Scan for Paper

GAIA2: BENCHMARKING LLM AGENTS ON *DYNAMIC* AND *ASYNCHRONOUS* ENVIRONMENTS

Romain Froger, Pierre Andrews, Matteo Bettini, Amar Budhiraja, Ricardo Silveira Cabral, Virginie Do, Emilien Garreau, Jean-Baptiste Gaya, Hugo Laurençon, Maxime Lecanu, Kunal Malkan, Dheeraj Mekala, Pierre Ménard, Gerard Moreno-Torres Bertran, Ulyana Piterbarg, Mikhail Plekhanov, Mathieu Rita, Andrey Rusakov, Vladislav Vorotilov, Mengjue Wang, Ian Yu, Amine Benhalloum, Grégoire Mialon, Thomas Scialom

Presented by: Romain Froger, PhD Student at Meta Superintelligence Labs
ICLR Oral 2026 - Friday 24 April

Meta

The "Sim-to-Real" Gap in Agent Evaluation

Current limitations

- Most benchmarks (e.g. GAIA, SWE-Bench, TerminalBench, Appworld, …) are **synchronous and agent-driven**.
- Environments are **strictly reactive** and lack independent temporal dynamics.
- Missing key elements to test real-world deployments failure modes (temporal reasoning, situational awareness, ...).

Dynamic Environments

- **Asynchronous & Event-driven**.
- Agents face **temporal constraints, background noise, ambiguity**, …

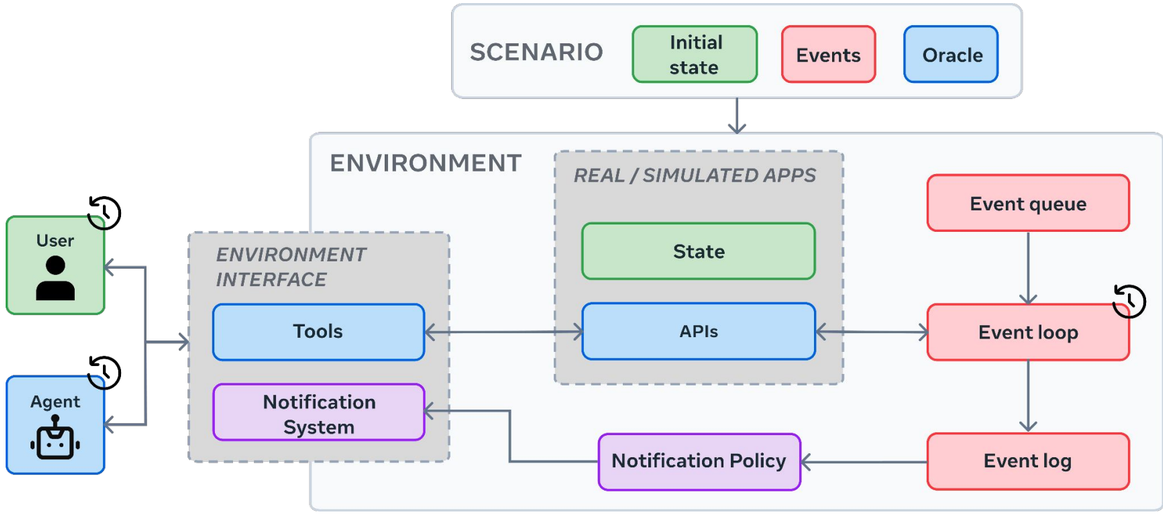
The Enabler: Agents Research Environments (ARE)

- A general-purpose **simulation platform** where time flows continuously and independently of the agent .
- ARE provides the foundation for **GAIA2: our new benchmark for General Agents in dynamic environments.**

Agents Research Environments

Core Abstractions:

1. **Apps**: Stateful APIs with read/write tools
2. **Environment:** Time-indexed collection of apps.
3. **Events**: The user receives an email from a friend as an external event, which notifies the Agent. **Everything is a an event**: logged into a single EventLog. Events are either “read” or “write” (modify the user space).
4. **Notifications**: Observability layer that pushes relevant events to the agent.
5. **Scenarios**: Dynamic tasks with verifiable goals.



GAIA2: 1,120 scenarios for Personal General Agents

Mobile Environment (ARE Instantiation)

12 apps • 101 tools • 10 universes

Contacts, Email, Calendar, Messages, Shopping, ...

- 400K–800K tokens per universe
- Persona-driven, cross-app consistent data

A Personal, API-Driven World

- **User-Centric Universes**: Each instance is a complete smartphone environment centered around a specific user.
- **Rich Personal Content:** Populated with cross-app consistent data (e.g., contacts match messaging history and calendar events).
- **Simple Agent Scaffold:** Agents run with a simple ReAct loop where they can perform a single tool call per turn

Capability	Example Task	Explanation
Execution	*Update all my contacts aged 24 or younger to be one year older than they are currently*	Evaluates the ability to chain long seq. of **`write` actions** in the right order
Search	*Which city do most of my friends live in? In case of a tie, return the first city alphabetically*	Evaluates the ability to chain long seq. of **read actions** in the right order
Ambiguity	*Schedule a 1h Yoga event each day at 6:00 PM from October 16, 2024 to October 21, 2024*	Tests whether agents ask for **clarification** on impossible, contradictory, or ambiguous tasks
Adaptability	*I have to meet my friend Kaida to view a property [...] If she replies to suggest another property or time, update the calendar event*	Requires agents to **adapt dynamically** to environmental changes
Time	*Send messages to each of the colleagues I am supposed to meet today, asking who is supposed to order the cab. If after 3 minutes there is no response, order a cab from [...]*	Evaluates whether agents can **complete tasks in due time** & maintain temporal awareness
Agent2Agent	**Same Search task as above but the Contacts and Chats apps are replaced by app sub-agents**	Tests whether agents can **collaborate with other agents** to use tools & complete tasks
Noise	**Same Adaptability task as above but with random tool execution errors and random environment events occurring during execution**	Evaluates whether agents are **robust to environment noise & distractors**

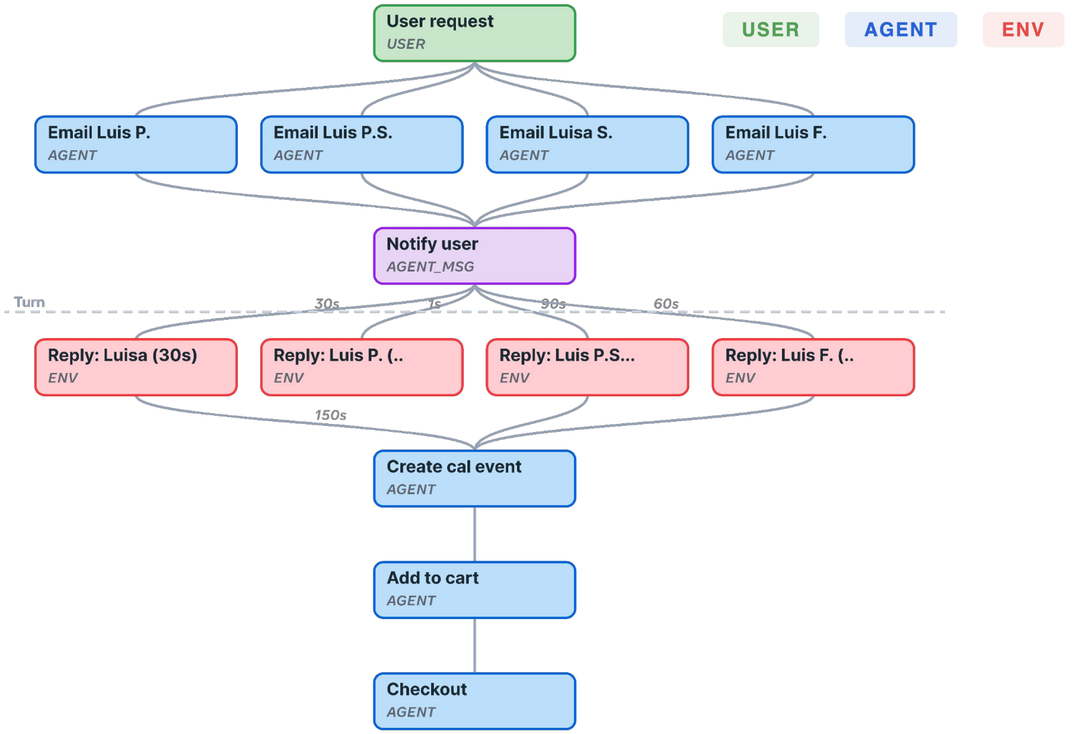
Scenarios as DAGs of Events

Scenario annotations **can** be time-indexed to schedule events at specific timestamps.

Time Scenario Task:
*"Email my Friends to **ask for their availability**, and **wait for their reply** to choose a Party time slot that fits in everyone's calendar"*

Environment:
Friends reply to the email with different delays

Challenge:
Capture email ***notifications, wait*** for all replies and then ***book accordingly***.



Time Scenario DAG are time-indexed

Anatomy of the Grader

RQ1: What do we grade?

- Agents can do unlimited read actions (exploration).
- Only write actions (environment-altering) are graded using the `EventLog`.

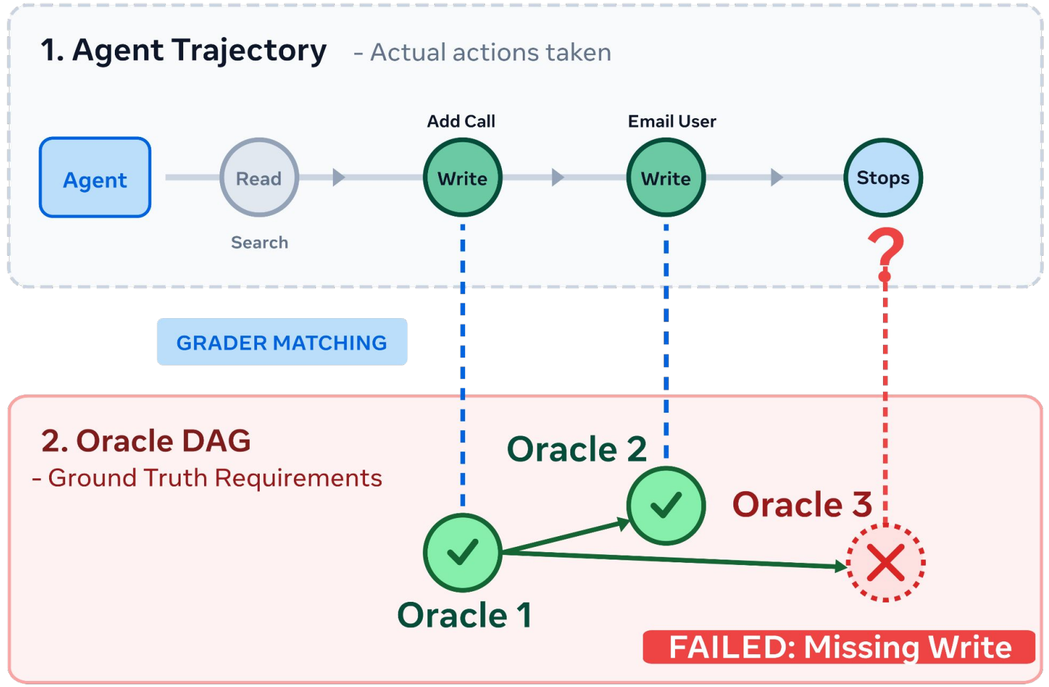
RQ2: How to grade long trajectories?

- Remap the linear `EventLog` to the DAG Oracle annotations. 4 pillars of the grader:
  - **Consistency**: Exact match / LLM rubrics
  - **Causality**: Enforce DAG dependency
  - **Timing**: Tolerance windows for timely actions
  - **Completeness**: Matching all Oracle actions

RQ3: Can we beat a baseline LLM-as-Judge?

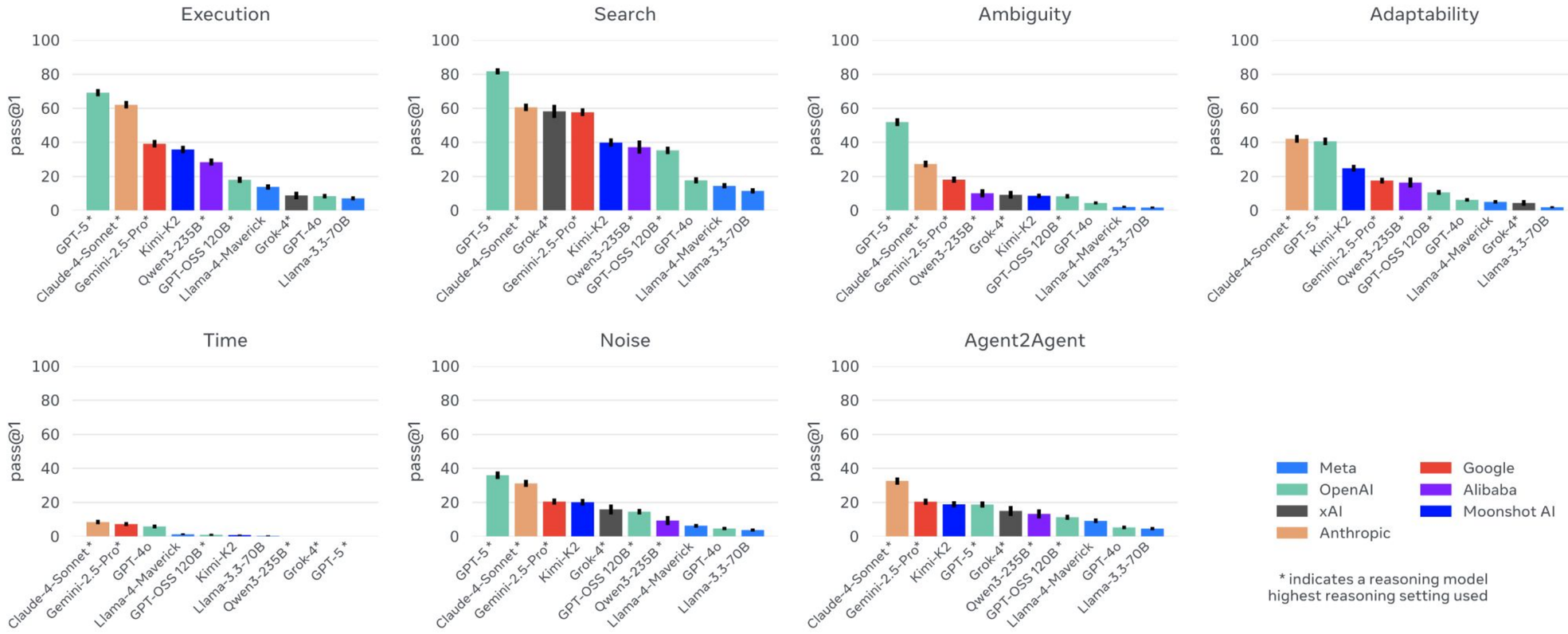
- We compared our DAG-based grader with a pure LLM-as-Judge on a held-out trace set.
- Result: The DAG structure adds significant robustness, preventing context-overload and hallucinations.

GRADER MODEL	AGREEMENT	PRECISION	RECALL
LLM Judge	0.72	0.53	0.83
✓ ARE Grader (Ours)	0.98	0.99	0.95



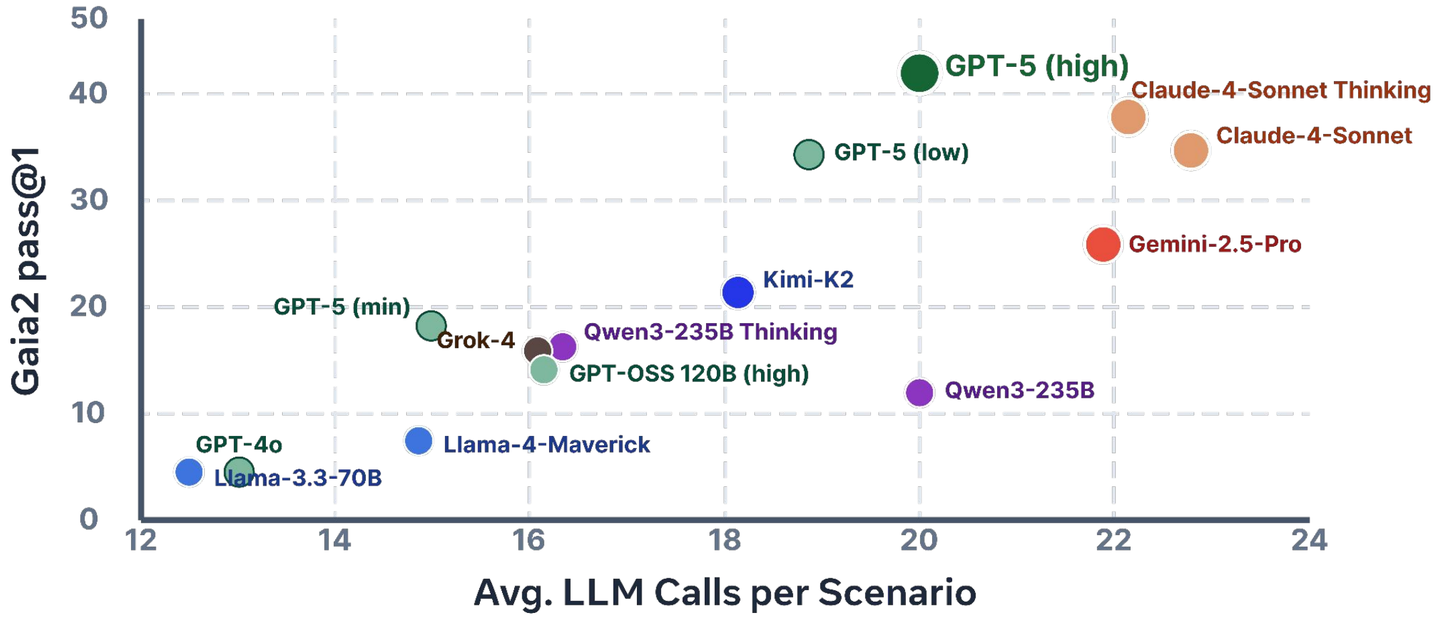
Main Results

1. Frontier closed-source models dominate across the board.
2. Execution and Search are almost already saturated.
3. Room for improvement on all other capabilities, required in real Agents deployment.



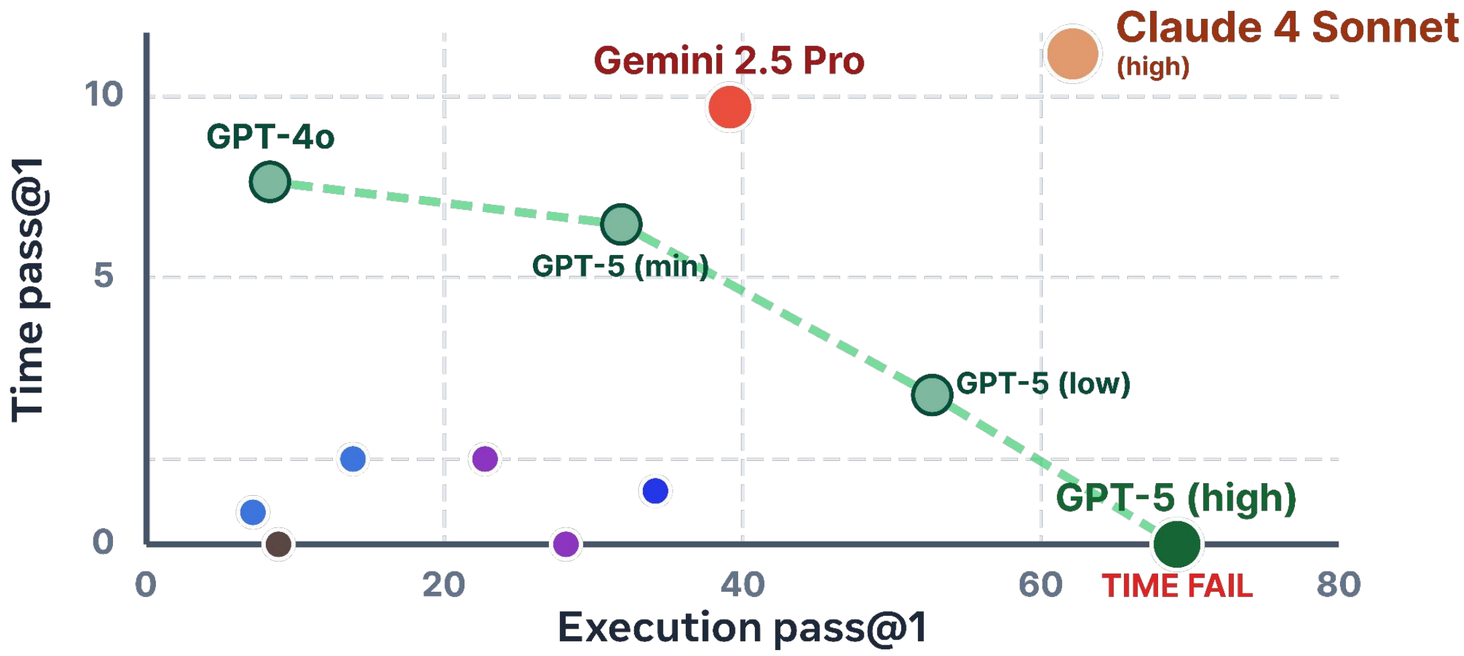
Deeper Search Drives Performance

- **Scale with Effort:** Overall performance scales linearly with the number of average tool calls per scenario.
- **The "Give Up" Factor:** Weaker models tend to stop after just a few failed calls.
- **Read Before You Write:** Strong models (e.g., GPT-5, Claude-4-Sonnet) execute significantly more read calls before making their first write action.
- **Takeaway:** Slow and steady wins the trace.



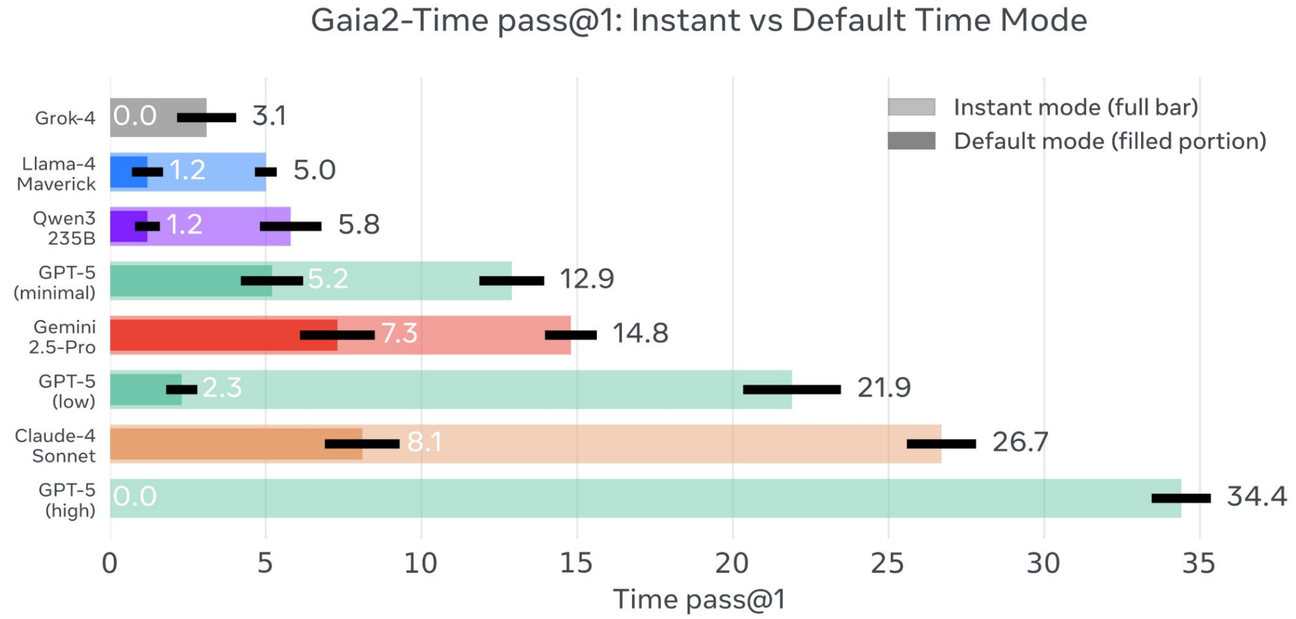
The "Inverse Scaling" Trap on Time Tasks

- **The Deliberation Cost:** Top models use extensive compute during inference, resulting in slow execution.
- **The Penalty:** Heavy reasoning models (like the GPT-5 family) frequently miss deadlines.
- **Takeaway:** An observed inverse scaling law; high execution pass@1 correlates with high failure rates on strict temporal tasks.



The Need for Adaptive Compute

- **The "Instant Time" Ablation:** By freezing time during agent inference in ARE, we isolate the impact of reasoning speed.
- **Capability vs. Speed:** Models have the reasoning capability to solve time-aware tasks (cf.'Instant' mode), but fail in real-time due to slow inference.
- **Takeaway:** Future agents require adaptive compute; fast generation for urgent/routine tasks, and deep reasoning only when the environment permits.



Conclusion

Key Takeaways:

- **The "Sim2Real" Gap is Real:** No single model dominates. SOTA (GPT-5) caps at 42% pass@1.
- **Crucial Trade-offs:** Thinking models perform well when there is no time deadline.
- **Action-Level Verification:** End-state checks are insufficient. The ARE Verifier ensures highly accurate verification and prevents approving models with *destroy and repair* behaviors.

What's Next:

- **ARE and GAIA2 provide the open infrastructure** needed to train the next generation of personal agents à la OpenClaw.

Read the full paper

Code & Datasets

References

[1] Mialon, G., Fourrier, C., Wolf, T., LeCun, Y., & Scialom, T.. Gaia: a benchmark for general ai assistants, 2023.

[2] Merrill, M. A., Shaw, A. G., Carlini, N., Li, B., Raj, H., Bercovich, I., ... & Schmidt, L. (2026). Terminal-bench: Benchmarking agents on hard, realistic tasks in command line interfaces. arXiv preprint arXiv:2601.11868.

[3] Jimenez, C. E., Yang, J., Wettig, A., Yao, S., Pei, K., Press, O., & Narasimhan, K. (2023). Swe-bench: Can language models resolve real-world github issues?. arXiv preprint arXiv:2310.06770.

[4] Trivedi, H., Khot, T., Hartmann, M., Manku, R., Dong, V., Li, E., ... & Balasubramanian, N. (2024, August). Appworld: A controllable world of apps and people for benchmarking interactive coding agents. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 16022-16076).