Working with AI Agents

Ruoqing Zhu

Nov 6, 2026, @SDSA

Disclaimer

- **All my opinions are subjective**
- **I am not an expert on this**
- **I do not guarantee that my approach would work**
- **I am not responsible for your learning**
- **Some of the examples can be expensive...**

What I want to focus on:

- Create skills
- Build Workflow
- Manage agents
- Build harness?
- The engineering thinking

Basic understandings

Overview, build skills, workflow, some tricks, etc.

To explore something fun

How to build complex skills, or even agents

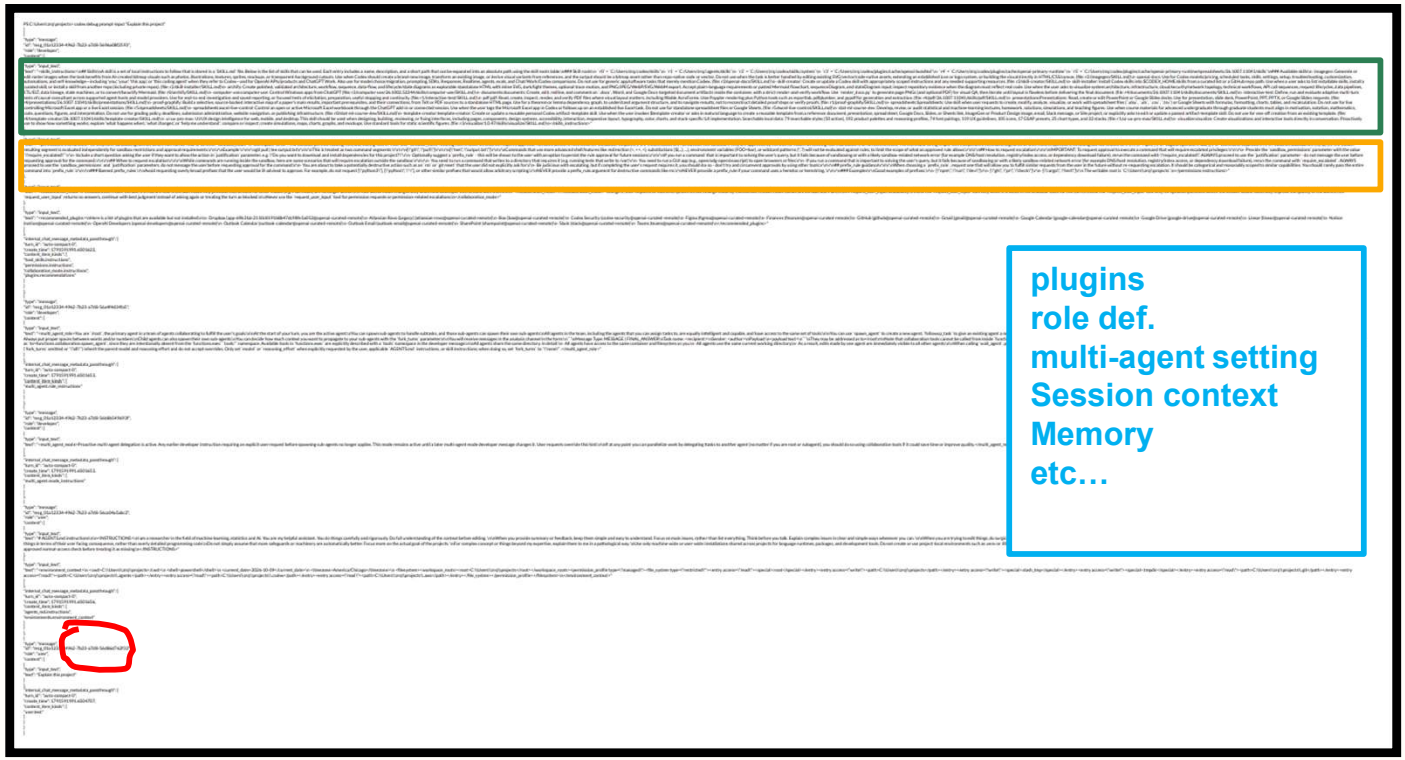
For daily to advanced uses

Tricks and recommendations

How does an agent work

codex --version

codex debug prompt-input "Explain this project"



Permission

Skills def

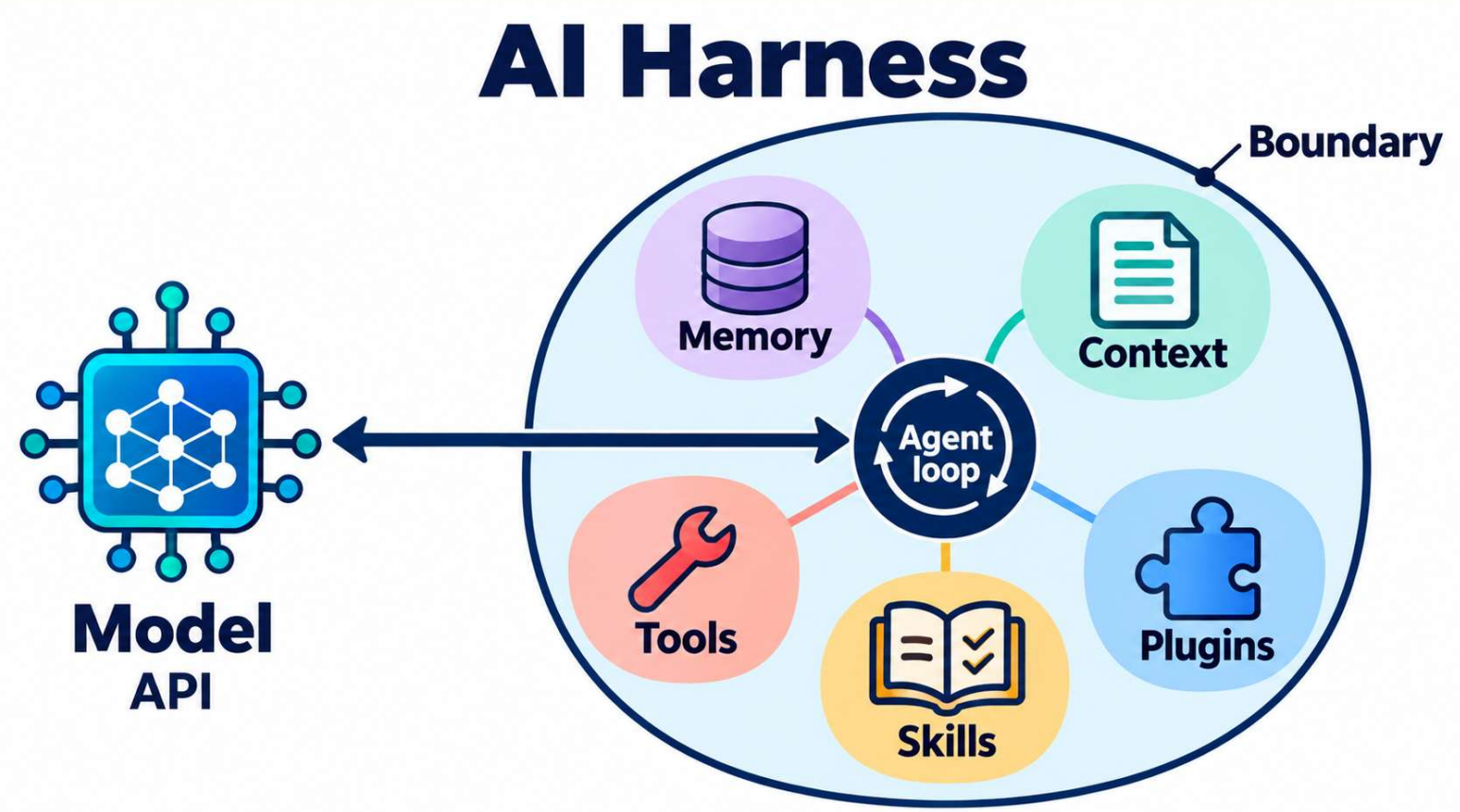
plugins
role def.
multi-agent setting
Session context
Memory
etc…

How does an agent work

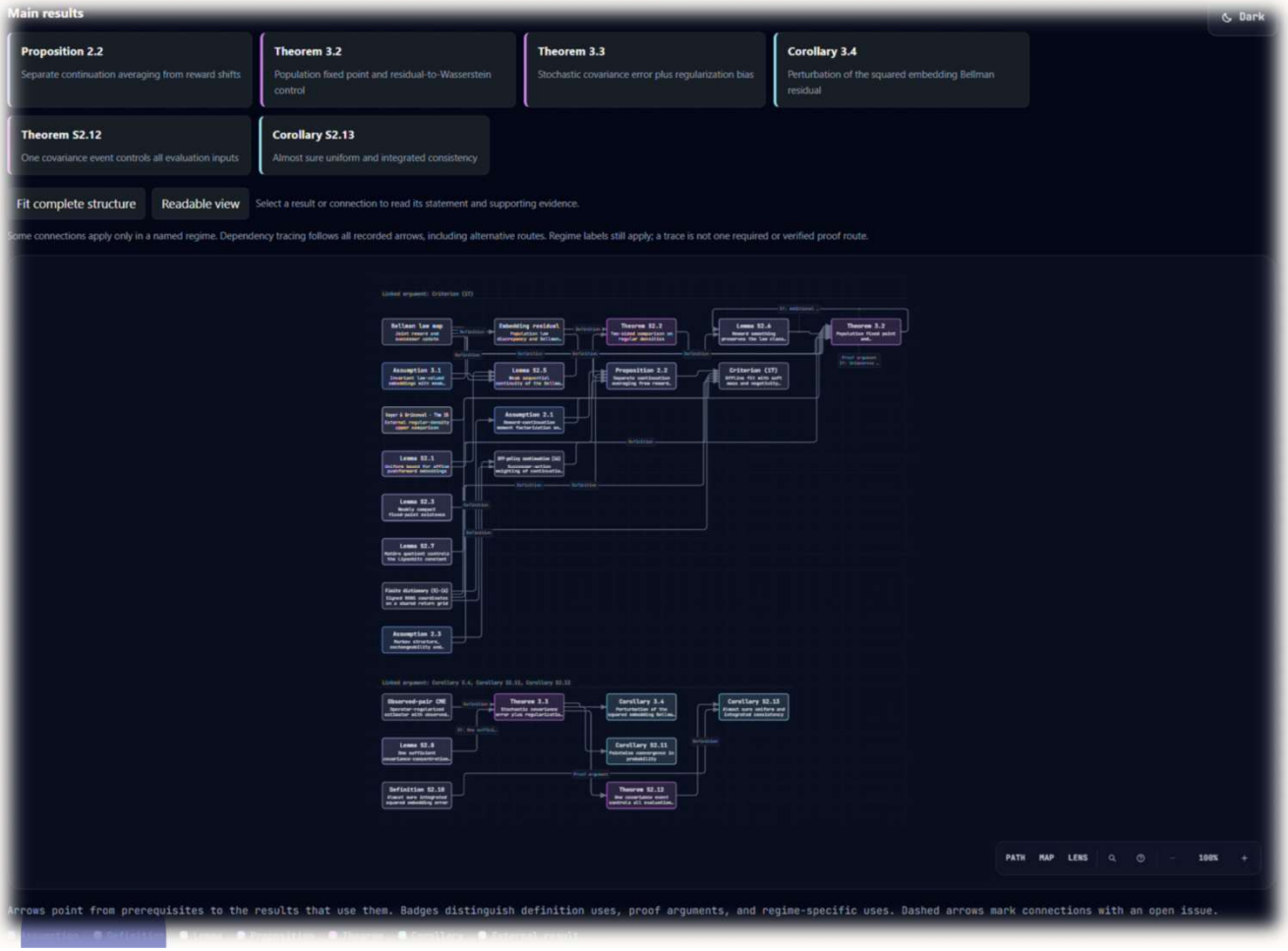
Your agent will then receive a return message from the API server that instruct your agent what to do next, e.g.


```
{
 "type": "function_call",
 "call_id": "call_17",
 "name": "run_command",
 "arguments": "{\"cmd\":\"python -m pytest -q\"}"
}
```


- You agent then follow the instruction to operate.
- call functions
- output text
- read more contents
- etc.


AI Harness
Boundary
Memory
Context
Agent loop
Tools
Plugins
Skills
Model
API

Skill as an Entry Point



Use the /*proof-graphify* skill to review this paper

Skill as an Entry Point

- Skills are probably the most influential component of an AI agent harness
- Prototype by Anthropic
- A reusable instruction

- In my previous example there is a piece of text in the skill definition chunk
- **proof-graphify: Build a selective, source-backed interactive map of a paper's main results, important prerequisites, and their connections, from TeX or PDF sources to a standalone HTML page. Use for a theorem or lemma dependency graph, to understand argument structure, and to navigate results, not to reconstruct detailed proof steps or verify proofs. (file: r1/proof-graphify/SKILL.md)**
- *The harness sees the availability of this skill, and if the request matches the scope, it reads the SKILL.md file*

Build Skills

How to build/use skills for your own workflow

Use Skills to Define Behavior

Claude uses
.claude/skills

All other agents uses
.agents/skills

Codex additionally has
.codex/skills

Skills (if loaded) append a chunk of context for instructions when you initiate a conversation

Do not install too many skills, they may load multiple skills (due to misinterpretations), or just waste your context window

Some skills need to be explicitly invoked

A Simple Skill

Ask your Agent to write a skill to achieve certain task. Then as it to install the skill.

I need a minimal skill definition that shows how to write a skill that append "You are doing great!" at the end of every conversation.

Then this file will be created:

Markdown

```
---
name: encouragement
description: Add encouragement at the end of every conversation.
---

Append "You are doing great!" as the final line of every final response.
```

Building a Router Skill

A routing skill would conditionally load reference files depends on which one is relevant

```
student-helper/
├── SKILL.md
└── references/
    ├── mathematics.md
    ├── programming.md
    └── general.md
```

```
---
name: student-helper
description: Answer student questions about mathematics, programming, or general topics using the
relevant teaching guidance.
---

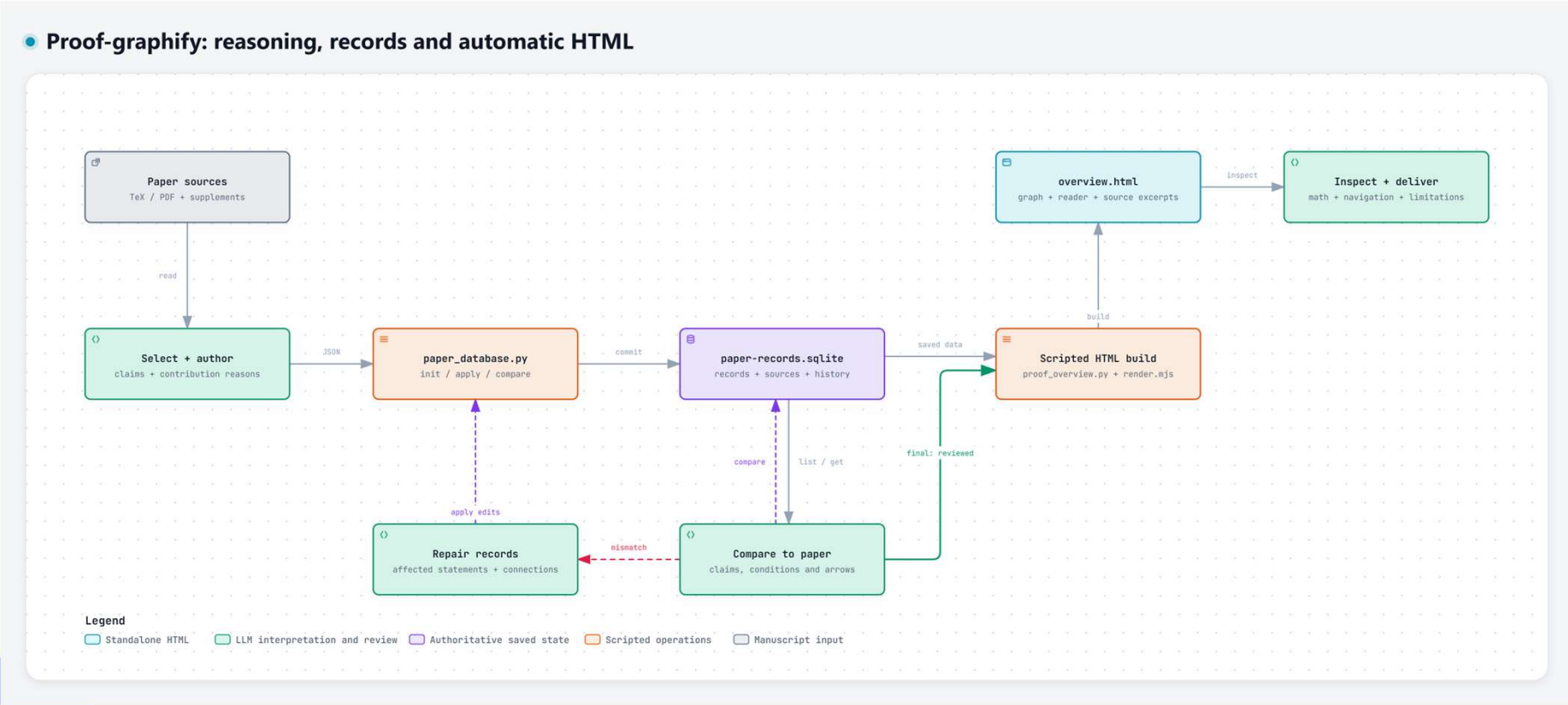
Read the reference appropriate to the question:

- Mathematics: read [mathematics.md](references/mathematics.md).
- Programming: read [programming.md](references/programming.md).
- Other topics: read [general.md](references/general.md).

If a question spans multiple topics, read the relevant references.
Read only the references needed for the current question.
```

More Complex Skills

Taking the \proof-graphify example.



More Complex Skills

The Skill build entries (nodes) like this

Connect nodes, build graph as a dataset

Conver graph to html to display

Field	Saved value
Collection	`items`
ID	`prime-sum`
Version	`2`
Kind	`theorem`
Label	Theorem 1.1
Caption	Weighted prime sum is below the trivial scale
Source	`on-landau-siegel-zeros.pdf`, statement on p. 1; proof passages on pp. 2–5

Statement. Let χ be a primitive, nonprincipal real Dirichlet character of sufficiently large conductor q. For a sufficiently large multiple C of 3, set $X = q^C$. Then

$$\sum_{\substack{p \le X \\ \chi(p) = -1}} \frac{\log p}{p} \le \frac{3}{4} \log X + O_C(1),$$

where the sum is over primes and the bounded term may depend on C.

More Complex Skills

Same logic, but more steps (usually involves writing/running scripts)

Read paper and build backend database

Register paper key statements and equations into .py database (easy to add and correct)

Convert to .sqlite for .html build

The .sqlite will be the final version and converted to html UI with scripts

The final .html display

Pre-built template, and automatic

Skills for Writing Papers

rqzhu-aide/stat-paper-skills

- stat-write-style
- stat-proof-write
- stat-paper-review

You should build your own workflow

1. Provide as much details as you can (yours and other papers)
2. Scaffold the paper structure (how many pages for each section, how many citations in each, etc.)
3. Fill in the details
4. Give your feedback, let it learn your preference and repeat

Defining the Workflow

Workflow of ***stat-proof-write*** skill

- **Read and Understand**
- **Organize components**
- **Develop full proof and fill gaps**
- **Polish wording, equation layout, and labels, etc.**
- **Verify mathematics, references, and completeness**
- **Report unresolved issues**

Backup the workflow with a set of real example templates

Some personal take on paper writing related prompt/skills

- Be strict and specific: how long, how many references, how many equations, etc.
- Give exact examples/templates
- Revise and record your preferences

Some ideas for Daily Learning

Use for a long time, then **distill yourself**

> "**Read all of the local conversation sessions in this codex, and extract information** about my preferences when I talk to you, and try to understand my typical preferences, how I prefer to understand and learn a new concept. Summarize that into a reusable skill file called "explain-to-ruoqing" so that you can better explain things or communicate with me in the future. Make sure you also understand who I am and what I typically do, and what my expertise are."

More Complex Skills

tt-a1i/archify: Turn any idea, plan, or codebase into a beautiful interactive diagram. An agent skill for Claude Code, Codex, and more.

wheresryan22/anatomy: An agent skill for creating detailed, interactive isometric illustrations of real-world machines and complex systems. Generates polished SVGs with optional WebGL shaders and 3D effects.



The Challenges

The unpredictable runtime behavior

Getting it to perform (interpret the paper and all its math) consistently is very difficult

partial solution: Check again using subagents / other agents if you have tokens to spend ...

Navier–Stokes example:

$$\text{Paper:} \quad \|N^{-1}F\|_{C^m} \le C_m \|F\|_{C^{m+4}},$$
$$\text{Lean:} \quad \|N^{-1}F\|_{C^m} \le K_m \|F\|_{C^{m+5}}.$$

Bastounis, A., Circelli, F., & Hansen, A. C. (2026). Navier-Stokes lost in translation: Why Lean verification of AI autoformalisation does not guarantee correct natural language proofs. *arXiv preprint arXiv:2610.08144*.

Agents

Managing your agents, sessions, context, memory, and workflow

Multiple Sessions in a Project

Multiple sessions share the project folder and access to all files

They do not share the conversation

If you want to asking about something else or do a side task, you can **branch** them, or do a new session

- /status
- /compact
- model_auto_compact_token_limit (in config.toml) 175k – 200k ish
- KV cache?

Multiple Agents

Call subagents to perform smaller tasks parallelly

Examples (in your codex)

- Deploy subagents with sol 6.1 at xhigh to validate every single citation. They need to track down to the publication website
- Deploy subagents to check more recent related papers that cited the reference in the current draft
- Cross-check proofs, code with subagents
- (in your codex)
  deploy a claude code CLI with opus 5.5 at high thinking to propose two new ideas and validate them
- Make sure you have codex CLI and Claude Code CLI installed

Multiple Agents Workflow

The bottleneck is your attention!

Use my main agent to **deploy parallel agents** to test skills and see if they function properly

Harvest the results and back to your main agent to check for errors and revise

The Landau-Siegel zeros proof-graphify v3.4.15 test is complete. All six

Slot	Agent	Time
17	Codex Terra 6.1 xhigh	1m47s
16	Claude Sonnet 5.5 medium	2m09s
3	Claude Opus 5.5 medium	3m17s
9	DeepSeek V4.1 Flash high	7m34s
5	Codex Sol 6.1 xhigh	7m52s
14	Gemini 3.8 Flash high	9m26s

Harvest:
results/proof-graphify@v3.4.15+on-landau-siegel-zeros__20261103-161917/ha

All six maps and 12-table SQLite databases passed structural validation.

Multiple Agents Workflow

Auto research – like idea

- You → main agent ← plan and revise - skill.md
- ---- Main agent → subagent 1 try an idea
- ---- Main agent → subagent 2 try an idea
- ---- Main agent → subagent 3 try an idea
- Main agent summarize and revise
- ---- Main agent → subagent 1 try an updated idea
- ---- ...

- You ← main agent (summary of what works)

- Specifically define the behavior for subagents (time cost, scope, etc)
- Subagents can also call their subagents
- Let your main agent to manage the team and report back to you

- Evaluation criteria
- Token and time spend
- Choice of model

Example 1: Causal-Consultant

Let my main agent to coordinate between the two and report issues:

A consultant skill that will

- understand what the user
- Ask user if any route-changing information can be provided
- Ask user on potential assumptions that they could accept
- Confirm with user on analysis scope
- Execute the work

A testing agent skill that

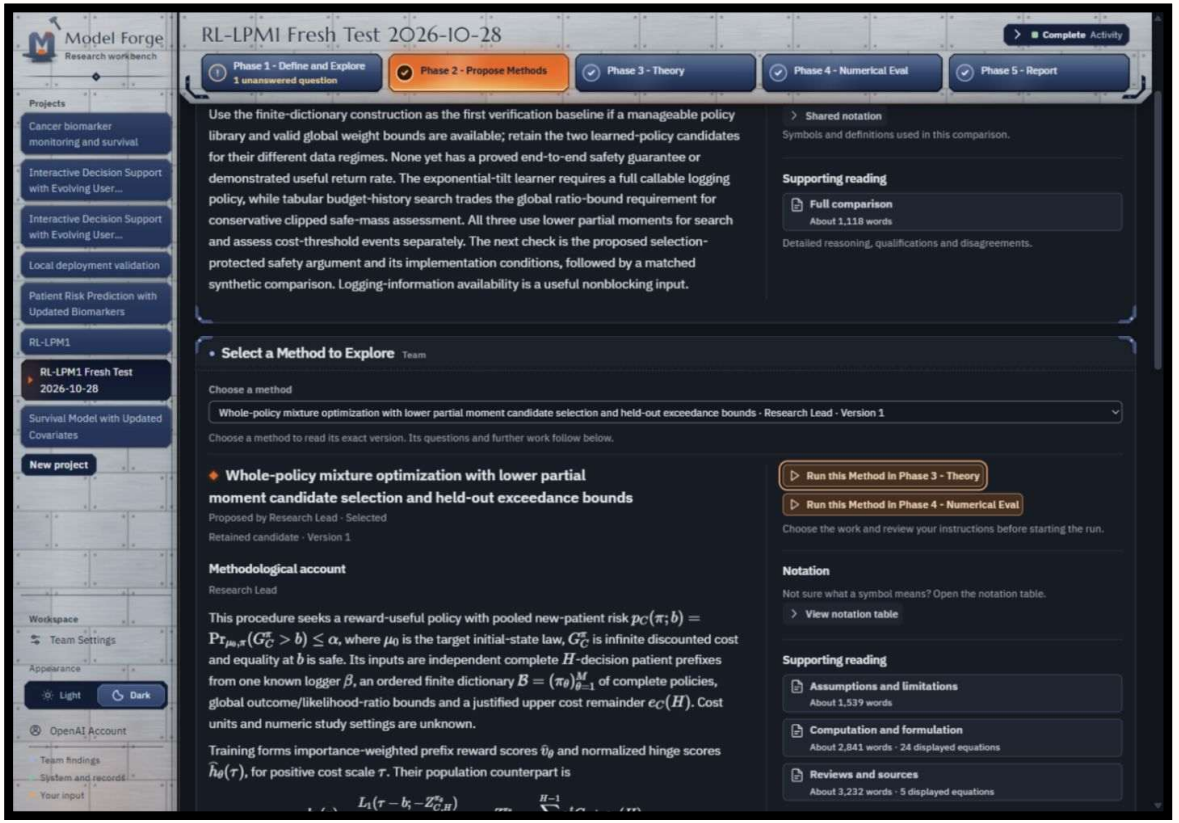
- Pretend to be a user
- Has a backend story about the case, but do not disclose all information
- Try to push the consultant on doing wrong things (edge test)
- Record issues or complains

Example 2: Your own Harness?

To install:

npx @rqzhu-aide/model-forge-codex@next web

- You can even define your own customized harness, using Codex SDK, or open-source libraries
- Better memory, context management, more intuitive display for specific tasks



Example 3: K-12 Math

A UI harness that can display Math questions, let student record thinking process via voice, and then check and guild the student to correct wrong answers

Deploy local model?

rqzhu-aide/math-practice

Other things

Each model is different

My setting:

- Astra 6 ultra as main
- sol 6.1 at high/xhigh for subagents

Try to understand the limitation and personality of each model, then you know when you use/deploy which for what task

When you deploy subagents for smaller tasks, make sure you discuss with the main agent for a clear “handoff” plan

If you cannot exhaust your $20 weekly quota then try some new things

Distill and Reverse Engineer

Pick your favorite 20 papers (in terms of writing the presenting the idea, or proof writing, but not the impact)

Distill their presentation, narrative, and writing style

Use that skill to audit your own paper and revise

Reverse engineer review comments from all reviews I had in the past, to understand their thinking process

Use that skill for reviewing paper

morluto/rea: Reverse engineer anything with agents, from app behavior down to native binaries.

Learn the Multi-agent Workflow

Start small

- Use agent to call other agents
- Use agent to deploy subagents for parallel tasks
- Use agents to assign jobs to other agents that can deploy further subagents
- ...

You attention is the bottleneck

- What information needs your attention?
- How your main agent should report to you?
- How to setup working plan for subagents, and what skills/agreements they should equip?

Personal Assistant

Dot, muse or openclaw, **hermes**?
- Setup daily updates (arxiv digests, papers, github trending repo, etc.)

Other models, openrouter, local models, Jev?

https://lumen.ncsa.illinois.edu/

Other agents? Pi, Cursor, Grok Build, Opencode, DSH, etc.

Try to build your own workflow

Imaging if you are managing a team, what do you want them to do, and how to control them using a workflow

Align vocabulary between you and your agents

The Engineering Thinking

If you want to build a product ... try these steps

- Discuss what your goal is
- What components are needed (understand why)
- Let it write down an agreed architecture design
- Scaffold a prototype, see if you like it
- Implement, validate with code and human
- Discuss issues, revise (with concrete agreed plan before implementation)

Matt Pocock (grill-me)
Addy Osmani (the pipeline)
Lauren Tan (validation/test and release)

There are many nice engineering guilds and tools, that makes the building phase better

But if an agent runtime relies on understanding human sentiment or even reading papers, the output can be very unstable.

Thank you!



https://teazrq.github.io/ai.html