The Second Half in AI Agents: Models and Beyond

@Salesforce AI Research

Presenter: Zhiwei Liu
Senior Research Scientist

First-half

* Work from Salesforce

- **Benchmarking/Environment**
  - BOLAA*, AgentBench, AgentBoard
- **Model Training**
  - xLAM*, AgentOhana*, APIGen* AgentTuning
- **Library** - how to build an agent?
  - AgentLite*, Llama_index, langchain, CrewAI, AutoGen, etc
- **Reasoning/Planning** - how to conduct next actions?
  - ReAct, Reflexion, Divergent Thinking*
- **Memory** - how can we leverage the past trajectory of agent
  - MemGPT, Mem0
- **Agent Prompt Optimizing**
  - PRAct*, TextGrad
- **Prototype applications**
  - AutoGPT, MetaGPT, etc

1. ReAct: Synergizing Reasoning and Acting in Language Models, ICLR 2023
2. Reflexion: Language Agents with Verbal Reinforcement Learning, NeurIPS 2023
3. DRDT: Dynamic Reflection with Divergent Thinking for LLM-based Sequential Recommendation, arxiv 2023
4. BOLAA: Benchmarking and Orchestrating LLM-augmented Autonomous Agents, LLMAgents@ICLR 2024
5. AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation Framework, arxiv 2023
6. AgentBench: Evaluating LLMs as Agents, arxiv 2023
7. AgentBoard: An Analytical Evaluation Board of Multi-turn LLM Agents, arxiv 2024
8. AgentTuning: Enabling Generalized Agent Abilities for LLMs, arxiv 2023
9. AgentOhana: Design Unified Data and Training Pipeline for Effective Agent Learning
10. AgentLite: A Lightweight Library for Building and Advancing Task-Oriented LLM Agent System
11. MemGPT: Towards LLMs as Operating Systems
12. APIGen: Automated Pipeline for Generating Verifiable and Diverse Function-Calling Datasets

Current

- Agent Libraries are already well developed
  - Langchain, llama_index, or simple function call message loops

Current

- Agent Libraries are already well developed
- LLMs support function-call by default
  - Api-based: gpt series, gemini, claude, etc
  - Open-source: https://python.langchain.com/docs/integrations/chat/
  - Serving: vLLM tool calling, SGLang tool and function calling
-

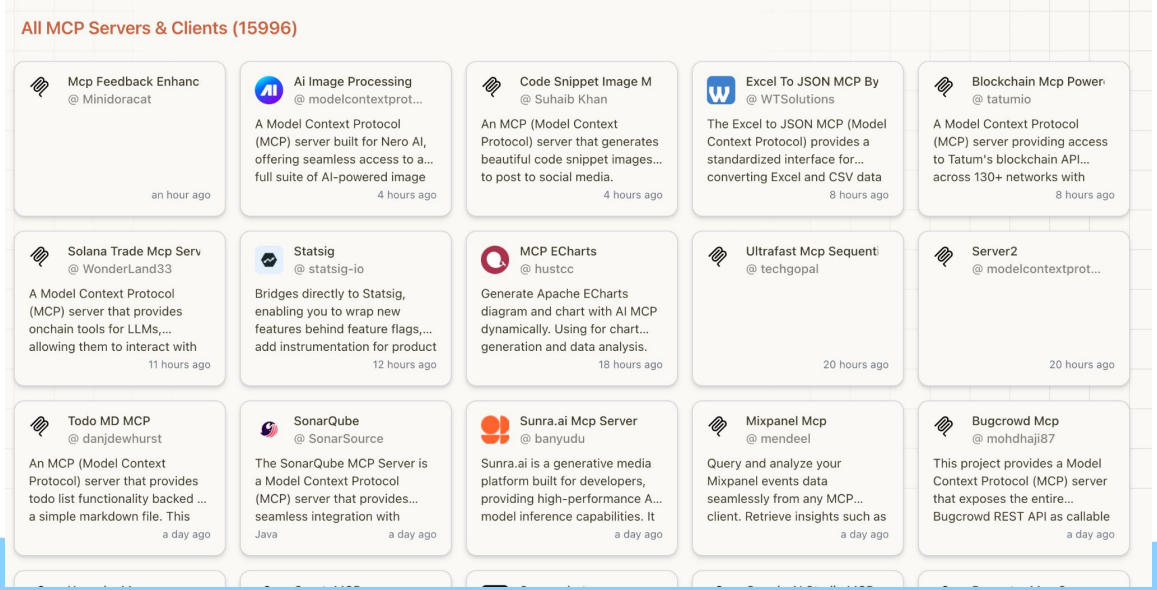
Provider	Tool calling	Structured output	JSON mode	Local	Multimodal	Package
ChatAnthropic	✅	✅	❌	❌	✅	langchain-anthropic
ChatMistralAI	✅	✅	❌	❌	❌	langchain-mistralai
ChatFireworks	✅	✅	✅	❌	❌	langchain-fireworks
AzureChatOpenAI	✅	✅	✅	❌	✅	langchain-openai
ChatOpenAI	✅	✅	✅	❌	✅	langchain-openai
ChatTogether	✅	✅	✅	❌	❌	langchain-together
ChatVertexAI	✅	✅	❌	❌	✅	langchain-google-vertexai
ChatGoogleGenerativeAI	✅	✅	❌	❌	✅	langchain-google-genai
ChatGroq	✅	✅	✅	❌	❌	langchain-groq
ChatCohere	✅	✅	❌	❌	❌	langchain-cohere

All chat models

Name	Description
Abso	This will help you get started with ChatAbso chat models. For detaile...
AI21 Labs	This notebook covers how to get started with AI21 chat models.
Alibaba Cloud PAI EAS	Alibaba Cloud PAI (Platform for AI) is a lightweight and cost-efficie...
Anthropic	This notebook provides a quick overview for getting started with Anth...
Anyscale	This notebook demonstrates the use of langchain.chat_models.ChatAnysc...
AzureAIChatCompletionsModel	This will help you get started with AzureAIChatCompletionsModel chat ...
Azure OpenAI	This guide will help you get started with AzureOpenAI chat models. Fo...
Azure ML Endpoint	Azure Machine Learning is a platform used to build, train, and deploy...
Baichuan Chat	Baichuan chat models API by Baichuan Intelligent Technology. For more...

Current

- Agent Libraries are already well developed
- LLMs support function-call by default
- [Model context protocol (MCP)](#) unifies the model/tool interaction protocol



Current

- Agent Libraries are already well developed
- LLMs support function-call by default
- Model context protocol (MCP) unifies the model/tool interaction protocol
- Real-world applications success
  - Deep research
  - Manus, cursor, claude code

Current

- Agent Libraries are already well developed
- LLMs support function-call by default
- Model context protocol (MCP) unifies the model/tool interaction protocol
- Real-world applications success
  - Deep research
  - Manus, cursor, claude code

Shifting from *assembling components* to **engineering intelligent, deployed systems**.

The Second Half

- **The Evolving Engine**: Specialization of core models for reasoning and tool use.
- **Agents in the Wild**: Real-world products and the realistic benchmarks that evaluate them.
- **The Next Horizon**: The future of human-agent interaction and agent cognition.

Part 1: The Evolving Engine

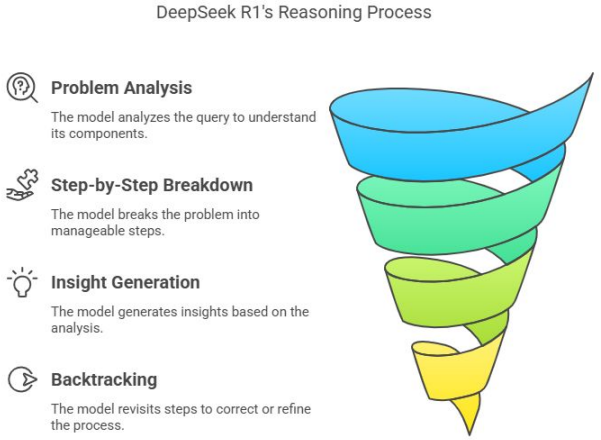
The Evolving Engine

New Frontiers in Agent Models

- The core intelligence of an agent is moving beyond a single, general-purpose LLM.
- This evolution is driven by three key advancements:
  - The rise of specialized Reasoning Models .
  - The increasing sophistication of Function Calling for tool use.
  - The revolution in feedback loops.

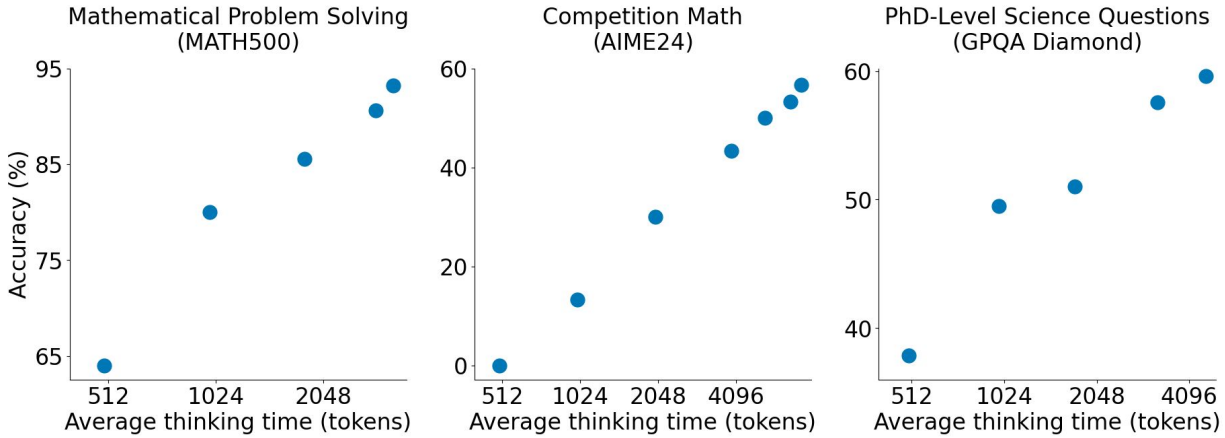
The Rise of Reasoning Models

- A new distinct category of AI models emerges and adopted quickly
  - OpenAI's o-series,
  - Google gemini,
  - Claude
  - DeepSeek-r1
- Optimized for multi-step logical reasoning and complex problem-solving .



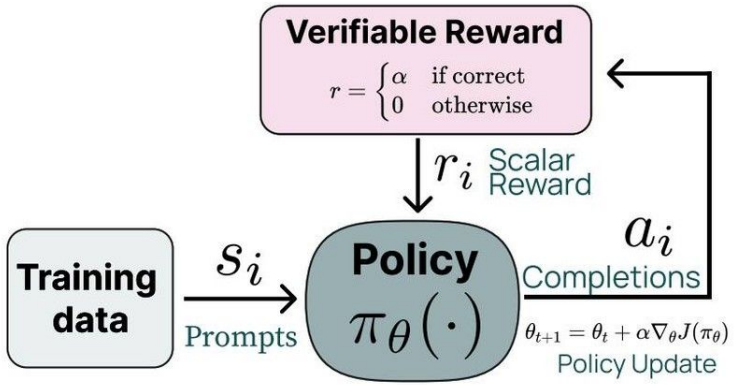
How They Achieve Superior Reasoning

- Increased Test-time Scaling
  - More computational power is dedicated at the time of the query to analyze, explore, and verify solutions.



How They Achieve Superior Reasoning

- Increased Test-time Scaling
  - More computational power is dedicated at the time of the query to analyze, explore, and verify solutions.
- Specialized Post-Training
  - Techniques like Reinforcement Learning with Verifiable Rewards (RLVR) explicitly train for structured reasoning.

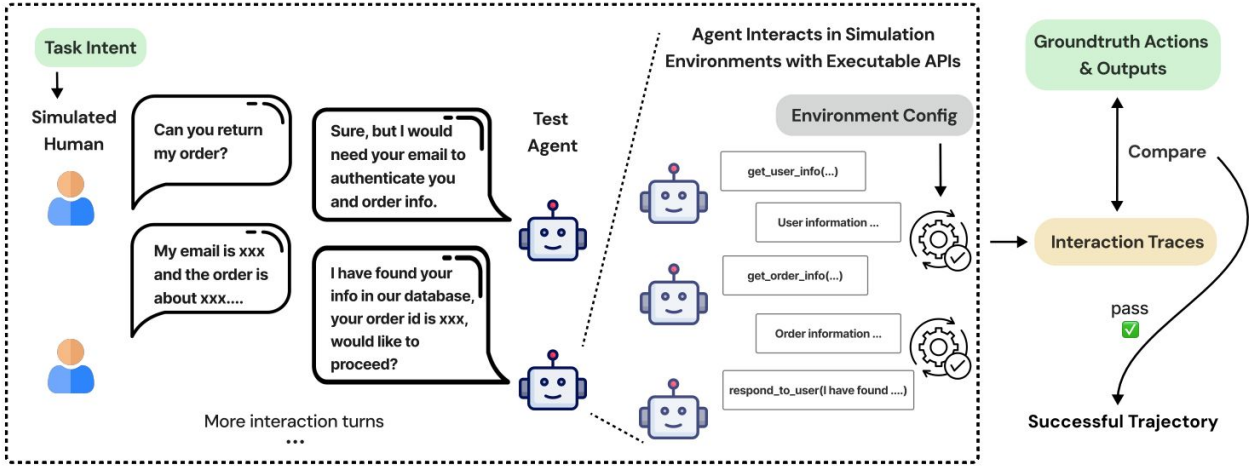


How They Achieve Superior Reasoning

- Increased Test-time Scaling
  - More computational power is dedicated at the time of the query to analyze, explore, and verify solutions.
- Specialized Post-Training
  - Techniques like Reinforcement Learning with Verifiable Rewards (RLVR) explicitly train for structured reasoning.
- Key Implication
  - Future breakthroughs will come from more sophisticated training and inference algorithms, not just scaling model size.

The Sophistication of Function Calling

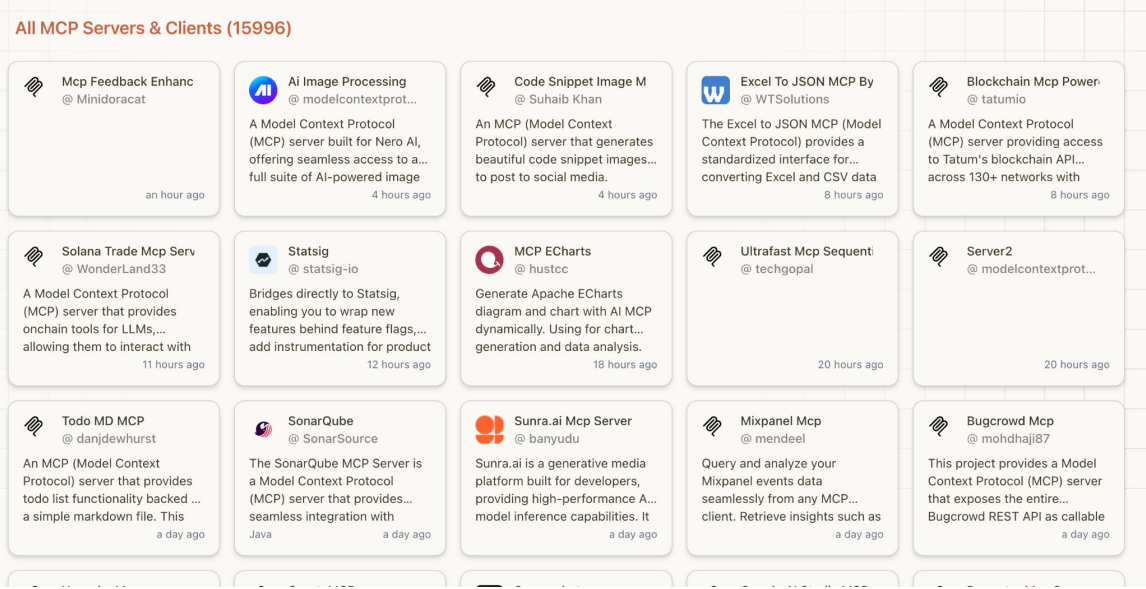
- Parallel multi-turn function calling, especially long steps



API-gen MT

The Sophistication of Function Calling

- Parallel multi-turn function calling, especially long steps
- Though unified protocol, select correct tools is not easy

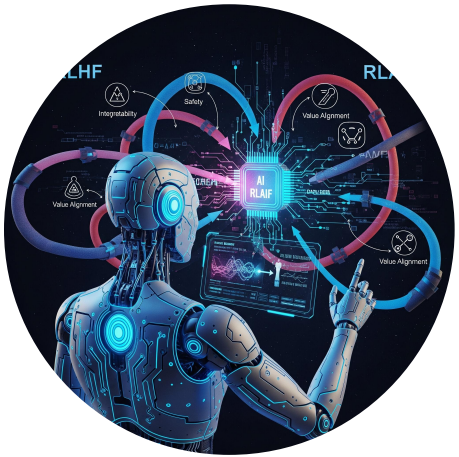


API-gen MT

The Feedback Loop Revolution

- The Old Way (RLHF): Reinforcement Learning from Human Feedback
  - slow, expensive, and can embed the biases of its human evaluators .

- The New Way (RLAIF): Reinforcement Learning from AI Feedback
  - replaces the human with an AI labeler guided by a predefined set of principles or a "Constitution"



The Feedback Loop Revolution

- The New Way (RLAIF): Reinforcement Learning from AI Feedback
  -
  - Scalability & Cost: It enables alignment at an industrial scale and is estimated to be over 10x cheaper than RLHF .
  -
  - Performance: Achieves results on par with, or even superior to, RLHF on complex tasks .
  -
  - Robustness: Creates a more consistent, principle-based foundation, which is critical for deploying safe and reliable autonomous agents.
  -
  -

Part 2: Agents in the Wild

Case Study

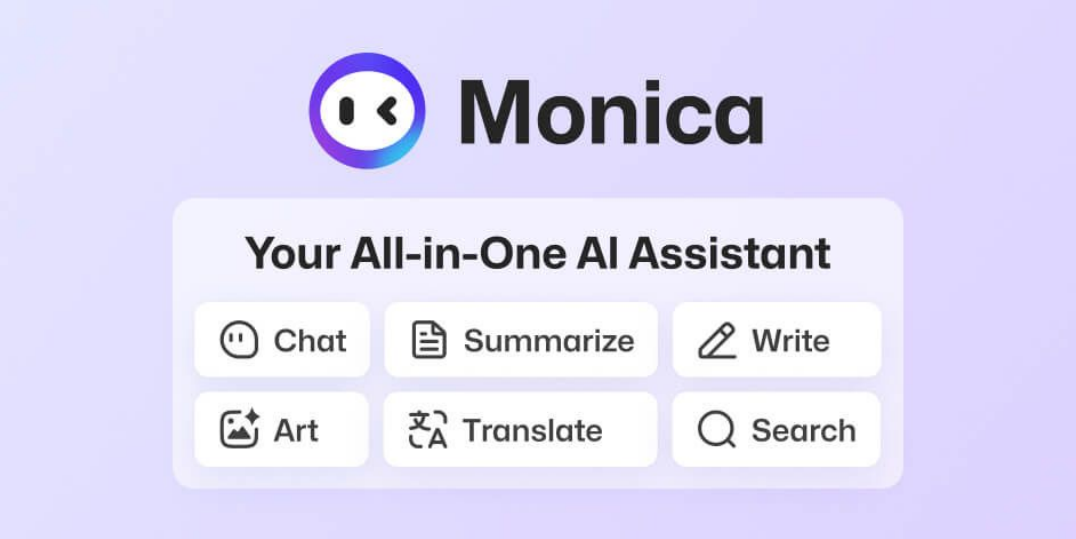
- OpenAI's Deep Research

Case Study

- **OpenAI's Deep Research**
  - Multi-step autonomous research
  - Built on OpenAI's o3 reasoning model for enhanced analytical capabilities
- How does it work
  - Prompt-driven: Simply describe what you need researched
  - *Autonomous operation*: Agent works independently without constant guidance
  - Web crawling: Searches across public web data sources
  - Analysis & synthesis: Processes and combines information intelligently
  - Report generation: Creates detailed, cited reports at research analyst quality

Case Study

- Monica's Manus AI

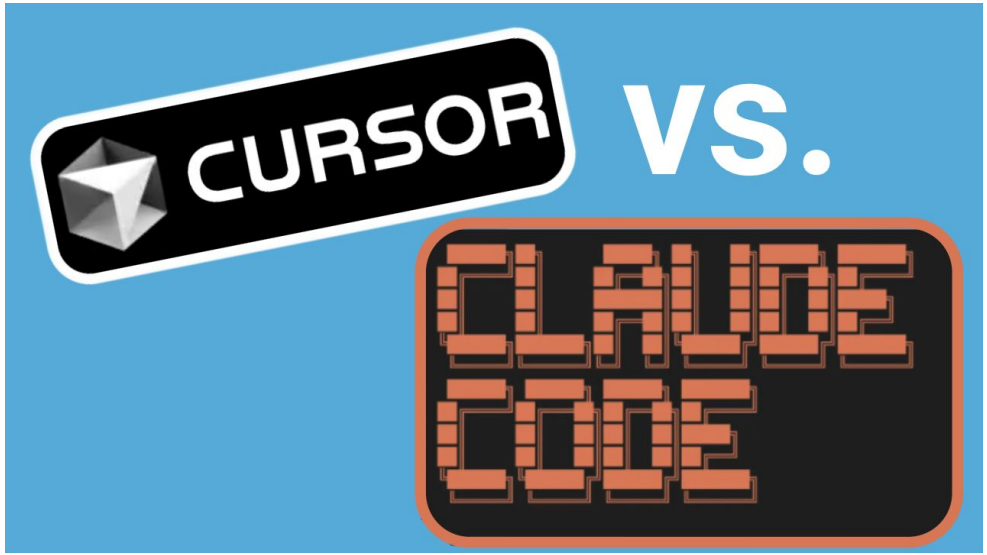


Case Study

- Monica's Manus AI
  - Architecture: A single, highly capable model (Claude 3.7 Sonnet) equipped with a versatile suite of 29+ specialized tools .
  -
  - Philosophy: A single, powerful "polymath" agent can solve the problem if given a powerful enough toolkit.
  -
  - Strengths: Flexibility, speed, and state-of-the-art performance on the GAIA benchmark, outperforming competitors .

Case Study

- Cursor and Claude Code



Benchmarking in Realistic Environments

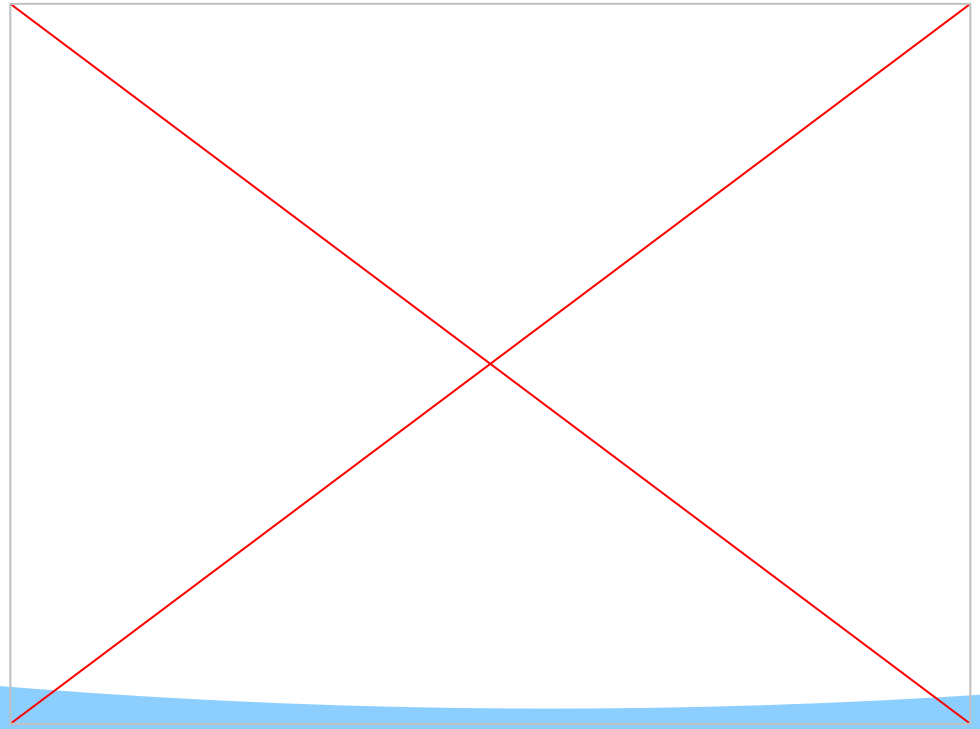
- GAIA (General AI Assistants)
  - What it tests: Reasoning, multi-modality, web browsing, and tool proficiency .
  -
  - Key Failure Modes: "Access Issues," "Misinterpretation of Requirements," and "Data Extraction Challenges," pointing to weaknesses in tool robustness and planning

Benchmarking in Realistic Environments

- GAIA (General AI Assistants)
- SWE-bench (Software Engineering Benchmark):
  - What it tests: Autonomously resolving real-world GitHub issues, requiring deep codebase navigation and reasoning .
  -
  - Key Failure Modes: "Inadequate Testing," "Context Loss," and "Dependency Oversight," highlighting challenges in memory and maintaining an accurate mental model of the environment .
  -
-

MCPEval

- https://github.com/SalesforceAIResearch/MCPEval



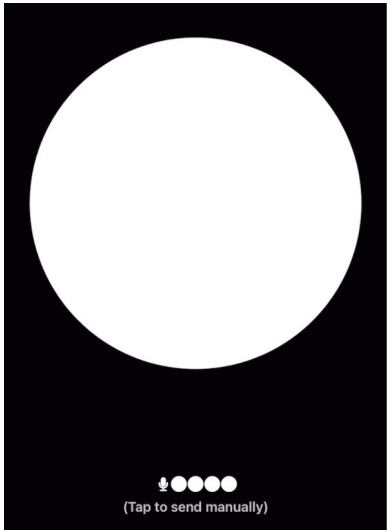
Frontier AI agents

- Large complex systems
- Requires powerful model capability in both reasoning and tool calling

Part 3 - The Next Horizon

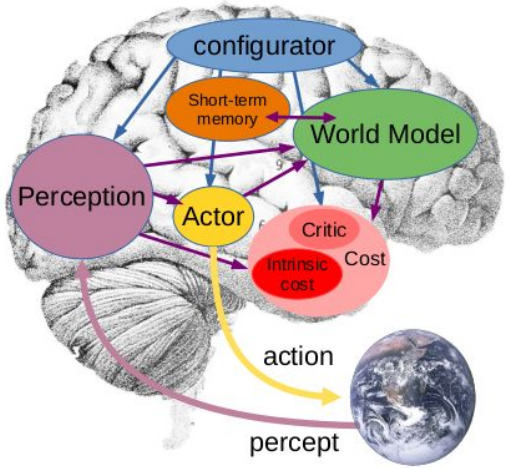
Two interconnected frontiers

- human-agent interface
  - Moving from text to seamless, real-time voice and embodied interaction.
  - Coding
  - Everyday Context
- OpenAI voice agent
- Gemini multi-modal agent
- Robotics RT-2



Two interconnected frontiers

- human-agent interface
- agent's cognitive architecture
  - From simple memory to predictive world models
  - An internal, learned, predictive model of an environment. It allows an agent to run internal "what if?" simulations to predict the future consequences of its actions .
- Why are they a "Necessary Ingredient"?
  - any agent capable of generalizing to novel, multi-step tasks must have a predictive world model.
  - An agent's performance and the complexity of goals it can achieve are directly proportional to the accuracy of its internal world model.



Beyond Performance

Safety and Explainability

- Reverse World Models are a new research direction for creating powerful counterfactual explanations .

- Instead of asking "What will happen if I do X?", they can answer, "What would the world have needed to look like for me to have done Y?", providing deep insight into an agent's decision-making .

Conclusion

- Evolving Engines: From general LLMs to specialized Reasoning Models, advanced Function Calling, and scalable RLAIF alignment.

- Agents in the Wild: A clear architectural debate (Multi-Agent vs. Monolithic) and the rise of realistic benchmarks (GAIA, SWE-bench) that provide a roadmap via failure analysis.

- The Next Horizon: A future defined by real-time voice interaction and the cognitive leap enabled by world models.

Open Research Questions

- Architecture & Efficiency: Will powerful monolithic models plus tooling make complex multi-agent pipelines obsolete?
- Evaluation & Reliability: How can we build agents that are fundamentally robust to real-world friction like context loss and unreliable tools?
- Safety & Control: How do we verify that an agent's internal world model is accurate and aligned with human values, and what new safety methods are needed for these increasingly autonomous systems?

Thanks!