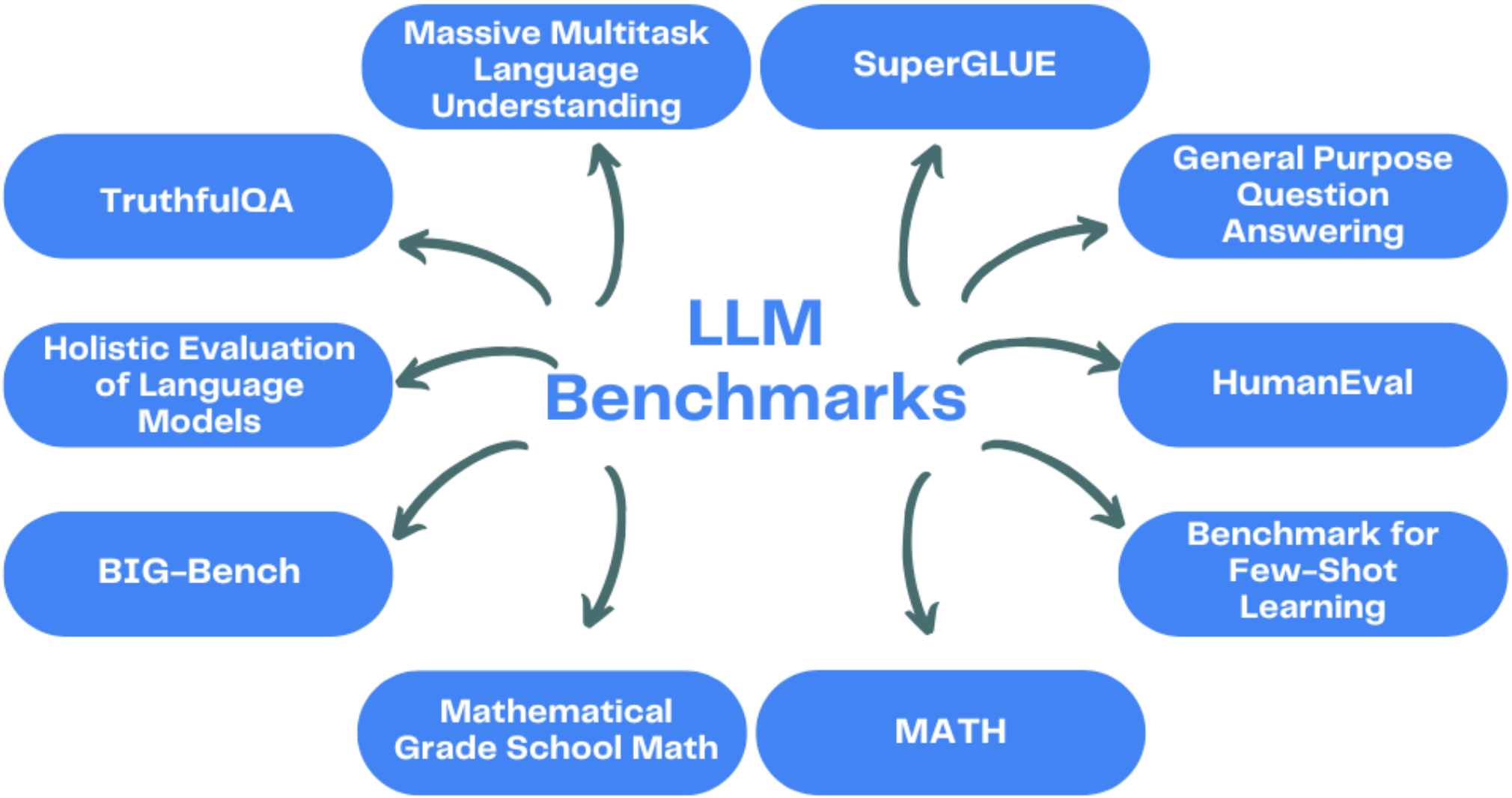
Harness design is key to performance at the frontier of agentic coding. Here's how we pushed Claude further in frontend design and long-running autonomous software engineering.

*Reliable agents are increasingly seen **not just as a model problem,** but **harness-design problem** (the control, tool-use, monitoring, and testing layer around a model).*

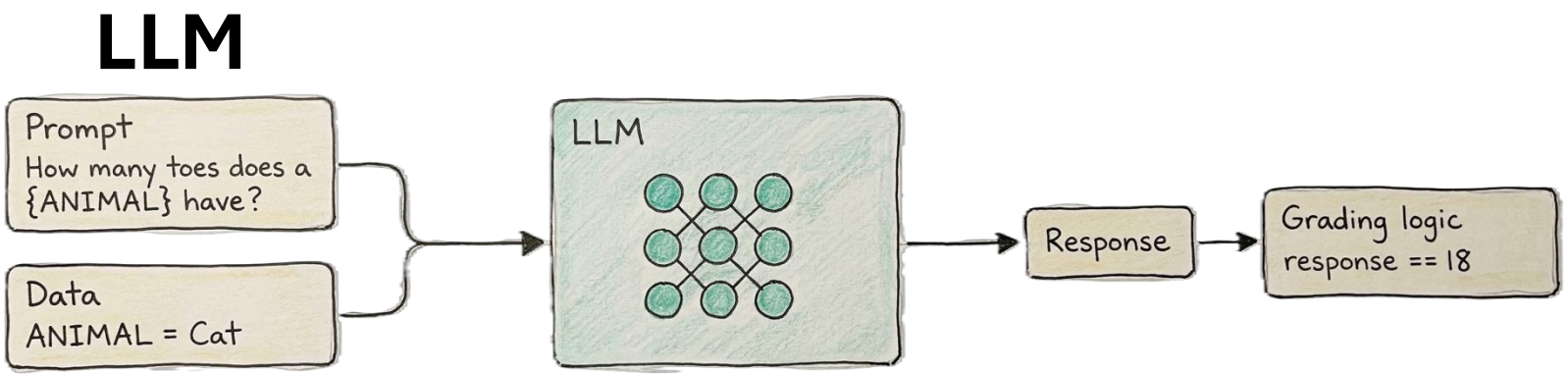
If agents are increasingly designed as complex systems,

how should we evaluate them?

Conventional LLM evaluation falls short for agent systems

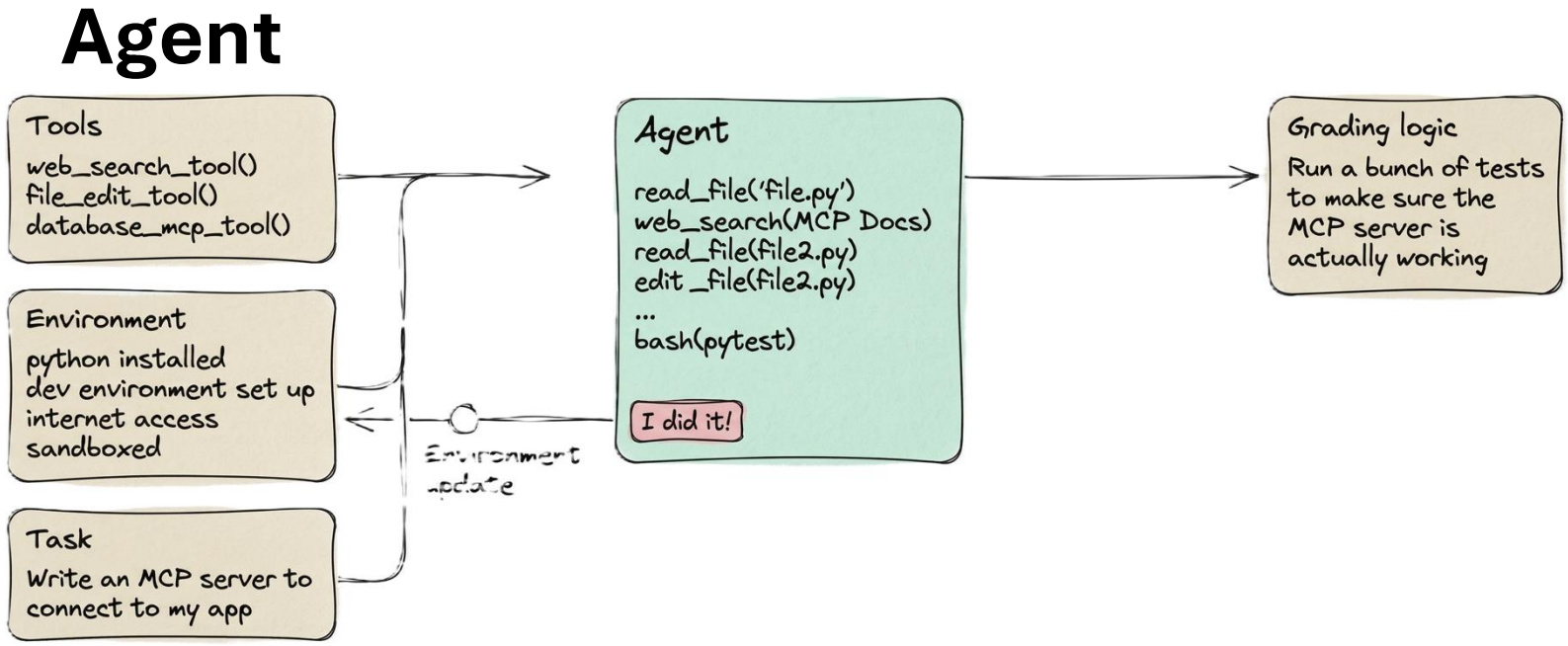


LLM evaluation *vs.* Agent evaluation



LLM evaluation
(**final outputs**):

- Is the final answer correct?
- Is the response fluent?
- Does it follow instructions?



Agent evaluation
(**full trajectory**):

- Planning; tool calling;
- observing feedback;
- revising;
- recovering from mistakes

Today's Talk: Evaluation of LLM-based agent

1 How to evaluate?

Data; metric; framework

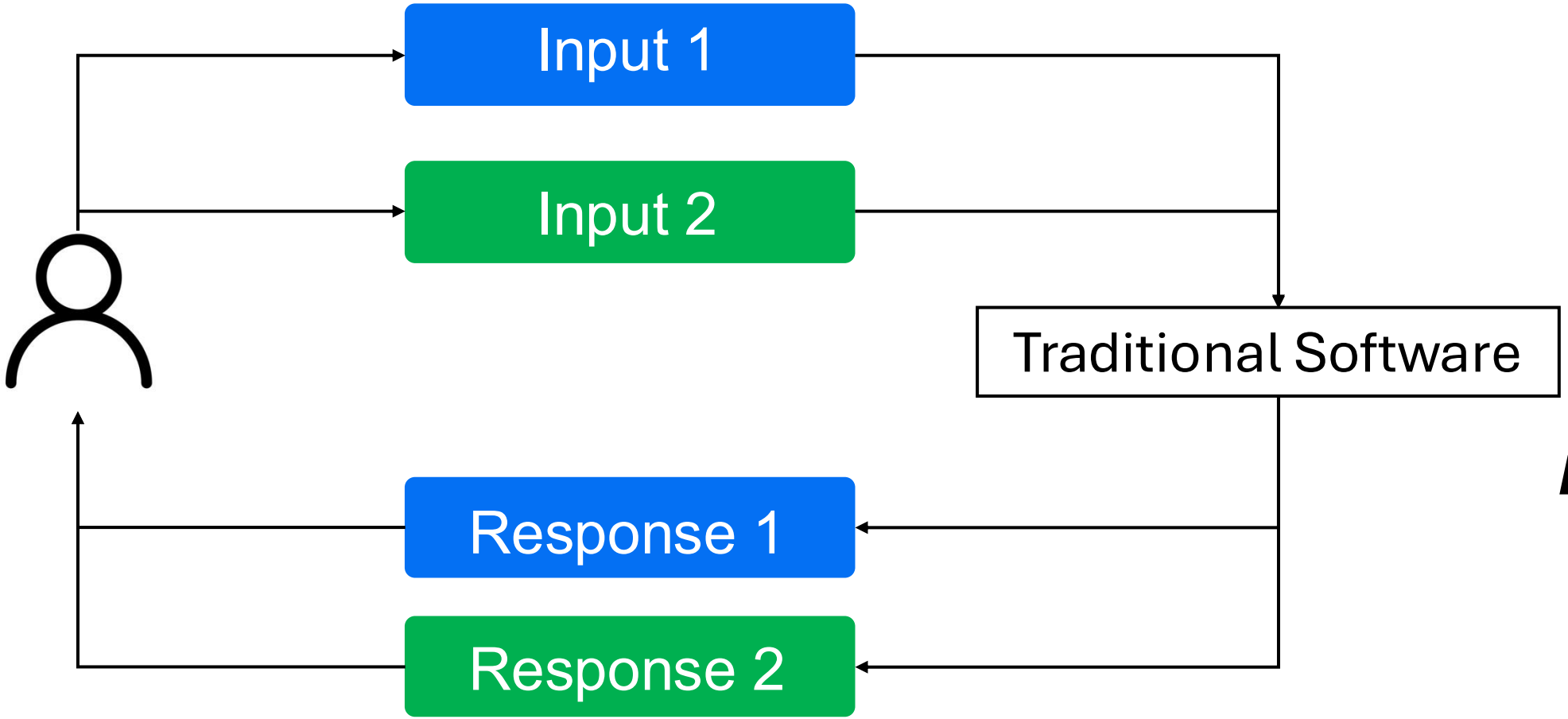
2 What to evaluate?

Core capability + Specific applications

3 Future directions

Holistic evaluation; Realistic setting; Saling and automating

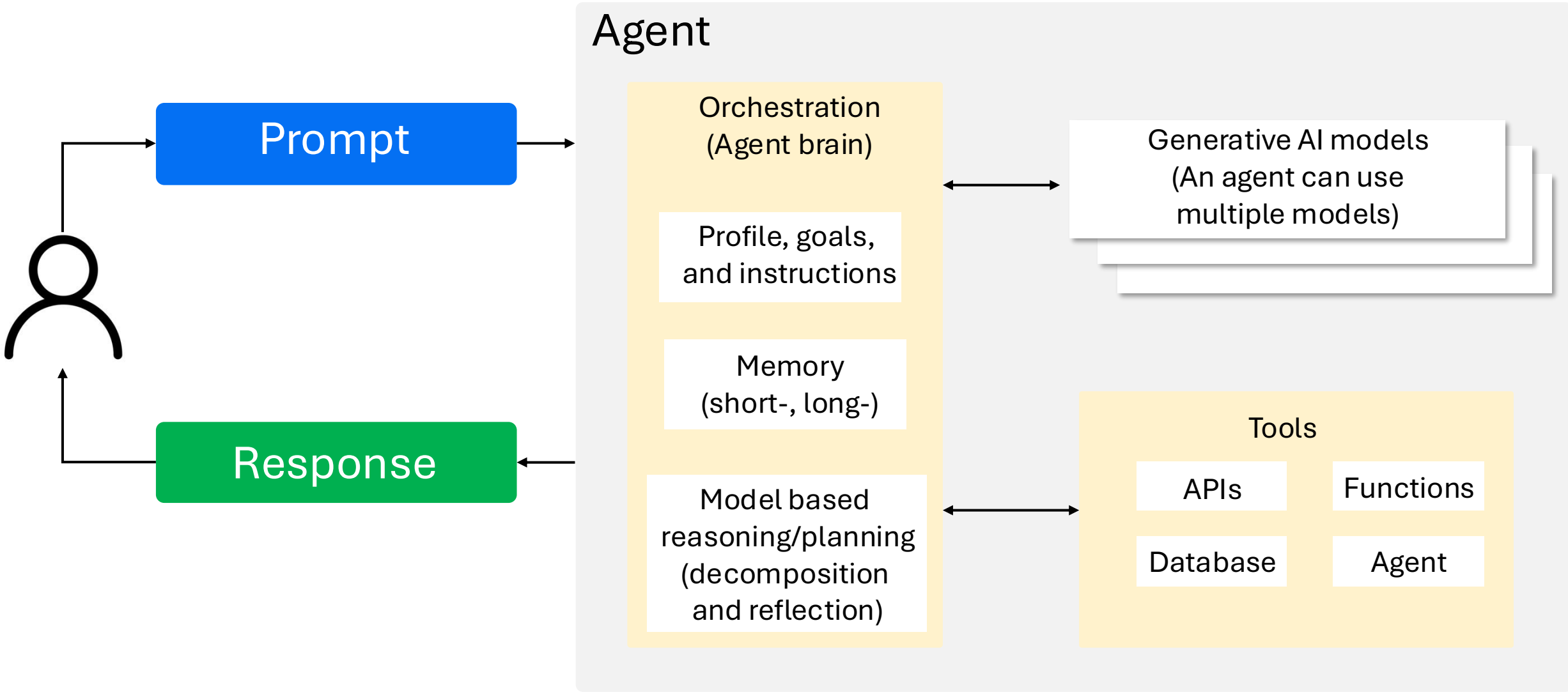
LLM agent vs. Traditional Software



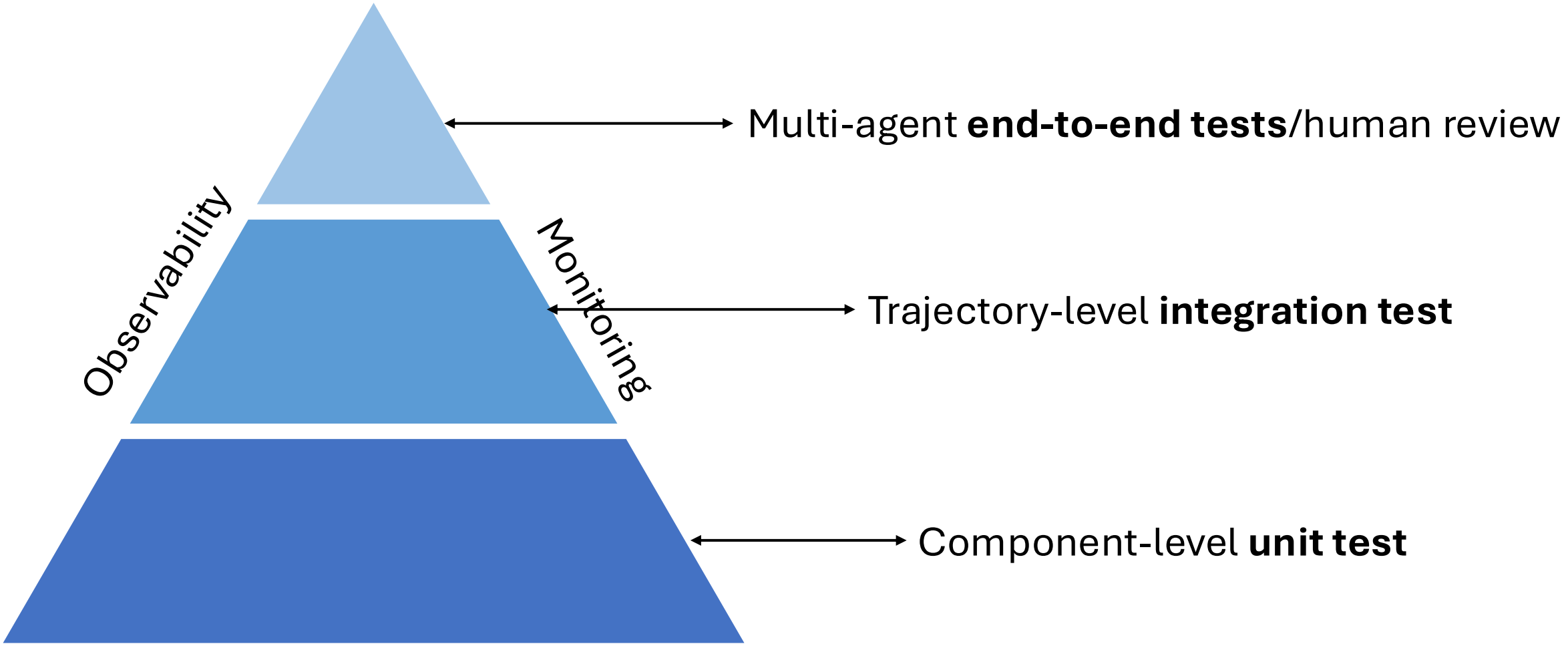
Deterministic!

LLM agent vs. Traditional Software

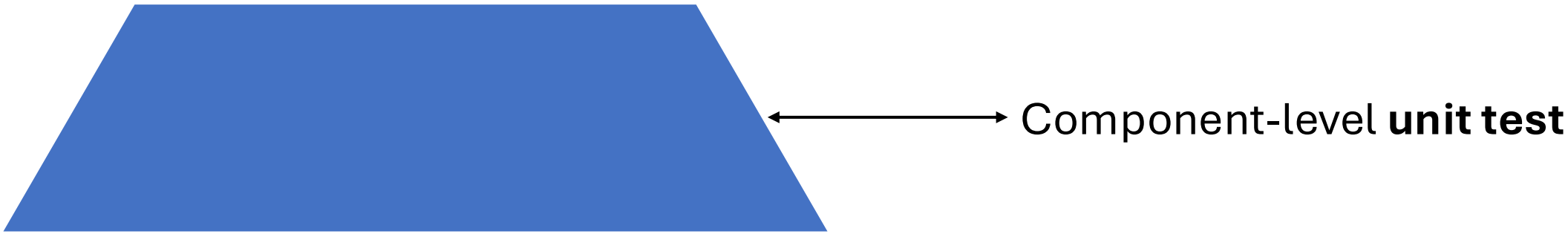
Probabilistic!



The agent evaluation pyramid



The agent evaluation pyramid



The agent evaluation pyramid

Final outcome

Success rate;

Output quality.

Reasoning and planning

Decompose complex Tasks;
Set logically goals.

Tool use

Pick the right tools;

Pass right parameters;

Handle outputs.

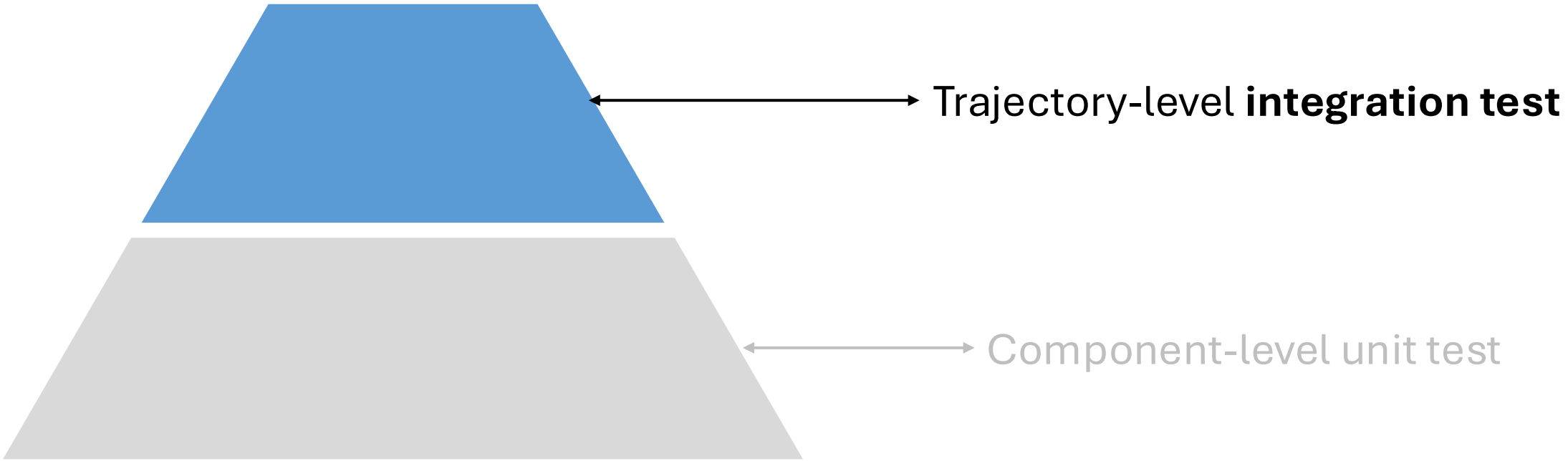
Memory

Retrieve past info;
Handle long-context and conflicts

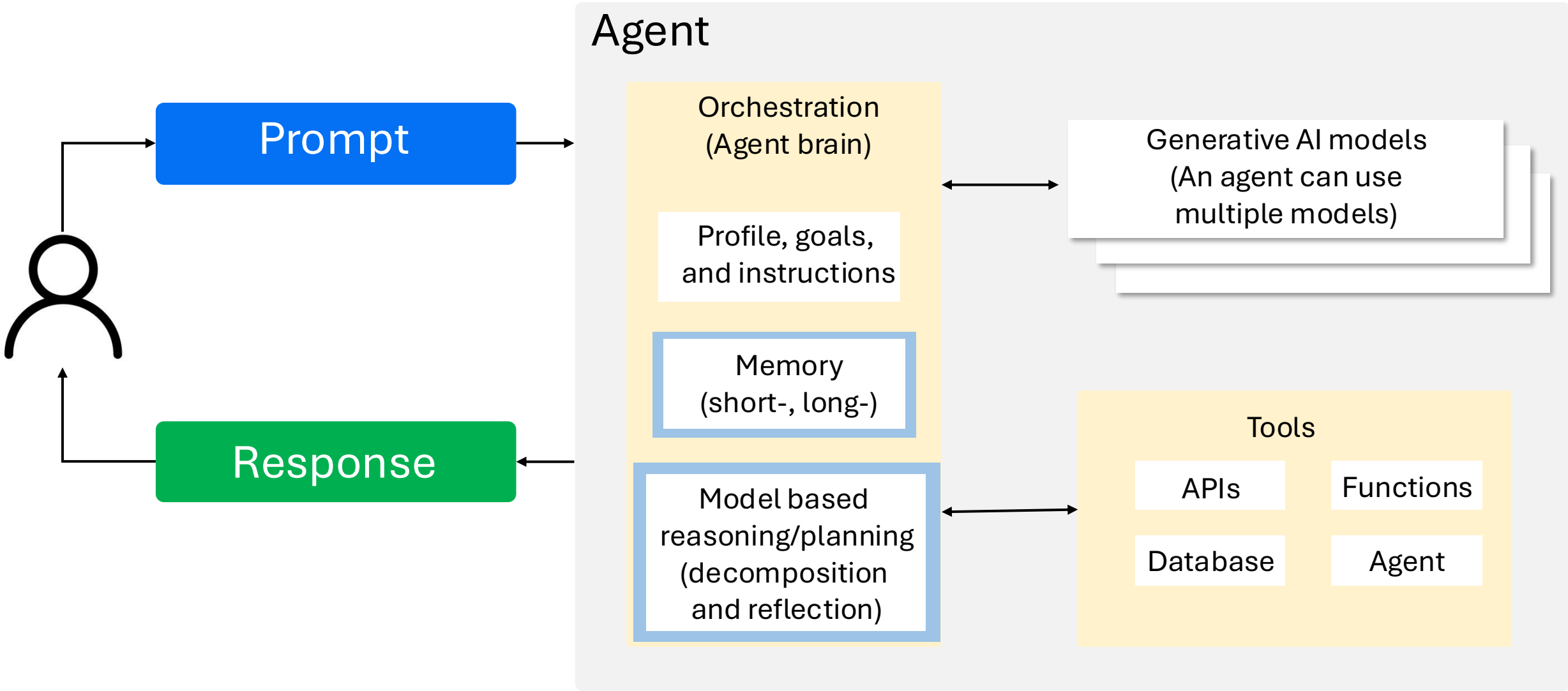
Testing smallest building block in isolation.

Evaluation: **Cheap, Fast,** and **Automated.**

The agent evaluation pyramid

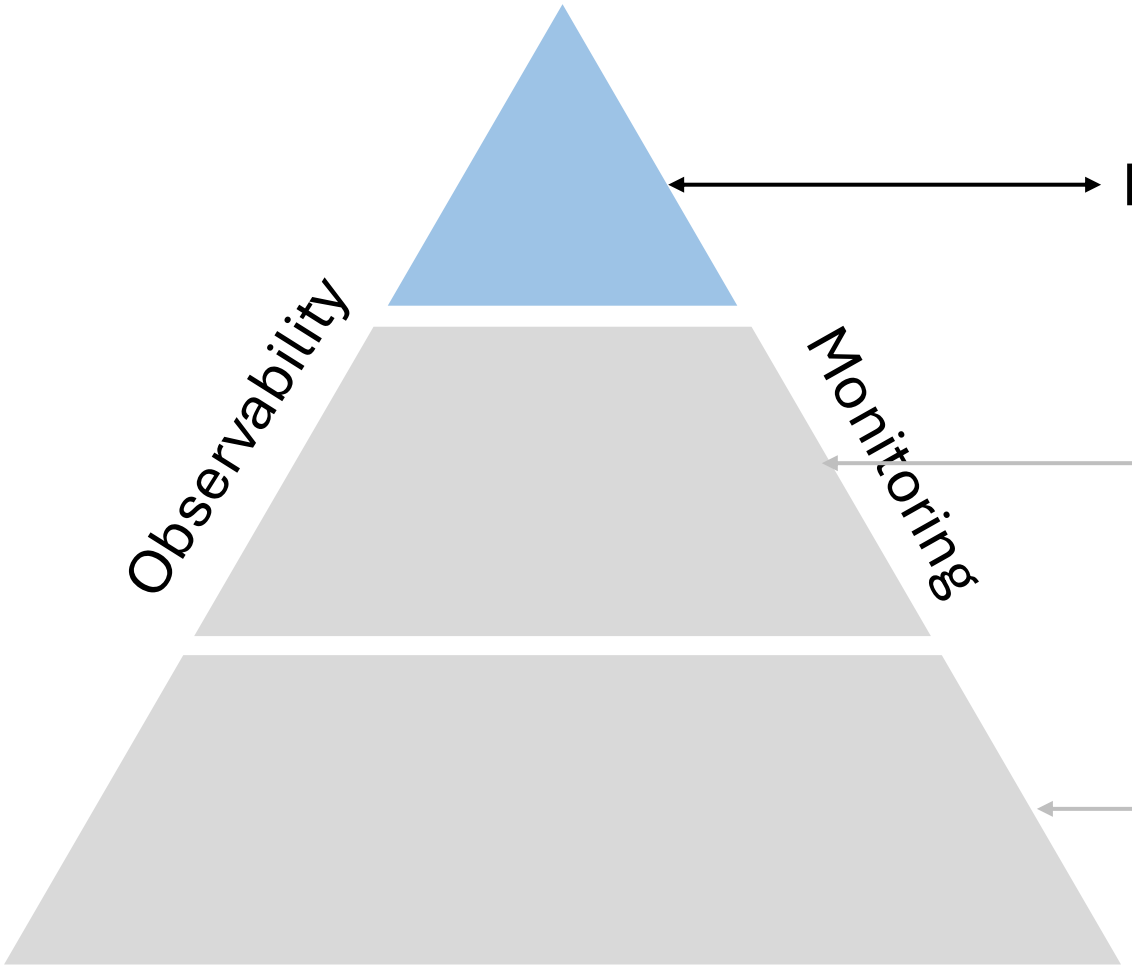


LLM agent vs. Traditional Software



Testing a full multi-step task end-to-end.

The agent evaluation pyramid



Multi-agent **end-to-end tests**/human review

Trajectory-level **integration test**

Component-level **unit test**

Involve humans to check helpfulness, commonsense, ...

Slow and **Expensive**

How to evaluate LLM agents?

Dimension	Description	Subcategories
Interaction Mode	How is the evaluation data provided to the system? For multi-turn, is the data flexible?	Static (Offline) vs. Dynamic (Online)
Evaluation Data	What data do we use to evaluate the system? How do we obtain it?	Data Sources, Data Generation, Benchmarks
Metrics Computation Methods	What method do we use to compute evaluation metrics?	Code Based, LLM-as-a-Judge, Human-as-a-Judge
Evaluation Tooling	What kinds of pre-existing tooling exists to support LLM agent evaluation?	Testing, Observability, Debugging, Monitoring
Evaluation Contexts	In what environments do we test the LLM agent?	Mocked APIs, Simulators, Live

Evaluation Process: **Interaction Modes**

Static / Offline

Agents are **evaluated in isolation** using **predefined prompts or datasets**, with no live interaction or dynamic responses based on previous turns.

Advantages

- **Reproducible and comparable results** between agent system versions.
- Static data means **lower cost; no need for live system integration.**

Limitations

- Prone to **error propagation in multi-turn conversations** if the system does not follow the sample response exactly.
- Fails to capture **emergent behavior**, such as tool failures, response drift, and adaptation.

Dynamic / Online

Agent evaluation happens in a **live or simulated environment**, where the agent interacts in real-time with tools (APIs, browsers), users, or environments. Outputs evolve across multi-turn conversations or tool-based workflows.

Advantages

- Captures **real-world complexity** (e.g., dynamic user or API responses).
- Tests **multi-turn reasoning** and **adaptive planning.**

Limitations

- Requires **simulation environments** and/or **live tool integrations**.
- Costly; **needs infrastructure** for tracking failures, latency, and human-in-the-loop feedback.

Evaluation Process: **Evaluation Data**

Data Types

Human-Annotated: Human labeled examples. Contains the most domain knowledge, policy understanding, and nuance.

Synthetic: Programmatically generated data, best utilized for reliability and robustness coverage. Cheap and scalable but may be lower quality.

Interaction-Generated: Data collected from real agent usage. The most representative of end-user interactions and usage.

Properties to Consider

Domain Specificity: Domain specific integrations (e.g., legal, medical) and enterprise constraints or policy.

Task Structure: Slot filling, disambiguation, multi-step, information retrieval, conversation length, etc.

Evaluation Process: **Evaluation Data**

Notable Benchmarks by Objective

Objective	Datasets/Benchmarks
Tool Use	ToolBench, API-Bank
Planning	TaskBench, ScienceAgentBench
Safety	AgentHarm, CoSafe, AgentDojo
Long-Term Memory	LongEval, SocialBench
Web Interaction	WebArena, BrowserGym

Evaluation Process: **Metrics Computation Methods**

Code Based

Evaluation via hard-coded rules or assertions that compare the agent's output to a known ground truth. Often used in tasks with structured outputs like code, APIs, or JSON.

Strengths

- **Deterministic:** Consistent, rule based scoring.
- **Reproducible:** Easy to automate and rerun. Great for structured formats.

Limitations

- **Brittle:** Small variations = failure.
- **Structural Requirements:** Poor at evaluating free-form responses.
- **Content Only:** Doesn't measure semantic equivalence or intent match.

LLM-as-a-Judge

A separate LLM is used to evaluate responses on criteria like clarity, reasoning, or satisfaction. Often used in subjective tasks, such as summarization or decision-making.

Strengths

- **Flexible Success Parameters:** Handles ambiguity and subjectivity.
- **Speed:** Quickly make judgements on unstructured, long form outputs.

Limitations

- **Hallucinations:** LLMs may hallucinate or provide incorrect objective assessments.
- **Fairness:** Special care must be taken to ensure fair and consistent grading for subjective metrics.

Human-as-a-Judge

Human judges annotate and/or score agent outputs by hand. Often used for assessing crucial subjective measures such as trust, safety, ethics, and satisfaction.

Strengths

- **Edge Cases:** Can flexibly assess edge cases, especially in niche or specialized domains.
- **Human Lens:** Provides human knowledge, nuance, and context.

Limitations

- **Poor Scalability:** Slow and costly to employ human experts to manually annotate data. Difficult to scale across tasks.

Evaluation Process: **Evaluation Tooling**

Enable scalable, repeatable, and automated evaluation pipelines, especially in **continuous deployment workflows**.

Some Evaluation Frameworks

Tool	Description
OpenAI Evals	YAML-based tests for LLMs, extensible for agents
DeepEval	Open-source metric + dataset evaluation runner
InspectAI	Input/output filtering, agent performance instrumentation
Phoenix (Arize)	ML observability and debugging
LangGraph, AgentOps	Monitoring agents in production

Evaluation Process: **Evaluation Contexts**

Evaluation Context = Testing Environment

Dimensions

- **Sandbox** vs. **Live Environment**
- **Simulated APIs** vs. **Real Services**
- **Open-world** (web) vs. **Controlled UI**

Use Case Examples

- **MiniWoB / WebArena:** Agents use browser-like sandbox
- **LangGraph:** Simulates workflows in business pipelines
- **AppWorld:** Mobile UI navigation with changing state

Trade Offs

Context Type	Pros	Cons
Mocked APIs	Reproducible, safe	Low realism, static tests only
Live	Realistic failures	Unstable, costly
Enterprise Simulator	Policy testing	Hard to generalize, costly

What are we evaluating?

Core capability

- **Planning and reasoning**
- **Tool use**
- **Self-reflection**
- **Memory**

Applications

- Web Agent
- SWE Agent
- Scientific Agent
- Conversation Agent

What are we evaluating?

Core capability

- **Planning and reasoning**
- **Tool use**
- **Self-reflection**
- **Memory**

Applications

- Web Agent
- SWE Agent
- Scientific Agent
- Conversation Agent

Agent Capabilities: **Planning & Reasoning**

Planning & Reasoning assesses the LLM agent's ability to plan multi-step actions and **adapt reasoning** to dynamic contexts. It is especially important for complex or long-horizon tasks, where multiple tool calls are important to solving the given task.

Subcategory	Metric	Description
Planning	**Plan Quality**	How well the agent's generated plan aligns with an expert or ground-truth multi-step plan.
	Node F1	Accuracy in selecting the correct tools or actions (nodes) used in a plan.
	Step Success Rate	Percentage of steps in a plan that are executed successfully.
Reasoning	**Next-tool Prediction Accuracy**	How accurately the agent predicts the next correct tool at each reasoning step.
	Fine-Grained Progress Rate	Quantifies how closely the agent's execution trajectory matches the expected one at each step.

Relevant Tooling & Benchmarks

- **ReAct**: Reasoning-Action loops.
- **AgentBoard:** Offers fine-grained progress rate metric.
- **T-Eval:** Evaluates step-by-step tool-utilization capability.
- **ScienceAgentBench:** Tasks in data-driven scientific discovery.

Agent Capabilities: **Tool use**

Tool Use measures the agent's ability to invoke tools (APIs, functions) effectively to complete a task. It answers questions such as "Should a tool be used?" Or "Which tools are appropriate?" Or "Are the parameters extracted and filled correctly?"

Metric	Description
Invocation Accuracy	Measures if the agent **correctly decides to call a tool** when needed [3]
Tool Selection Accuracy, MRR, NDCG	Evaluate how well the agent **chooses the right tool** among candidates, including ranking its choices [3]
Parameter F1, AST correctness	Assess whether the agent **generates correct parameter names and values** for tool calls, with syntactic accuracy [1]
Execution-Based Success	Checks if the tool calls **actually run correctly** and achieve the intended result [2]

Relevant Tooling & Benchmarks

- **ToolEmu**: Evaluates agents by simulating tool execution environments, without requiring actual tool calls [1]
- **Gorilla**: Evaluates agents on their ability to call and integrate massive sets of real-world APIs accurately [2]
- **MetaTool**: Focuses on tool usage awareness—assessing whether an agent can correctly determine when a tool is needed [3]

Agent Capabilities: **Self-reflection**

Self-reflection requires the model to understand the feedback and dynamically update its beliefs to carry out adjusted actions or reasoning steps over extensive trajectories.

Metric	Description
Success Rate (SR)	% of tasks where the agent achieves the main goal completely.
Pass@k	Did any of k trials succeed?

Relevant Benchmarks

- **LLF-Bench:** A standardized benchmark for interactive self-reflection. It diverse decision-making tasks and incorporates task instructions as part of the environment rather than as part of the agent.
- **LLM-Evolve**: Evaluate self-reflection on standard benchmark such as MMLU.
- **Refection-Bench**: Designed to assess LLMs' cognitive reflection capabilities, breaking down reflection into components like perception of new information, memory usage, belief updating following surprise.

Agent Capabilities: **Memory**

Memory measures the ability to retain relevant information over **long, multi-turn interactions**, and is key for long-horizon tasks or long spanning conversational agents.

An agent's memory may be described by its **memory span**, or how long information is retained, and its **memory form**, which determines how memory is stored, such as in vectors or raw text.

Metric	Description
Factual Recall Accuracy	% of times the agent correctly recalls facts given after a set number of turns/context presented after.
Consistency Score	Stability across turns; does an agent respond consistently in long interactions?

Relevant Papers & Benchmarks

- **LongEval:** Evaluates on 40+ turn conversations.
- **SocialBench:** Assesses sociality of agents on and group levels.
- **Optimus-1:** Tracks memory state over hundreds of interactions.

What are we evaluating?

Core capability

- **Planning and reasoning**
- **Tool use**
- **Self-reflection**
- **Memory**

Applications

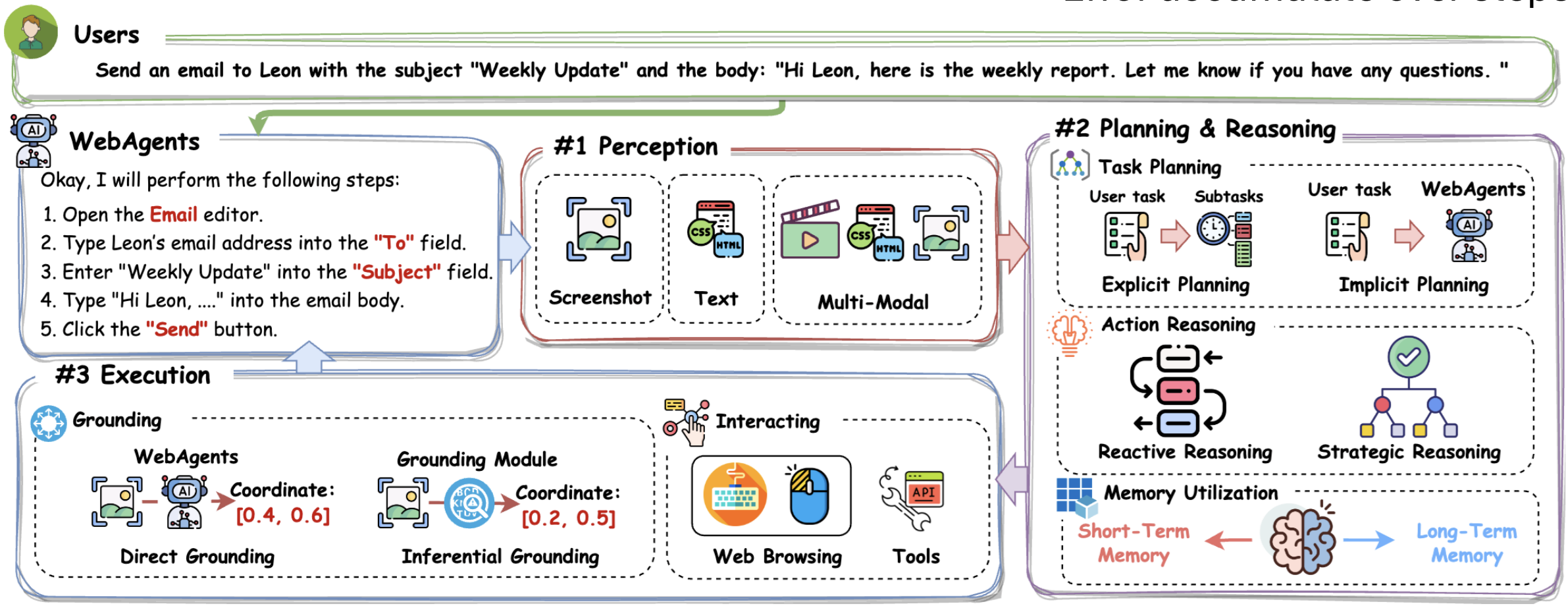
- Web Agent
- SWE Agent
- Scientific Agent
- Conversation Agent

Application: **Web agent**

An agent that perceive and act on the web.

Why hard:

- Live web change constantly
- Error accumulate over steps



Key metric:

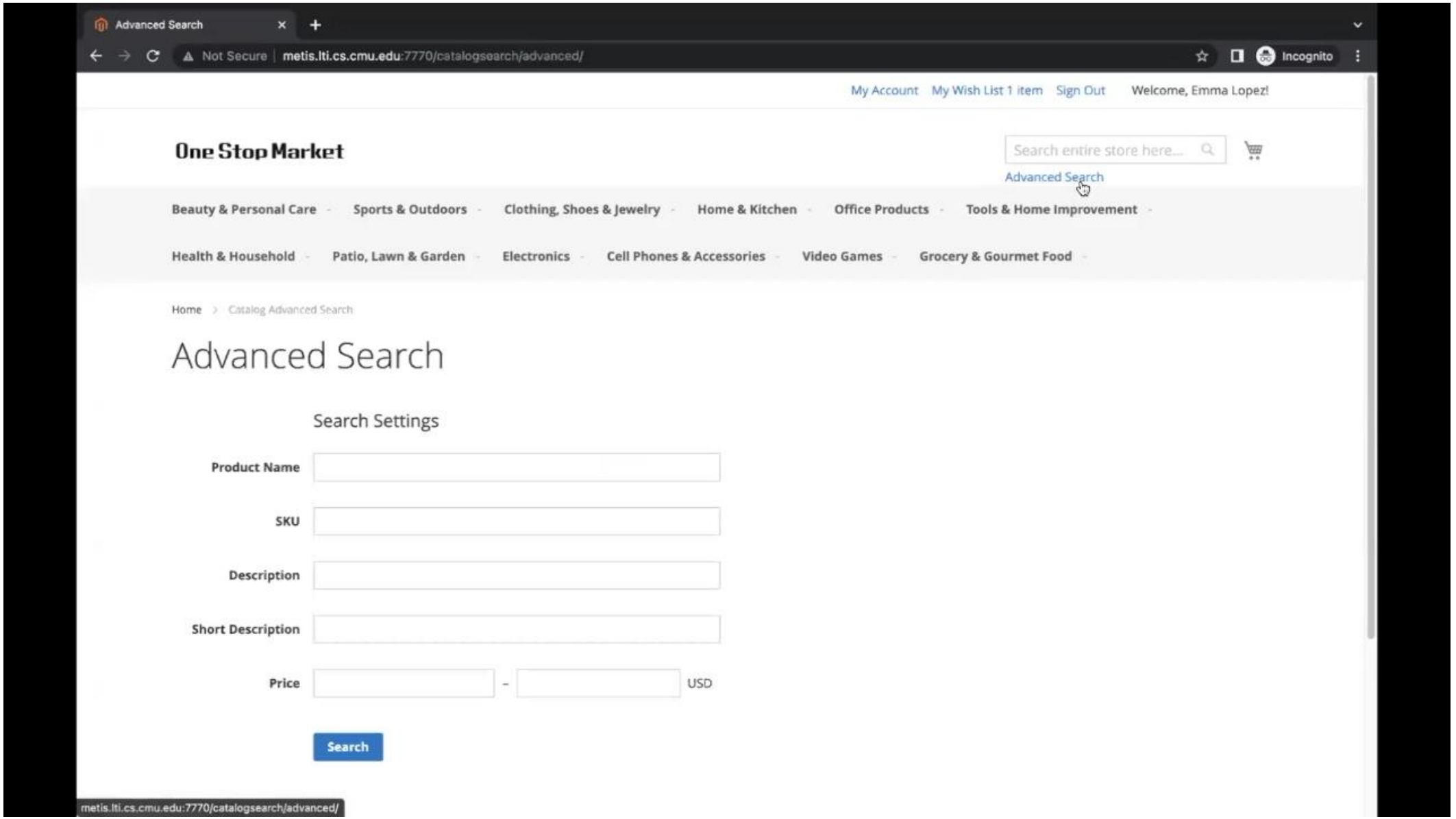
Success rate; Trajectory quality.

Benchmarks:

Mind2Wen; WebArena; WorkArena; ST-WebAgentBench

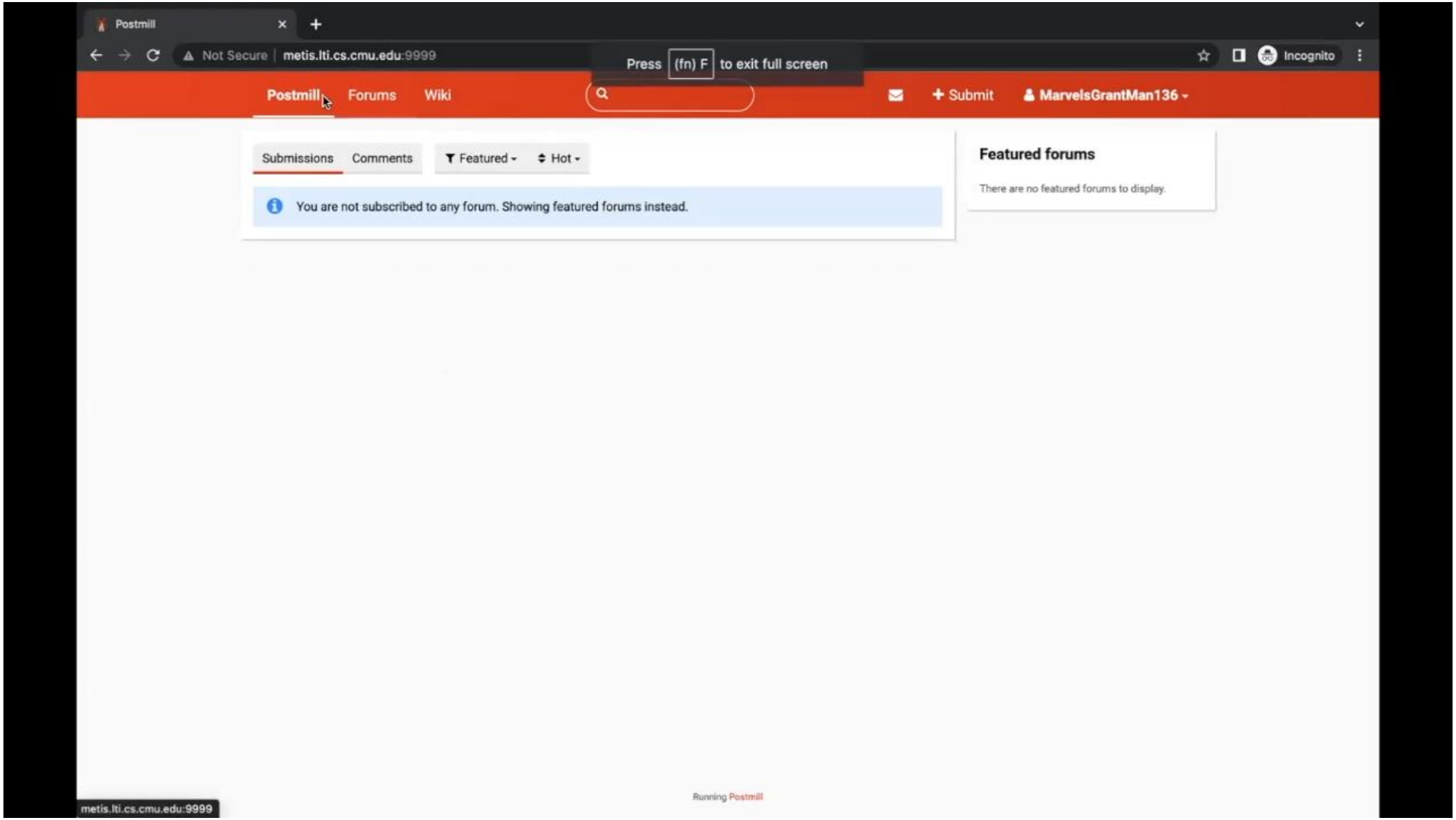
Application: **Web agent**

WebArena: Shopping



Application: **Web agent**

WebArena: Change Profile



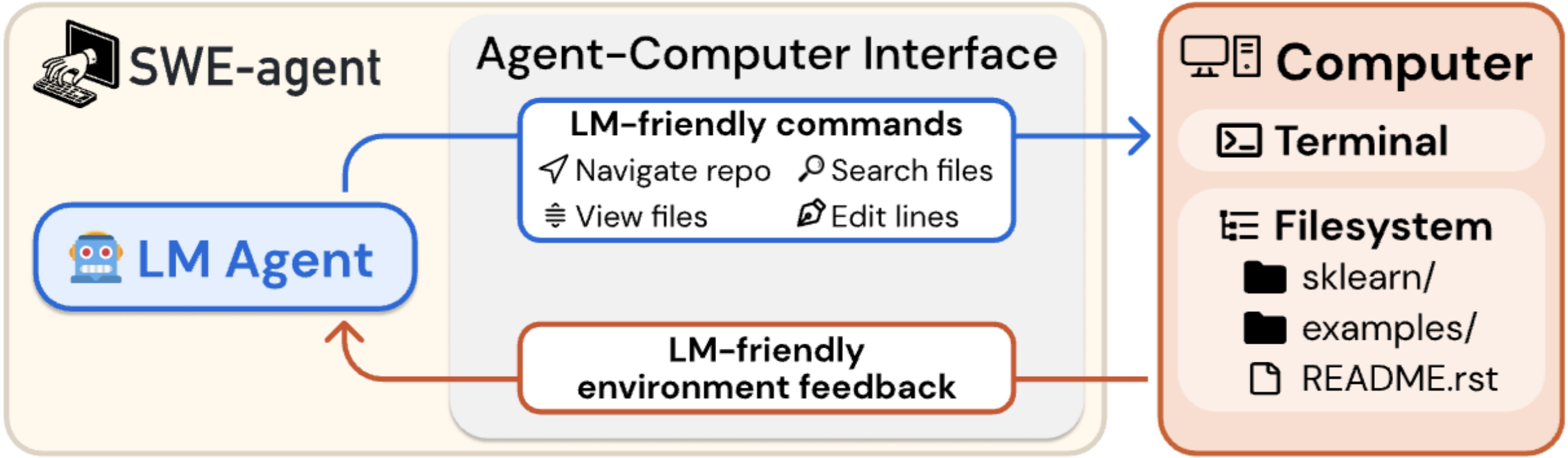
Application: **Web agent**

WebArena Leaderborad

	A	B	C	D	E	F	G	H
1	a	Open?	Model Size (billion)	Model	Success Rate (%)	Result Source	Work	Traj
2	02/2026	✔	-	Deepseek v3.2	74.3	WebTactix	WebTactix	Link
3	01/2026	✔	-	OpAgent	71.6	OpAgent	OpAgent	Link
4	12/2025	✔	-	ColorBrowserAgent	71.2	ColorBrowserAgent	ColorBrowserAgent	Link
5	10/2025	✔	-	Claude Code + GBOX MCP	68	GBOX AI	GBOX AI	Link
6	09/2025	✗	-	DeepSky Agent	66.9	Self-reported	DeepSky Agent	Link
7	10/2025	✗		Narada AI	64.2	Self-reported	Narada AI	Link
8	02/2025	✔	-	IBM CUGA	61.7	IBM CUGA	IBM CUGA	html+ json
9	01/2025	✗	-	OpenAI Operator	58.1	OpenAI CUA	OpenAI CUA	Link
10	08/2024	✗	-	Jace.AI	57.1	Reported by zetalabs.ai	https://www.jace.ai/	description + Scree
11	12/2025	✔	-	WebOperator + GPT-4o	54.6	WebOperator	WebOperator	Link
12	12/2024	✗	-	ScribeAgent + GPT-4o	53	ScribeAgent	ScribeAgent	Link
13	01/2025	✔	-	AgentSymbiotic	52.1	AgentSymbiotic	AgentSymbiotic	Link
14	01/2025	✔	-	Learn-by-Interact	48	Learn-by-interact	Learn-by-interact	Link
15	10/2024	✔	-	AgentOccam-Judge	45.7	AgentOccam-Judge	AgentOccam-Judge	Link
16	08/2024	✗	-	WebPilot	37.2	WebPilot	WebPilot	No op
17	10/2024	✔	-	GUI-API Hybrid Agent	35.8	Beyond Browsing	Beyond Browsing	Link
18	09/2024	✔	-	Agent Workflow Memory	35.5	AWM	AWM	
19	04/2024	✔	-	SteP	33.5	SteP	SteP	Link
20	06/2025	✔	12	TTI	26.1	TTI	TTI	Link

Application: **SWE agent**

An agent that read, write, fixes code through an agent-computer interface.



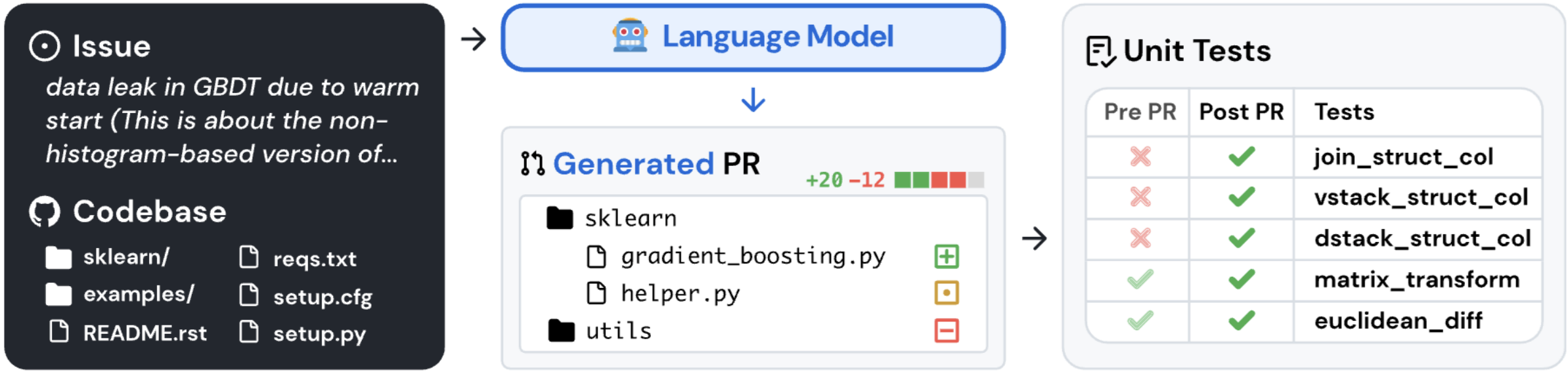
Why hard: Repo context could be huge; Many valid solution exist

Key metric:
% resolved; Test pass rate;
Edit efficiency.

Benchmarks:
SWE-Bench; SWE-bench verified; SWE-Bench Lite
LiveCodeBench/-Pro

Application: **SWE agent**

SWE-Bench



Tasks instances are obtained from real-world Python repositories by connecting GitHub issues to merged pull request solutions that resolve related tests

Application: **SWE agent**

SWE-Bench: Leaderboard

Verified | Multilingual | Lite | Full | Multimodal

Verified is a human-filtered subset of 500 instances. We use mini-SWE-agent to evaluate all models with the same harness (details).

Compare results | Agent: mini-SWE-agent v2 | Models: All models

Model	% Resolved	Avg. $	Trajs	Org	Date	Agent
Claude 4.5 Opus (high reasoning)	76.80	$0.75			2026-02-17	2.0.0
Gemini 3 Flash (high reasoning)	75.80	$0.36			2026-02-17	2.0.0
MiniMax M2.5 (high reasoning)	75.80	$0.07			2026-02-17	2.0.0
Claude Opus 4.6	75.60	$0.55			2026-02-17	2.0.0
GPT-5-2 Codex	72.80	$0.45			2026-02-19	2.0.0
GLM-5 (high reasoning)	72.80	$0.53			2026-02-17	2.0.0
GPT-5-2 (high reasoning)	72.80	$0.47			2026-02-17	2.0.0
GPT 5.2 Codex	72.80	$0.45			2026-02-19	2.0.0
Claude 4.5 Sonnet (high reasoning)	71.40	$0.66			2026-02-17	2.0.0

Application: **SWE agent**

LiveCodeBench: Leaderboard

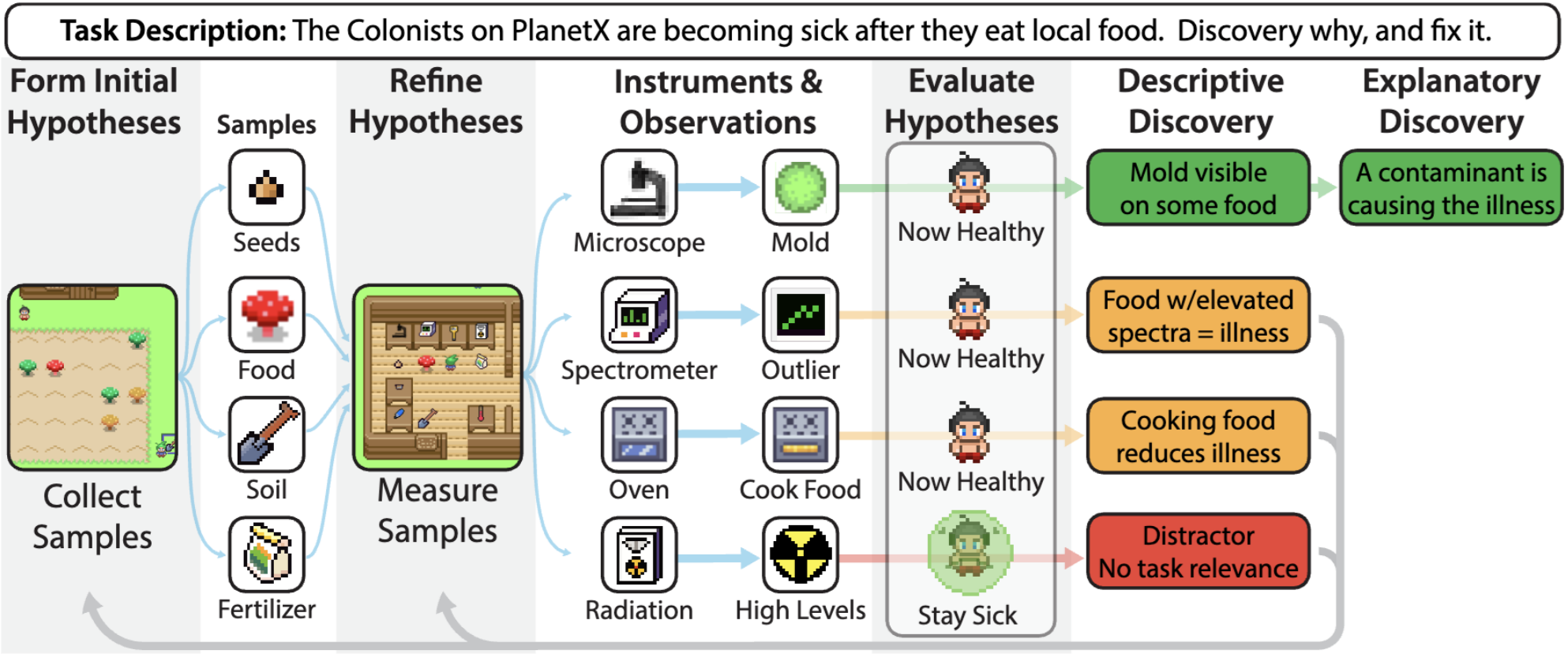
RANK	MODEL	PASS@1	EASY	MEDIUM	HARD
1	O4-Mini (High)	80.2	99.1	89.4	63.5
2	O3 (High)	75.8	99.1	84.4	57.1
3	O4-Mini (Medium)	74.2	98.2	86.5	52.7
4	Gemini-2.5-Pro-06-05	73.6	99.1	87.2	50.2
5	DeepSeek-R1-0528	73.1	98.7	85.2	50.7
6	Gemini-2.5-Pro-05-06	71.8	98.2	82.3	50.2
7	EXAONE-4.0-32B	70	98.4	82.3	46.2
8	OpenReasoning-Nemotron-32B	69.8	98.3	81.4	46.3
9	O3-Mini-2025-01-31 (High)	67.4	99.1	84.4	38.4
10	OpenCodeReasoning-Nemotron-1.1-32B	66.8	97.9	79.6	41.1

Application: **Scientific agent**

An agent that accelerate scientific research.

Why hard:

- Ground-truth rarely exist
- Need domain expertise



Key metric:

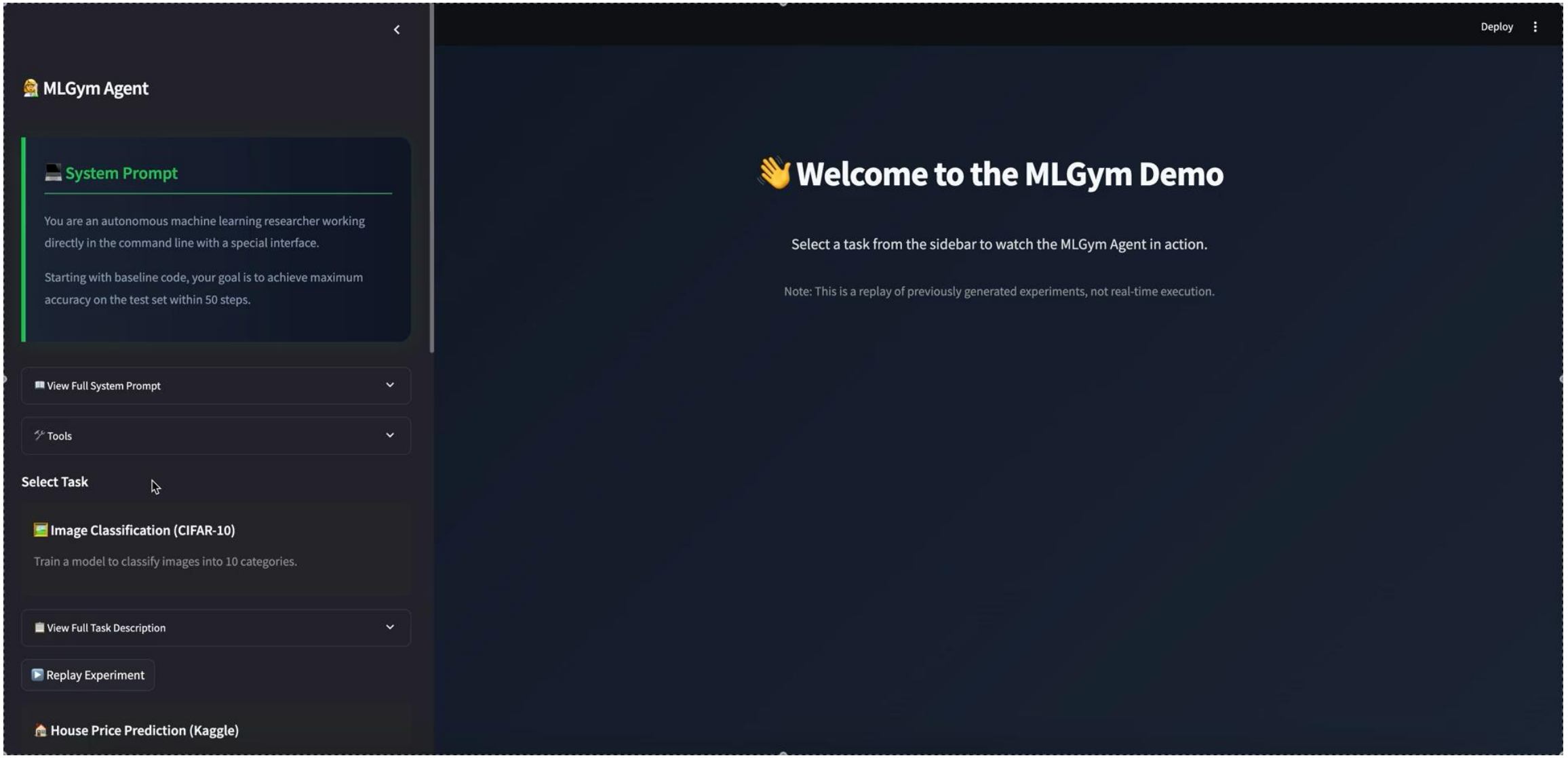
Scientific idea; Experiment design; Experiment code; Peer-review.

Benchmarks:

AAAR-1.0; ScienceAgentBench; CORE-Bench; DiscoveryWorld; MLGym.

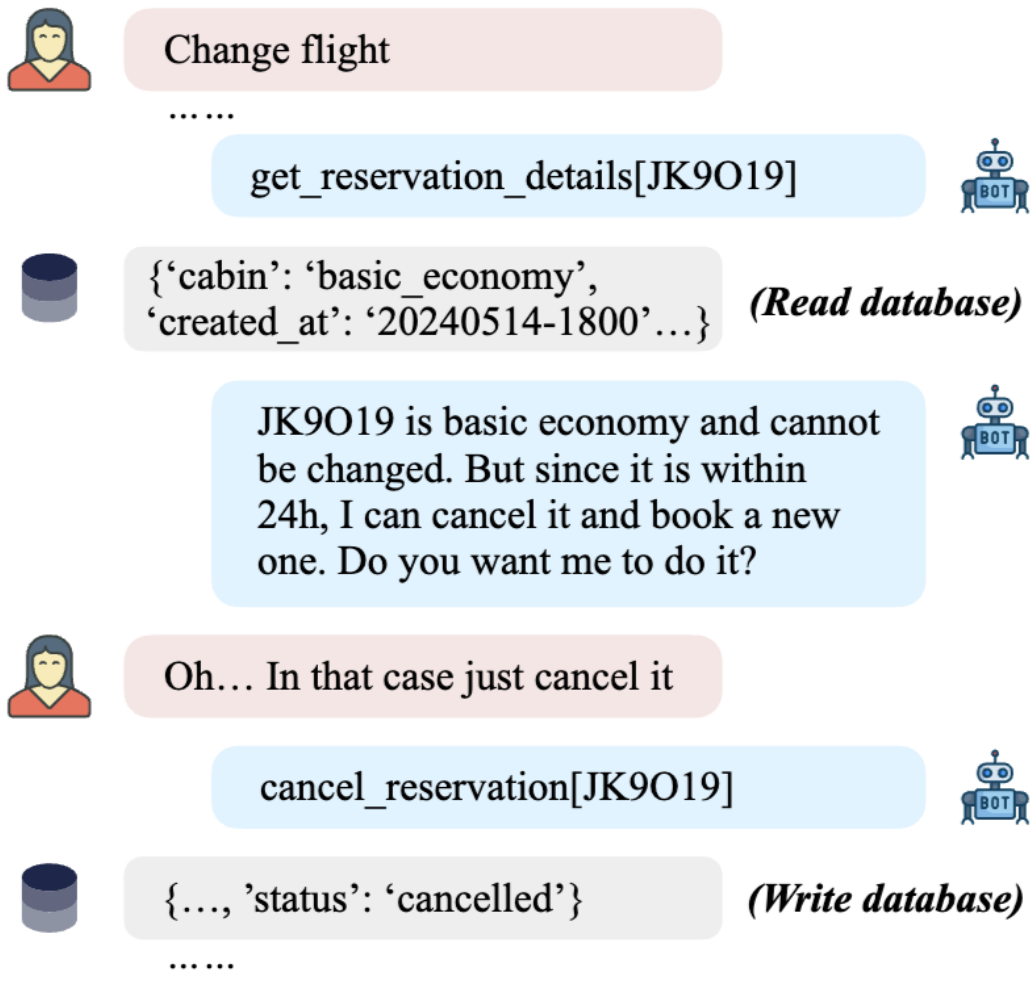
Application: Scientific agent

MLGym: Improve an RL task



Application: Conversation agent

Customer service agents are required to handle user requests, while adhering to the company's policies and procedures.



Why hard:

- Context span multi-turns
- No single correct answer

Key metric:

- Task completion
- User satisfaction
- Coherence

Benchmarks:

- ABCD
- ALMITA bench
- IntellAgent
- τ-Bench

Application: Conversation agent

τ-Bench

```
{"order_id": "#W2890441",
"user_id": "mei_davis_8935",
"items": [{
    "name": "Water Bottle",
    "product_id": "8310926033",
    "item_id": "2366567022",
    "price": 54.04,
    "options": {
        "capacity": "1000ml",
        "material": "stainless
        steel",
        "color": "blue"
    }}, ...], ...}
```

(a) An orders database entry in τ-retail.

```
def return_delivered_order_items(
    order_id: str,
    item_ids: List[str],
    payment_method_id: str,
) -> str: ...

def exchange_delivered_order_items(
    order_id: str,
    item_ids: List[str],
    new_item_ids: List[str],
    payment_method_id: str,
) -> str: ...
```

(b) An API tool in τ-retail.

```
## Return delivered order
- After user confirmation, the order status
will be changed to 'return requested'...

## Exchange delivered order
- An order can only be exchanged if its
status is 'delivered'...
```

(c) Domain policy excerpts in τ-retail.

```
{"instruction": "You are Mei Davis in 80217.
You want to return the water bottle, and
exchange the pet bed and office chair to the
cheapest version. Mention the two things
together. If you can only do one of the two
things, you prefer to do whatever saves you
most money, but you want to know the money
you can save in both ways. You are in debt
and sad today, but very brief.",
"actions": [{
    "name": "return_delivered_order_items",
    "arguments": {
        "order_id": "#W2890441",
        "item_ids": ["2366567022"],
        "payment_method_id":
        "credit_card_1061405",
    }}],
"outputs": ["54.04", "41.64"]}
```

(d) User instruction ensures only one possible outcome.

Model	retail	airline	avg
gpt-4o	**61.2**	**35.2**	**48.2**
gpt-4-turbo	57.7	32.4	45.1
gpt-4-32k	56.5	33.0	44.8
gpt-3.5-turbo	20.0	10.8	15.4
claude-3-opus	44.2	34.7	39.5
claude-3-sonnet	26.3	27.6	27.0
claude-3-haiku	19.0	14.4	16.7
gemini-1.5-pro	21.7	14.0	17.9
gemini-1.5-flash	17.4	26.0	21.7
mistral-large	30.7	22.4	26.6
mixtral-8x22b	17.7	31.6	24.7
meta-llama-3-70B	14.8	14.4	14.6

Table 2: Pass^1 across models via function calling, except Llama-3 via text-ReAct. Average is weighted by domains, not by tasks.

Future Directions

Holistic Evaluation Frameworks

- Most current evaluations target **single objectives** (e.g., tool use or behavior).
- Real-world agents must balance **multiple skills simultaneously** (e.g., safe, fast, accurate).
- Need for **multi-dimensional evaluations** integrating behavior, reasoning, and safety.

More Realistic Evaluation Settings

- Move beyond lab-style evaluations to **realistic enterprise environments**.
- Include **dynamic users**, **role-based access**, and **long-horizon workflows**.
- Simulated agents (e.g., in CRM, IT, finance systems) can help approximate production settings.

Scalable & Automated Evaluation Methods

- Manual evaluations are costly and limited.
- Push toward **LLM-as-a-judge**, **agent-as-a-judge**, and **synthetic data generation**.
- Reduce human overhead while preserving insight.

Time- and Cost-Bounded Protocols

- Repeated trials (e.g., pass^k) are expensive.
- Need **efficient evaluation pipelines** that balance depth and runtime.
- Useful for **evaluation-driven development** (EDD) in continuous deployment settings.

Thanks