

AI for Finance Research



Manish Jha

October 30, 2026

Why I am the one talking about this

- My research uses language models as measurement tools. Managerial expectations from conference calls, CFO expectations, the politics of shareholder voting, finance sentiment.
- Some of it studies AI as the topic. What generative AI does to firms, managers, and forecasts.
- I use AI agents every working day. Data pipelines, writing, referee-style checks, teaching tools.
I have also borrowed from a few articles, listed in the further readings.
- I have also been on the other side. I referee LLM papers, so I see what gets rejected and why.
- A promise for today: most examples come from other people's papers. Mine show up only where they illustrate a design choice.

What you leave with today (hopefully :P)

1

A tool plan

Which tool to use for which task, what it costs, and what I would buy first.

2

A working loop

Plan, review, execute, verify. A stress-test prompt you can paste into any chatbot.

3

Your own CLAUDE.md

A one-page instruction file so the AI stops guessing who you are and how you work.

4

One skill

A reusable procedure for a task you repeat: a referee check, a voice file, exam versions.

5

A reading map

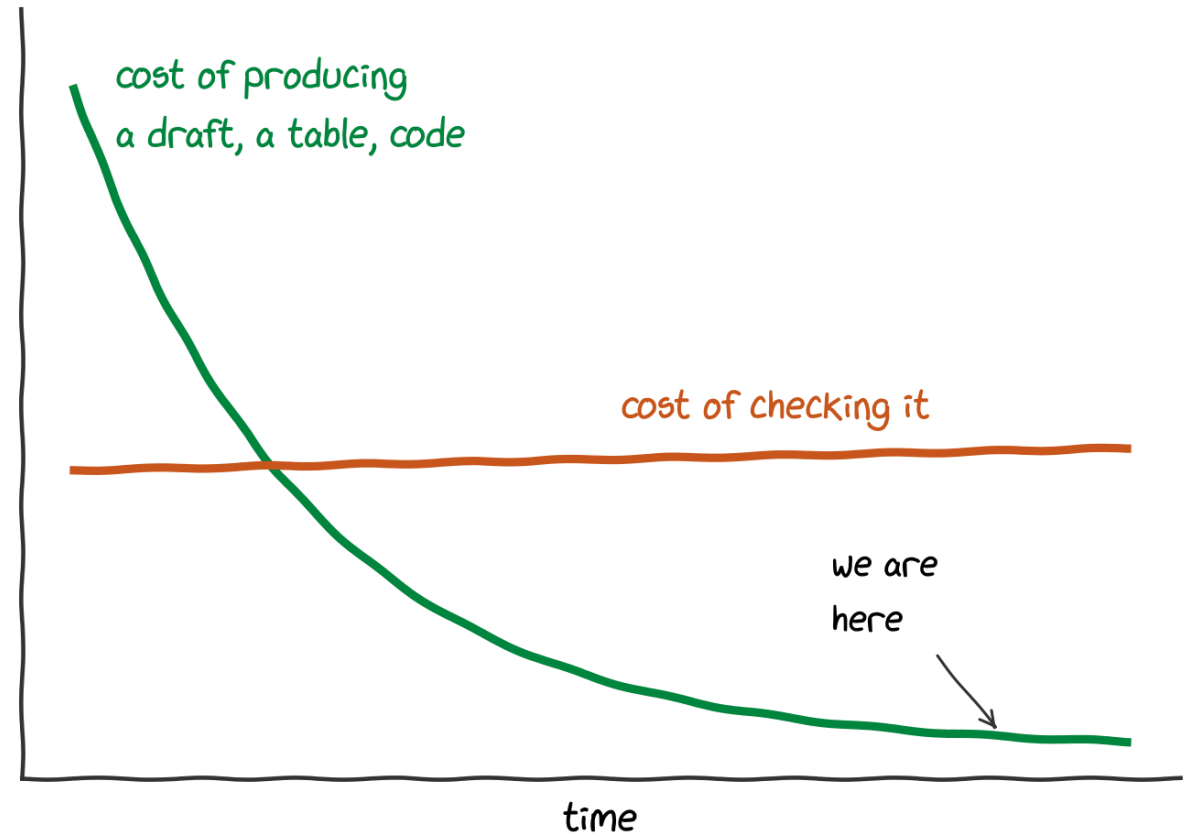
The 20 or so papers that define AI as a finance research topic, by area.

6

A referee checklist

The questions every LLM paper gets asked, so you can answer them before submission.

Producing answers is cheap now. Putting your name on one is not.



AI is both a research tool and a research topic

AI as a tool

- Coding, data collection, cleaning, and pipelines
- Measurement: classify, score, and embed text at scale
- Simulated agents and survey respondents
- Writing, editing, literature search, referee-style checks
- Teaching materials and course logistics

AI as a topic

- A **technology shock** to firm values and corporate policies
- **Labor**, productivity, and who gains inside finance jobs
- **Information production**: analysts, disclosure, prices
- **Financing and investment** in AI capital, energy, concentration
- What cheap generation does to the **research market itself**

Module 1

The tool landscape

What are the tools, who does what well, and what it costs

- Three layers of tools
- Who does what well (my view, October 2026)
- Smaller tools, Office add-ins, artifacts
- What it costs and what I use
- Hermes, Qwen, and everyday chores

Three layers of tools

	Examples	How it works	Good for	Limit
1. Chat assistants	Claude, ChatGPT, Gemini, Copilot	You paste text in, copy answers out	Questions, drafts, explanations	It never sees your folder or runs your code
2. Agents on your files	Claude Code, Codex, Gemini CLI, desktop agents	Reads your folder, writes scripts, runs them, fixes errors	Data, pipelines, replication, long documents	Needs supervision and a backup
3. Models you run	APIs (pay per token) and open-weights models (Qwen, Gemma, Llama)	Run on a laptop, a university cluster, or a cloud account	Measurement at scale, privacy, reproducibility	Setup cost

Chat versus agent: in chat you run the code and paste the error back yourself. An agent plans first, runs the code, reads the error, and fixes it, and the scripts stay in your folder so you can rerun everything.

Who does what well: my working view, October 2026

Task	What I reach for	Why	Watch out for
Quick general lookups	ChatGPT	Fast web search, good enough for facts you will check	Stale or invented specifics
Deep literature or market research	ChatGPT or Gemini Deep Research; run two	Long multi-source reports with links	Fake or misattributed citations
Writing and editing in your voice	Claude	Follows style rules and structure well	Drifts to a generic voice without a style file
Code, data, pipelines on your files	Claude Code or Codex	Agent runs and debugs code in your folder	Silent sample changes, scope creep
Very long PDFs, many documents	Gemini, NotebookLM, or an agent	Large context and grounded answers	Misses details in the middle of long inputs
Second opinion on a plan or paper	A different vendor than the author	Different blind spots	Agreement is not proof
Confidential or bulk measurement	Open-weights model run locally or on HPC	Data stays home, fixed model version, low cost	Weaker reasoning, setup time
Images for slides	ChatGPT or Gemini	Native image generation	Do not use for data figures

Rankings change every few months. Pick by task, test on something you know the answer to, and re-test after major releases.
Polymarket, late October 2026: Anthropic 65% to have the #1 coding model at year end, OpenAI 12%, Google 9%. A crowd forecast, not a benchmark.

Smaller tools worth knowing

Voice dictation

Wispr Flow or similar. You talk in full paragraphs, so prompts get longer and better.



NotebookLM

Upload a set of papers and ask grounded questions with citations back to the source. Useful for a reading group or a new literature.

Refine.ink

Ben Golub's tool that checks a paper's internal consistency: do the claims match the results? Different from a taste-based referee report.

GitHub Copilot in VS Code

Inline code suggestions; free with an .edu address. An easy first step before agents.



DuckDB and parquet

Not AI, but agents work well with them. A 70 GB set of CSVs became 6 GB of parquet.

Git

Commit before you let an agent edit anything, then read the diff.

Claude inside Excel, Word, and PowerPoint

 Claude

Getting started

Install "Claude for Microsoft 365" from Microsoft AppSource and sign in with your Claude account.

Excel

Reads every sheet, explains and writes formulas, merges and cleans data. Built-in skills for **model audits** and data cleaning.

Word

Edits land as **tracked changes** you accept or reject. Works through comment threads and replies with what it changed.

PowerPoint

Edits slides in your own template and updates a deck when the numbers change.

Across apps

One conversation spans the open workbook, document, and deck, so you explain the task once.

Gamma and Claude artifacts

Gamma (gamma.app)

- Paste an outline or notes; it lays out a deck, a one-page document, or a website. Exports to PowerPoint and PDF.
- Good for a first draft. **Do not use it for data figures.**

Claude artifacts

- Ask Claude for an interactive page (a quiz, a calculator, a small dashboard) and share it as a link.
- Example on the right: **Finance JMP Tracker**, built by Siyang Liu (PhD student, UBC). Tabs for candidates, data usage, topics, publications, placement.

Finance JMP Tracker

Finance job market candidates from 53 universities. Updated 30 Sep 2026. Links go to the original sources.

Disclaimer: compiled from public sources with AI assistance; may be incomplete or contain errors, and some earlier-year job market papers are inferred. Not affiliated with any listed school. Corrections or removal requests: siyang.liu@sauder.ubc.ca

Candidates

Data usage

Research topics

Publication outcomes

Placement outcomes

2026–27 market

2025–26 market

2024–25 market

2023–24 market

2022–23 market

2021–22

2019–20 market

2018–19 market

All years

61

 candidates

26

 universities posted

12

 with PDF

40

 with JMP title

SCHOOLS

Berkeley	not available	Bocconi	1	Boston College	4	Carnegie Mellon	2	Chicago	
Copenhagen	not available	Cornell	not available	CU Boulder	not available	CUHK	not available	Duke	
Florida	1	Harvard	not available	HEC Paris	3	HKU	1	HKUST	
Indiana	2	INSEAD	2	LBS	not available	LSE	6	Maryland	

The API is much cheaper than people expect

Scoring 100,000 call excerpts, each about 3,000 tokens in and 300 tokens out, at October 2026 list prices.

Tier	Models	Per M tokens, in / out	Cost for the full corpus	With Batch API
Small For extraction and first-pass classification	GPT-6 Luna, Claude Haiku 5.5	\$0.10 / \$0.50	<div></div> \$45	\$23
Middle For rubric scores that need reasoning	GPT-6.1 Sol, Claude Sonnet 5.5	\$2 / \$10	<div></div> \$900	\$450
Frontier Rarely needed for measurement	GPT-6 Astra, Claude Fable 5.1	\$10 / \$50	<div></div> \$4,500	\$2,250

A \$20 chat subscription cannot run 100,000 calls at all, and Claude Code should not be the loop.

My order: start with the small model, hand-check 200 scores against my labels, and move up a tier only if it fails. OpenAI's small and batch tiers make this first pass very cheap.

What I use

1

Claude (\$20)

Most of my work runs through

Claude Code:

data pipelines, writing and editing, referee-style checks on my own papers, teaching tools.

2

ChatGPT (\$20)

Search, quick questions, Codex, and a second model on my writing.

3

Gemini (free)

Another second opinion, and long documents. I have used it as one of three AI 'referees' on my own paper.

4

Hermes + Qwen

Local. Small chores that need little reasoning and should stay on my machine. Complements Claude Code.

I pay \$20 a month each for Claude and ChatGPT. Both \$20 tiers include their coding agents (Claude Code, Codex). For writing, one drafts and the other edits. Rule I follow: pay for what you use daily, use the rest free until you hit their limits.

Hermes with Qwen for small jobs

What it is

- Hermes Agent: open-source agent from Nous Research
- **Runs on your machine** and keeps memory across sessions
- Ollama runs language models locally; Qwen and Gemma are the usual picks



HERMES
AGENT



Qwen3

How I use it

- Jobs that need **little reasoning where privacy matters**
- Example: Hermes pulled about 100 sent Outlook emails into text files
- Claude read them and wrote my email-voice rules (Module 3); raw emails never reached a cloud model
- I use Hermes to collect files and Claude Code for anything that needs judgment
- **Model size:** "35B-A3B" means 35B parameters, about 3B active per word. Plan on 16 to 24 GB of free memory.

Small chores I now hand to an agent

Affiliation (University) (required)

Please enter your affiliated university

fill this form



Sonnet 5 High



Fill in a web form (Claude in Chrome)

In the Chrome side panel I type "fill this form." The agent reads the page and fills affiliation and position. I check every field before I submit.

add to outlook calendar

Dear All,

Our next brown bag presenter is Ms. Sirui Cai. She will present her JMP paper titled, "What Lenders

I need to find the right tool to add an Outlook calendar event. Let me search for available tools.

Add a seminar to my calendar (Hermes)

I paste the brown-bag email into Hermes and write "add to outlook calendar." It searches its tools for a calendar connector and creates the event.

save this skill in my claude and codex

github.com/phuryyn/work-humanizer

Done. The work-humanizer skill is installed in both:

- **Claude Code:** `~/.claude/skills/work-humanizer/` (SKILL.md, check.py, tells.md)
- **Codex:** `~/.codex/skills/work-humanizer/` (SKILL.md, check.py, tells.md)

Install a skill in two agents (Hermes)

I paste a GitHub link and ask Hermes to **save the skill in Claude and Codex.** It clones the repository and copies the files into both skill folders.

Each chore needs little judgment and is easy to check afterwards. That is what makes it safe to delegate.

Module 2

How you can use AI

Where it helps in the research process, what to delegate, and how to check what comes back

- Where AI helps in the research process
- The core loop, and specific prompts
- Fresh-agent review, failure modes, diffs
- From EDGAR filings to a database
- Big batches through the API
- Writing, literature, the stress test

Where AI helps across the research process

Stage	AI does well	You keep	My rating
Idea	Many candidate ideas, alternative explanations, quick novelty search	Which question matters, taste	Useful
Literature	Finding papers, summaries, multi-source reports	Checking every citation exists and says what is claimed	Useful/ Strong
Data	Downloads, scraping, APIs, parsing messy files, building databases	Sample choices, what counts as an observation	Strong
Measurement	Classifying and scoring text at scale	The construct, the rubric, the validation sample	Strong
Analysis	Regressions, tables, figures, cross-language replication	Identification, fixed effects, clustering, control group	Strong code, weak design
Writing	First drafts of routine text, editing, comments	The argument and your voice	Useful
Review	Internal consistency checks, simulated referees	Which criticism to act on	Strong
Identification	Listing threats you forgot	Whether the design is credible	Keep human

The core loop: plan, review, execute, verify

1. Plan	2. Review the plan	3. Execute	4. Verify
Ask for a plan before any code. In Claude Code, use plan mode (Shift+Tab) . In chat: "Plan this out before doing anything."	Check file paths, variable names, sample filters, assumptions. Have a fresh agent stress-test it. Answer its design questions yourself.	Only when the plan is verified, the scope is bounded, and the result is reversible (git, backup, untouched originals).	Row counts in and out. Variable names against the data. git diff for every change . Values against a known source.

Without a plan, the code often runs but the result is wrong, and the **debugging takes longer than the plan would have.**

Vague and specific prompts

Vague	Specific
"Clean this data."	"Drop observations where revenue is missing or negative. Keep all others. Report how many were dropped."
"Run a regression."	"Regress log revenue on trade_exposure with firm and year fixed effects, clustered by firm, 2005 to 2020."
"Make a table."	"LaTeX table, columns (1) to (3) with progressively richer fixed effects; N, R-squared, and FE indicator rows at the bottom."
"Fix the error."	"The merge fails because firm_id has leading zeros in the master file but not the using file. Fix the keys."
"Is this a good idea?"	"You are an external reviewer, not a collaborator. Find the top three ways this fails. Do not rewrite it."

Review with a fresh agent

prompt

```
# In Claude Code, after the work is done:  
Run a fresh, independent agent to review this script.  
Give it only the code and the data dictionary,  
not our conversation. Check:  
- variable names against the actual data  
- whether each merge keeps the observations it should  
- fixed effects match the level of the treatment  
- standard errors clustered at the right level  
Flag anything suspicious. Do not edit my files.  
  
# In a browser: paste the plan or draft into a NEW chat,  
# never the one where you built it.
```

Why it works

- A conversation that produced the code is biased toward defending it. It "knows what it meant."
- A fresh context reads only what is on the page, like a coauthor seeing it cold.

Eight ways AI fails in research

Hallucination

Invented file paths, variable names, Stata options, citations. **The errors look plausible, so they are hard to catch.**

Laziness shortcuts

Claims tables are "redundant" to **skip work.** If it concedes as soon as you push, it was not checking.

Memory over observation

"That table has a gvkey column." True in the schema. Three rows populated in your data.

Scope creep

Asked to add a control; also **changed the sample and switched estimators.**

Context degradation

After **30 to 40 exchanges it forgets rules** and repeats rejected ideas. Start fresh.

Confident domain errors

Wrong clustering level, wrong identification argument, stated with full confidence.

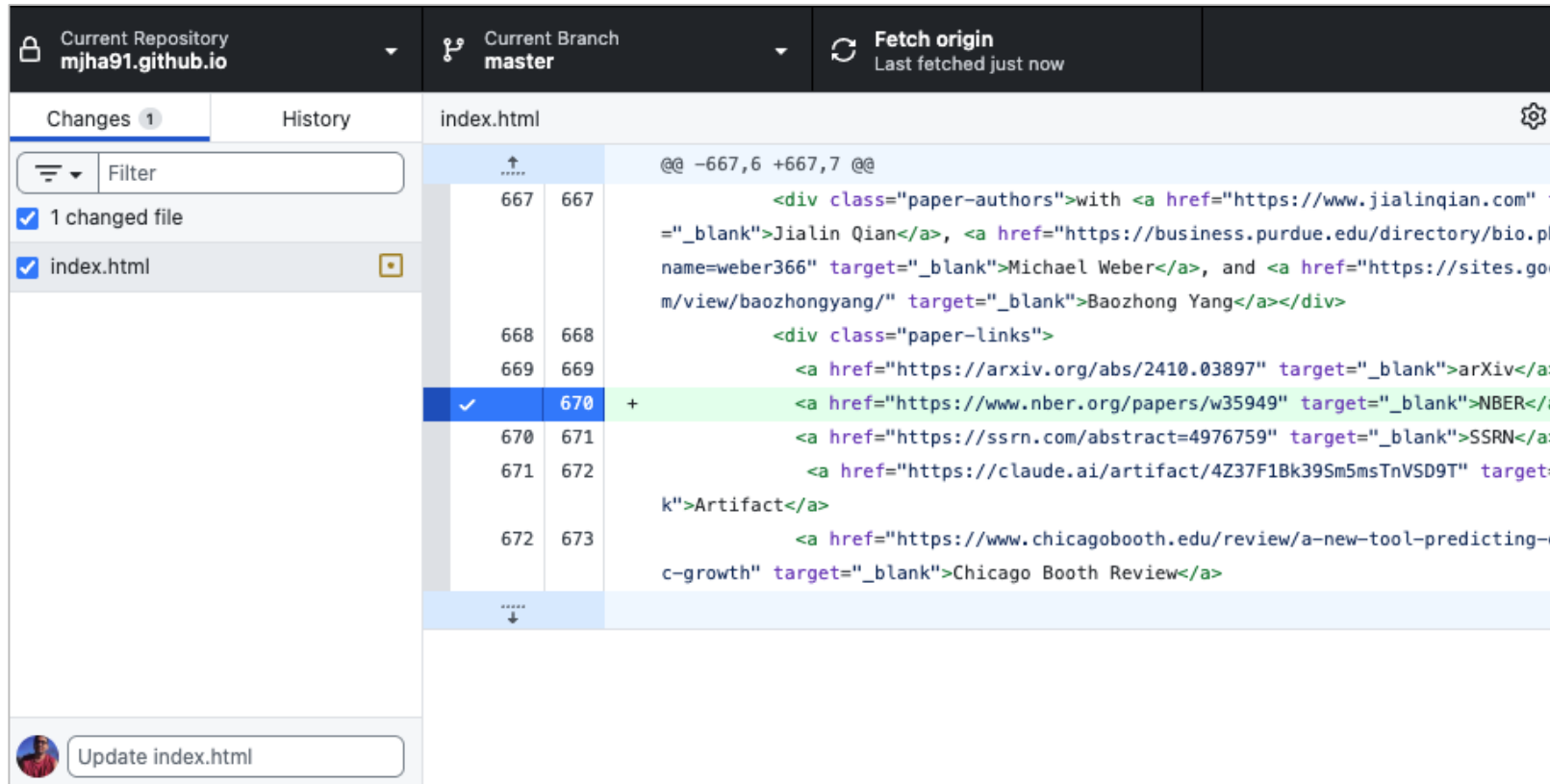
Silent data decisions

Drops observations or changes merge types without saying so.

Configuration bloat

Instruction files past about 500 lines get skimmed; rules at the bottom get ignored.

See what the agent changed: GitHub Desktop or git diff



What to look for

- Left panel lists every changed file. More files than you expected is the first warning.
- Green lines were added, red lines removed, line by line.
- **Untick anything you did not ask for**, then commit with a short message.
- **Free**, and works on any folder you put under git, not only code.
- **In a terminal:** `git diff HEAD~1 HEAD file.docx` shows the same green and red lines for one file.

From EDGAR filings to a queryable database in 30 minutes

Goldsmith-Pinkham (2026), "From EDGAR Filings to a Structured Database using Claude Code," video 3 of 5.

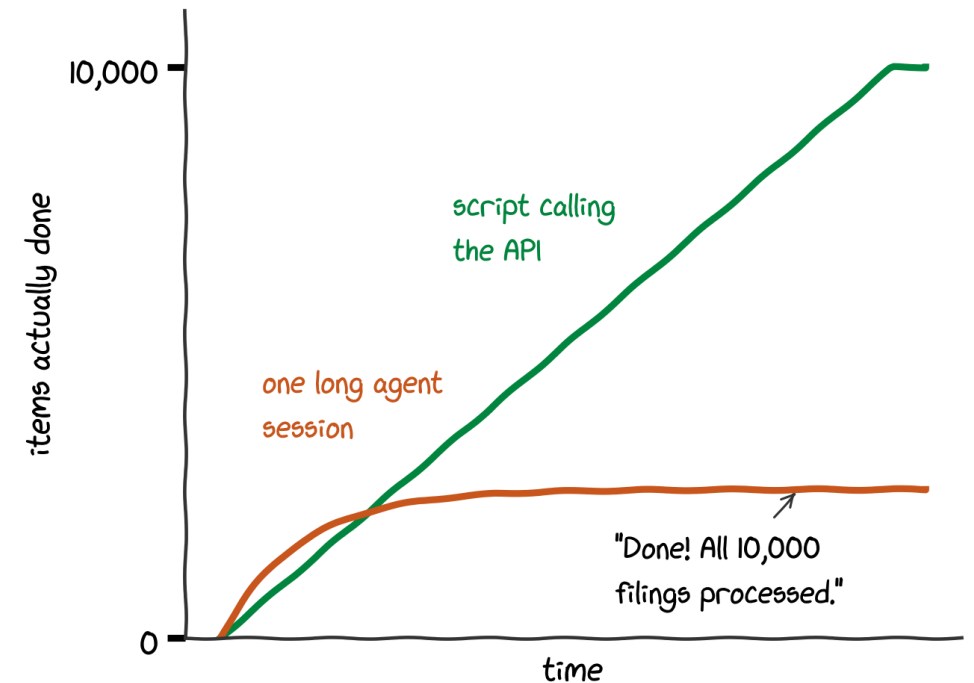
Question	Did firms add tariff language to 10-K Item 1A risk factors after the 2025 tariff escalation?
Context given up front	Every firm has a CLK; the API address; the 10 requests per second limit; a User-Agent header is required; about 30 trade-exposed firms.
Agent behavior	Entered plan mode on its own, asked three design questions (database format, keyword search vs LLM classification, User-Agent identity), saved the plan to a file.
Quality	Parsed Item 1A from 112 of 120 filings, investigated the 8 failures, fixed a regex: 119 of 120. One filing needs a human.

What I would copy

- Front-load the rules of the data source before code is written
- Let it ask you the design questions; answer them yourself
- Demand a quality report: what parsed, what failed, suspiciously short extracts
- Save the parsed database, not only the raw HTML

Big batches: have Claude Code write an API script

- Inside one Claude Code session, a long list goes wrong quietly. The context fills up, the agent starts to summarize or skip, and it can still report "done."
- The API takes **one request per item**. A script loops over every ID, logs each call, retries failures, and stops only when the list is empty.
- So for anything beyond a few hundred items: **use the API**. Let Claude Code write the script, test it on 20 items, check those by hand, then run the rest.
- The **Batch API** takes a large file of requests at half the normal price and returns results within **24 hours**.
- If you stay inside Claude Code, work in **chunks of about 1,000**. After each chunk, compare output IDs with input IDs and spot-check a few before starting the next.



What to hand off when writing

Good to offload

- Boilerplate emails and memos
- First-pass write-ups of regression output
- Variable definition appendices
- Data and code documentation
- Converting notes to LaTeX or slides
- Copyediting, consistency, typos

Keep, or use AI only as a critic

- The introduction and the contribution
- How you interpret a result
- Anything you must defend in a seminar
- Ask for comments, not rewrites:
 - "Do not edit my text. Insert comments where the argument is weak or unclear, the way a good editor would."
- Test: can you discuss the idea in depth without the AI?

Does it read as AI-written? Pangram (pangram.com) marks passages that look machine-written; Emulate (tryemulate.ai) rewrites them. Use both to find the sentences to rewrite yourself. A score alone is not evidence against a student.

Literature search with two deep-research tools

- One model gives you one set of blind spots.
- Run the same structured prompt through two or three deep-research tools. I read the places where they disagree most closely.
- Write the prompt with: task, scope (years, sources), evidence rules, output structure, anti-goals, a comparison anchor, a self-check, and an explicit "search live sources, cite by URL."
- File the reports inside the project folder they belong to, not in a central pile you never reopen.
- Verify every citation before it enters a paper or a referee report. Open the link.

Three agent example

Two tools treated a policing result as well replicated. **The third noticed most follow-ups traced to one original experiment and found a null replication the others missed.**

The stress-test prompt

paste into a fresh chat

You are an external reviewer, not a collaborator. Your job is to find problems, not to encourage.

Stress-test this plan. For each issue, point to the specific step or sentence that triggered your concern. Classify each finding as:

- MUST FIX: will likely cause failure or major rework
- WORTH CONSIDERING: a risk I can accept consciously

Find:

1. The top 3 ways this plan could fail; work backward from failure
2. What someone experienced in this area would expect that is missing (data, identifiers, sample periods, dependencies)
3. Assumptions or timelines that look unrealistic, and what evidence would confirm or refute them
4. What should change before proceeding

Do not rewrite the plan. Diagnose first. End with a verdict: PROCEED or REVISE FIRST.

THE PLAN:

[paste here]

Referee reports: sort the comments once, then move on

Each round of answers to an AI referee raises new objections, so the **loop does not converge**.

Quick fixes

do these right away

- Minor comments: wording, typos, missing citations, table notes
- Clear them all in one pass

Must do

do these before submitting

- Concerns that would hurt the paper if the referee is right
- Fix these properly and check the result

Aspirational

good to do, hard to do

- Needs new data, a new test or a new section
- Do it only if cheap; otherwise note it as a limitation

Sort the comments once, act on the list, and move on.

Module 3

Making it yours

Instruction files, skills, and agents

- CLAUDE.md: global, project, keep it short
- What a skill is
- A voice file in five steps; my skills
- A three-agent idea lab

A starter global CLAUDE.md for a finance researcher

~/ .claude/CLAUDE.md

Who I am

Finance professor. Empirical corporate finance and asset pricing.

Tools: Stata, Python, SAS on WRDS, LaTeX/Overleaf. Projects in ~/Research/.

Skill level: strong in Stata, learning Python. Explain shell commands.

How I work

- Be concise. Plan before executing anything non-trivial.
- Ask before deleting or overwriting files. Never touch data/raw/.
- Only change what I ask for. No refactoring or renaming unless asked.
- Script-first: every table, figure, or dataset comes from a script in code/ that I can rerun.

Finance conventions

- Firm identifiers: gvkey (Compustat), permno (CRSP). Link via CCM.
- Winsorize continuous variables at 1/99 unless told otherwise.
- Cluster standard errors by firm unless the treatment varies higher.

Verification

- **After any merge: report matched and unmatched counts.**
- After any regression: report N, sample period, FEs, clustering.
- Never invent a citation. Mark anything unverified as UNVERIFIED.

Notes

- **Start with 3 to 5 rules, grow from real mistakes.**
- The finance conventions block stops a whole class of silent errors.
- Verification rules turn Module 2's checklists into defaults.
- Test it: start a session and ask **"What do you know about me and my work?"**

A project CLAUDE.md: the facts the agent would otherwise guess

~/Research/bank-lending/CLAUDE.md

```
# Project: Bank lending and local deposit shocks
Main panel: data/clean/bank_county_year.parquet (2001 to 2023)
Keys: rssid_id (bank), fips (county), year. Not 'bank_id', not 'cnty'.
Outcome: log_cre_loans. Treatment: deposit_shock (built in 02_shock.py)

# Folders
code/    numbered scripts, run in order
output/tables/, output/figures/    generated only by scripts
data/raw/    read-only. data/temp/    scratch

# State
Done: 01_merge.py (do not modify without asking)
Now: baseline regressions in 03_baseline.py
Next: event study around 2008 and 2020

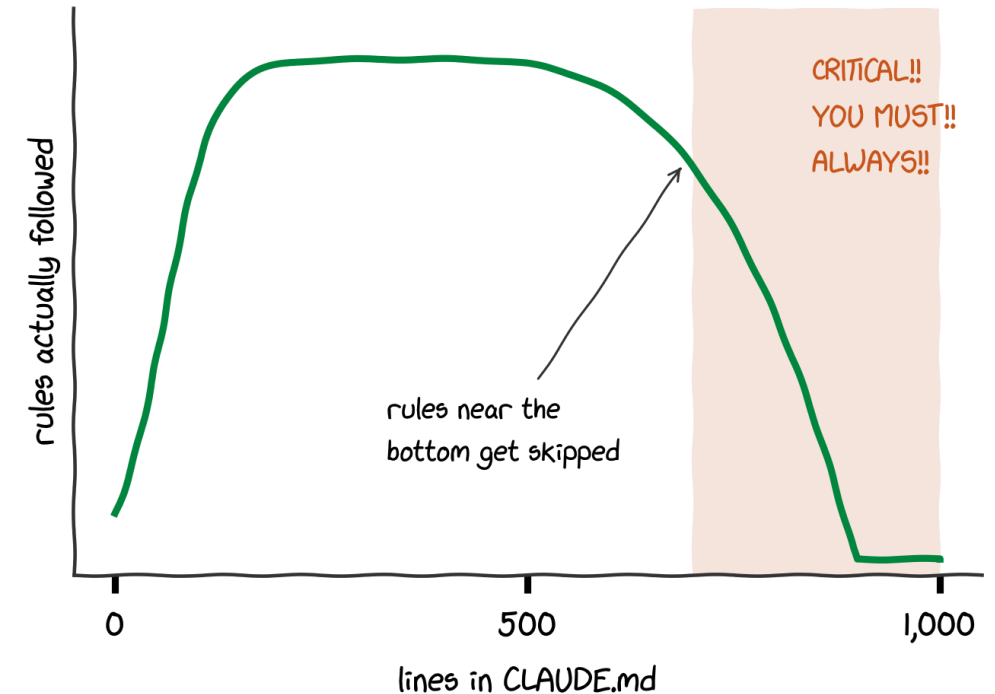
# Rules for this project
Cluster by county. Bank-by-year FEs in all main specs.
```

Why it matters

- Hallucinated variable names are the most common failure. This file removes the guessing.
- "Do not modify without asking" protects finished work.
- The State section doubles as a handoff note when you start a fresh session.

Keep instruction files short

- **Drop the shouting.** Strip ceremonial CRITICAL, YOU MUST, ABSOLUTELY. Newer models **over-react** to it and waste effort.
- Keep **emphasis only where a mistake cannot be undone**: sending email, deleting data.
- State the positive: "Plain paragraphs, no headers" beats "DO NOT use markdown." Put the **reason next to the rule**: "skipping step 3 leaves stale data."
- **Keep it short.** Past about 500 lines, files get skimmed and late rules are dropped.
- Global file holds universal rules only; project facts go in project files. Move growing multi-step instructions into a skill.
- Monthly: ask a fresh session to **prune your CLAUDE.md for dead**, duplicate, or conflicting rules.



A skill is a folder with one markdown file

```
~/ .claude/skills/
```

```
~/ .claude/skills/  
  check-merge/  
    SKILL.md          <- required, exactly this name  
  referee-own-paper/  
    SKILL.md  
    checklist.md      <- optional supporting files
```

```
--- SKILL.md ---  
---
```

```
name: check-merge  
description: Review the most recent merge in a do-file or script  
             for correctness. Use after any merge of two datasets.
```

```
---
```

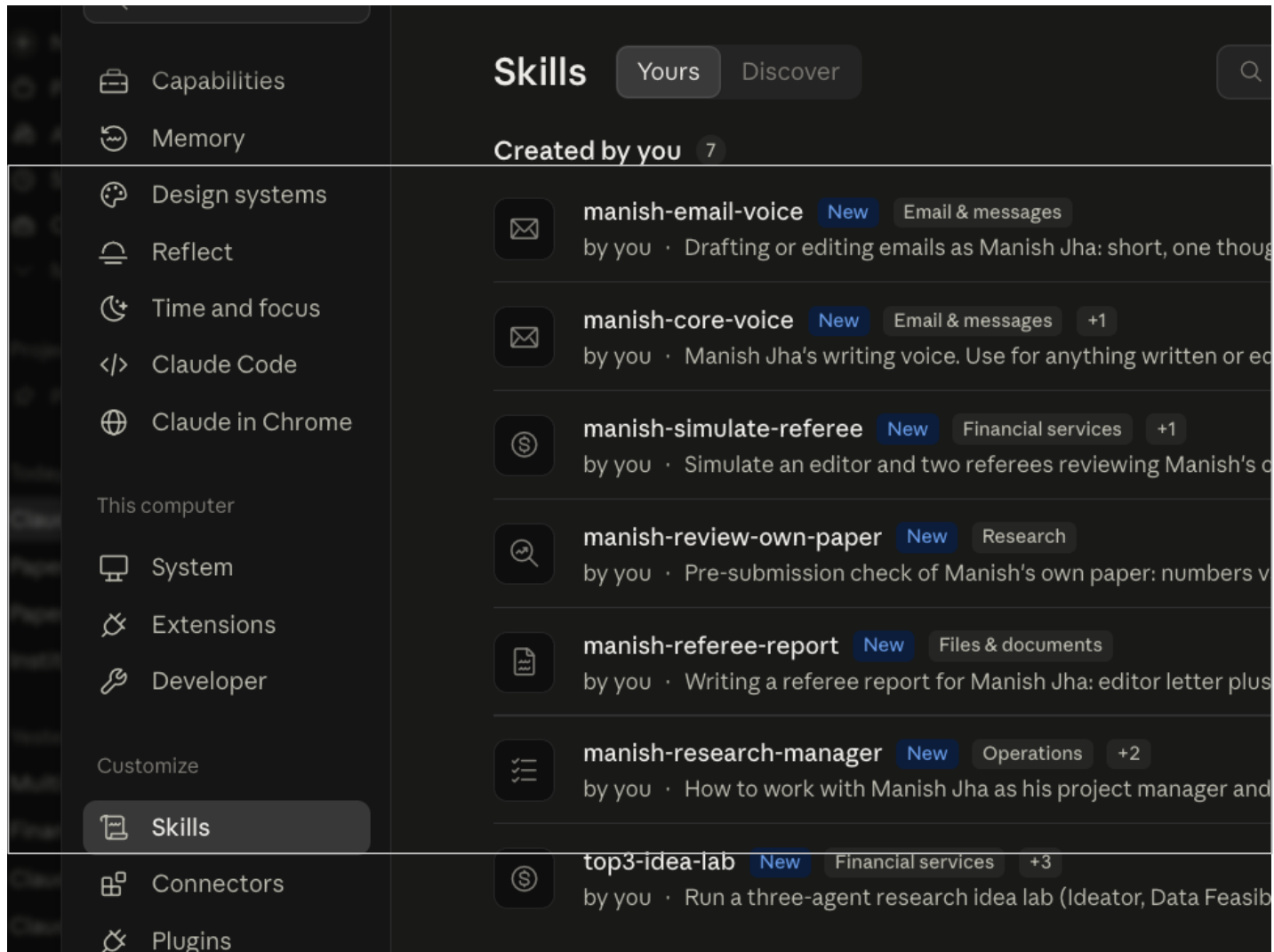
```
Check:
```

1. Are the merge keys right? Compare to the actual variable names.
 2. How many matched vs unmatched? Tabulate `_merge`.
 3. Did N change unexpectedly?
 4. Any duplicate keys that could make a many-to-many merge?
- Report findings. Flag anything suspicious. Do not edit files.

What to know

- Invoke it by name `(/check-merge)`, or the agent loads it when the description matches the task.
- The description line is what triggers it, so write it carefully.
- Flat files like `skills/check-merge.md` are silently ignored. Each skill needs its own folder.
- In the Claude apps, you upload a skill under Settings, and it works in chat too.
- Easiest way to make one: ask the agent. "Turn what we just did into a skill."

My seven skills, as they appear in the Claude app



- **Voice:** core voice, email voice
- **Papers:** review my own paper, simulate referees
- **Refereeing:** report format and workflow
- **Ideas:** a three-agent idea lab
- **Operations:** how to work with me as a research manager
- Each one grew out of a task I had done by hand at least three or four times.
- Next slides show short, sanitized excerpts.

Build a voice file in five steps

- 1 **Collect.** Five to ten pieces you are proud of, in different genres: a paper introduction, a referee report, emails.
- 2 **Analyze.** Ask the model to extract patterns and quote your own text as evidence for each one.
- 3 **Write rules.** Turn the analysis into a one-page spec with testable rules and a ban list.
- 4 **Add examples.** Three to five annotated excerpts with context and notes. This step matters most.
- 5 **Load and test.** Put it in Project instructions or a skill. Ask it to rewrite something you already wrote; compare.

When a draft comes back wrong, **fix the voice file as well as the draft.** After three or four rounds, most people find it reliable.

My core-voice skill: stance, evidence, and a ban list

core-voice/SKILL.md (excerpt)

Stance

1. Lead with the claim. The first sentence of a paragraph states the finding. Evidence follows.
2. First person, active. "We find", "I do not find these convincing".
3. Present both sides before testing.
4. State the limits of the claim plainly.
5. Calibrate verbs to the evidence. Correlations are "associations", not effects, unless the design earns causal language.

Numbers and evidence

- Every empirical claim points to a table, panel, or citation.
- "percentage points" for differences in rates, "%" for relative.
- For LLM-based measures, say which model, what the prompt asked, what information cutoff was imposed, and how look-ahead bias is handled.

Banned patterns (AI tells)

- delve, leverage, multifaceted, furthermore, compelling, navigate, foster, crucial, intricate, showcase, seamless, nuanced, ...
- "It's not X, it's Y". Aphorisms. Rule of three by reflex.
- No em-dashes.

Design choices

- Rules are testable: you can check each one on a draft.
- The LLM-measure rule makes every draft pre-empt the referee question we discuss in Module 5.
- The ban list catches most AI-sounding text quickly.
- Context skills (email, referee) sit on top and win when they conflict.

My email-voice skill, built from 100 of my own sent emails

email-voice/SKILL.md (excerpt)

```
## How my emails look
1. Very short. Most are 1 to 5 lines.
2. One thought per line, often with no blank line between them.
3. Greeting on its own line, no comma: "Hi [first name]".
4. Get to the point on the next line: "Please find the table",
   "I was wondering if...", "I thought to share..."
5. Sign off "Regards," then the first name. Never "Best,".
6. Keep my phrasing where it is not standard American email style.
   "Do come by", "I thought to share" are mine. Do not rewrite
   them into "Feel free to".

## Content habits
- Proposals use "We could...". Opinions use "I feel...".
- Offer concrete times, not "let me know when you are free".
- With data: paste the table, then one or two plain sentences.
- Decline plainly and leave the door open.
[Real examples and frequent recipients removed for this talk]
```

How it was built

- Hermes with a local Qwen model exported about 100 sent Outlook emails to text files on my laptop.
- Claude read them and drafted the rules and examples. I edited the draft.
- Rule 6 is my favorite: it protects phrasing that a model would "correct."
- The full file has real emails as examples; I left those out here.

My pre-submission skill runs eight separate checks

A

Consistency

Every number in the text matches the table it cites. Abstract, intro, and conclusion agree.

B

Claim discipline

Causal words on associations, "we are the first" claims, significance treated as magnitude.

C

Notation

Each symbol defined once, specification matches the table columns.

D

Tables and figures

Notes state sample, units, FEs, clustering, N in every column.

E

Citations

Each cited claim classified, then verified: SUPPORTED, PARTIAL, MISMATCH, UNCERTAIN.

F

Contribution

An advocate and a skeptic each rate the contribution for the target journal.

G

Voice

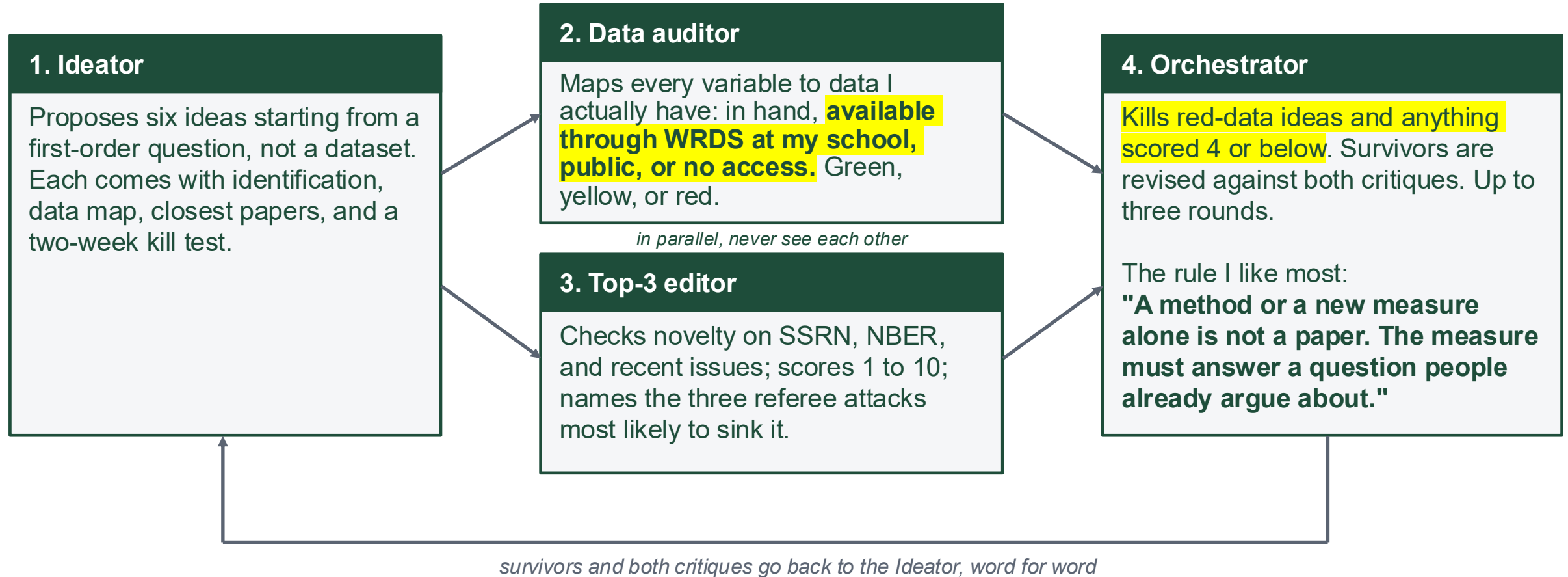
The core-voice rules: em-dashes, banned words, claim-first paragraphs.

H

Code audit

Merges, missing values, winsorization, clustering, hardcoded numbers.

A three-agent idea lab with kill rules



Module 4

AI as a research topic

What the finance and economics literature has learned so far, and where the open questions are

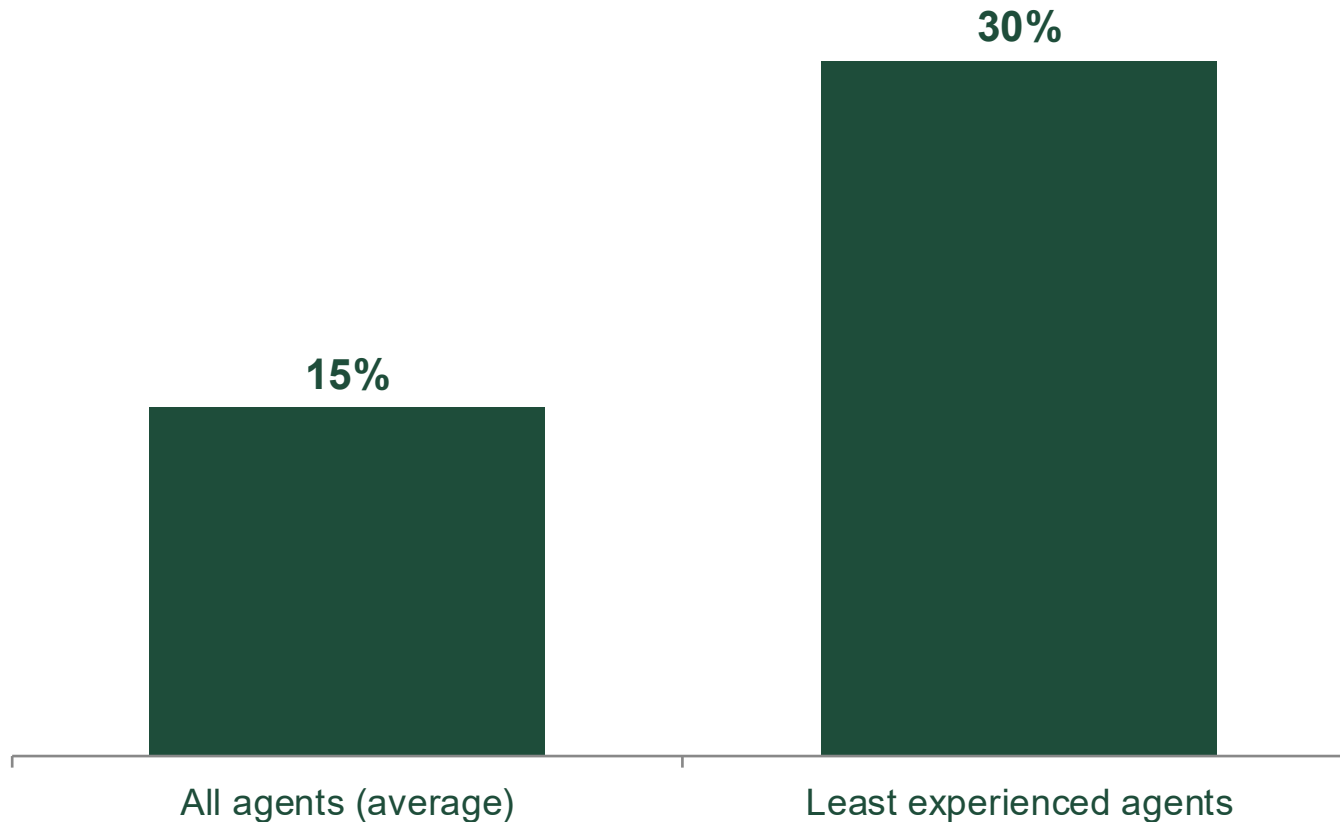
1. Measuring exposure to generative AI
2. Firm values and labor
3. Information production and markets
4. LLMs as measurement tools
5. Simulated agents and executive agents
6. What AI does to research itself

1. Markets priced generative AI within two weeks of ChatGPT

Eisfeldt, Schubert, Taska, and Zhang (2026), "Generative AI and Firm Values," *Journal of Finance*.

Question	Which firms did investors expect to gain from generative AI, and by how much?	What I would copy <ul style="list-style-type: none">▪ A clean event date plus a pre-period exposure measure▪ An exposure measure built by an LLM with a published rubric▪ Later work shows exposure predicts actual AI hiring (Schubert 2025)▪ Open: do these gains show up in investment, financing, and productivity?
Design	Firm exposure from occupation task scores times each firm's workforce mix. Sort into quintiles. Event: ChatGPT release, November 30, 2022.	
Finding	Long high-exposure, short low-exposure ("Artificial minus Human") earned 0.44% per day in the two weeks after release. Another run-up around GPT-4.	
Back of envelope	About \$1.4 trillion a year of US labor value exposed; \$91 billion in finance occupations alone.	

2. Who gains from AI assistance



- Customer support, 5,000+ agents: 15% average gain, about **30% for the least experienced, little or none for the most skilled** (Brynjolfsson, Li, and Raymond 2025, QJE).
- Patent lawyers, 133 in 11 firms: three months after the tool was removed, the **gains that survived sat with senior lawyers**; juniors kept roughly nothing on average (Autor et al. 2026, NBER WP 35720, unpublished).
- Why it matters to us: PhD training. If the tool does the hard parts, students may not build the skill.

Brynjolfsson, Li, and Raymond (2025, QJE): productivity gains from a generative AI assistant; the most skilled agents saw little or no benefit.

3. Natural experiments in who uses AI to process information

Italy's ChatGPT ban	Analysts issued fewer, less accurate forecasts during the ban ; market efficiency fell. Bertomeu, Lin, Liu, and Ni (2025, JAE).
ChatGPT outages	Trading volume fell during eight major outages : retail and institutional investors rely on it. Cheng, Lin, and Zhao (2025, JAE).
Seeking Alpha writers	Adopters published more and covered more firms; cheaper coverage improved liquidity even though each AI article moved prices less . Bradshaw, Ma, Yost, and Zou (2026, JAR).
AI in financial reports	Sharp rise in AI-generated text after ChatGPT, concentrated in firms with fewer reporting resources; clearer, but also more narrative management. Blankespoor, deHaan, and Li (2026, JAR).
A framework	Production, dissemination, and processing of information as three channels . Cao, Chen, Ma, and Srinivasan (2026, JAR).
Who captures the rents?	Larger hedge funds and those with AI-skilled staff earn more from LLM-extractable signals . Sheng et al. (2024)

4. Four ways LLMs become a measurement tool

Method	What it does	Finance examples	Main pitfall
Embeddings	Turn text or holdings into vectors; similarity, clusters, semantic axes	Text-based industries (Hoberg and Phillips 2016); asset embeddings from holdings (Gabaix et al. 2023); finance sentiment from books (Jha, Liu, and Manela 2026, RFS)	Interpreting dimensions
Classification and scoring	Label or score each document directly	Earnings-call sentiment (Chang et al. 2023); business complexity from footnotes (Bernard et al. 2026, TAR)	What does a "7 out of 10" mean?
Retrieval (RAG)	Find the relevant passages, then ask the model	Zoning rules (Bartik, Gupta, and Milo 2023); corporate culture from analyst reports (Li et al. 2026, RFS)	Missing passages, irrelevant ones confuse it
Simulation	Ask the model to answer as a person or agent would	Asset-class preferences (Fedyk et al. 2024); bank runs (Kazinnik 2026, MS)	Whose biases does it have?

4. Can LLMs read news and predict returns?

Headlines to returns

LLMs predict next-day returns from news headlines; **predictability appears only with more capable models (GPT-3.5 and later)**, and more so for complex news. Lopez-Lira and Tang (JFE).

Embeddings beat word counts

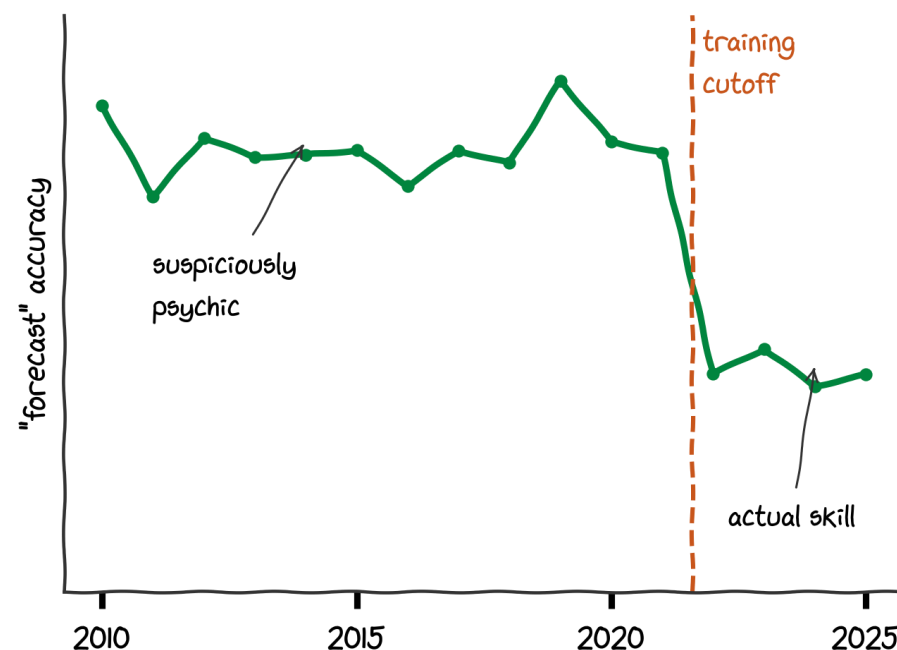
LLM embeddings of news predict the sign of returns better than simpler NLP methods. Chen, Kelly, and Xiu (2022). At the aggregate level: Chen, Tang, Zhou, and Zhu (2023).

But did the model already know?

Models have memorized economic and financial data from their training period, so forecasts on pre-cutoff data cannot separate skill from recall. Lopez-Lira, Tang, and Zhu (2025).

Implication

Every return-prediction result **needs a post-training-cutoff test** or a design that removes what the model could remember.



5. LLMs as survey respondents, forecasters, and executives

Predicting experiments

GPT-4 predicted relative **effect sizes of experiments better than human forecasters, but overstated absolute sizes.** Hewitt et al. (2024).

Expectations

LLM expectations from headlines track survey expectations and **reproduce known biases.** Bybee (2023). Synthetic forecasters compare well with **professional forecasters** (Hansen et al. 2024).

Preferences and runs

Persona-prompted models mirror demographic patterns in asset-class preferences but are **more transitive than people** (Fedyk et al. 2024). GPT agents reproduce **aggregate bank-run patterns** (Kazinnik 2026, MS).

Executives

Kothari and Ferris (2026, RCF) survey digital executives. The agents treat political contributions as a **strategic investment** for economic value and policy information.

Design question to settle first: **do you want the model to share human biases (to predict people) or to avoid them (to advise people)?** It depends on the application and can vary by setting (Ross, Kim, and Lo 2024).

5. How the CFO paper states its claim

Graham, Harvey, and Jha (2026), "CFOs Meet LLMs," SSRN 6920881.

Goal "Our goal is the prediction of CFO expectations. ... We make no claim that the model reasons in the same manner as the executive."

Concern 1 **Look-ahead:** the prior answers we supply are always dated before the survey cutoff.

Concern 2 **Memorization:** respondent-level answers were never released publicly, so the target is out of the model's corpus.

Concern 3 **Persistence:** controlling for the respondent's lagged answer shows the model is not just repeating it.

What I would copy

- **Say whether you claim prediction or mechanism**, in the first page
- **Name each objection before a referee does** and answer it with a design feature
- Choose a **target the model cannot have seen**
- Benchmark **against the naive forecast** (last answer)

6. What cheap generation does to the research market

288

return-predictability papers written by an LLM pipeline from mined signals, each with a plausible story. Novy-Marx and Velikov (2024), NBER WP 33363

3x / 5x

more publications and nearly 5x the citations for scientists using AI tools, while topic range contracts about 5%. Hao, Xu, Li, and Evans (2026, Nature)

7–17%

of peer-review text at four major ML conferences may have been substantially written by models. Liang et al. (2024, ICML)

Theory: as AI displaces human content, the pool of training data becomes less novel, an externality adopters do not internalize (Cai 2026, RFS).

Module 5

How these papers are designed, evaluated, and received

The questions every LLM paper gets, and the fixes referees accept

- What referees ask every LLM paper
- Look-ahead bias and memorization
- A validation playbook

The eight questions every LLM paper gets

1

Which model?

Exact version and date. **Open weights** or a time-stamped API model.

2

What prompt?

Full text in an appendix, plus how it was chosen and whether results move when it changes.

3

Look-ahead?

Could the model have **seen the outcome or the period in training**?

4

Validated how?

Against **human labels**, a survey, or a known benchmark. Agreement rates.

5

Better than what?

Dictionaries, FinBERT, simple controls. **What does the LLM add?**

6

What does it measure?

Is the construct the one you name, or tone, length, or firm size?

7

Replicable?

Will the same call give the same answer next year? Temperature, versions, code.

8

Economics?

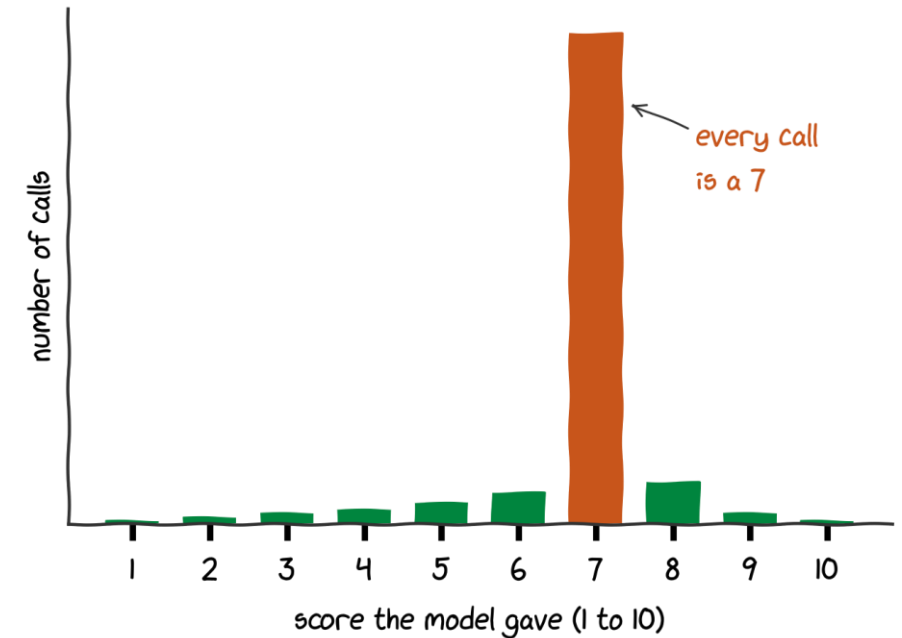
Magnitudes, a question people argue about, not "first to use LLMs for X." 2023 vs Now

Look-ahead bias: the first question, and the current fixes

Fix	Idea	Reference
Post-cutoff tests	Evaluate only on data after the model's training cutoff ; add live out-of-sample quarters as they arrive	Standard practice; see Lopez-Lira, Tang, and Zhu (2025) on memorization
Entity neutering	Remove names, tickers, dates, and identifying details so the model cannot recall what happened to that firm	Engelberg, Manela, Mullins, and Vulicevic, "Entity Neutering" (SSRN 5182756)
Anonymization trade-off	Measure how much information anonymization destroys along with the leakage	Wu, Yang, and Ying (2025), arXiv 2511.15364
Statistical tests	Test directly whether forecasts show lookahead bias	Gao, Jiang, and Yan (2025), arXiv 2512.23847
Out-of-corpus targets	Predict something never published, e.g., confidential survey responses	Graham, Harvey, and Jha (2026)
Fixed open models	Use open weights or time-stamped versions so the information set is known and reproducible	AFA disclosure policy for the JF

A validation playbook for any LLM-built variable

- 1 **Define.** Write the construct and a rubric with anchored levels before you see results.
- 2 **Hand-label.** Score a random sample yourself (or two RAs). Tune prompts only on this sample.
- 3 **Compare.** Agreement with human labels; correlation with an external benchmark such as a survey.
- 4 **Inspect.** Plot the score distribution. **Bunching at a few values signals a mapping problem.**
- 5 **Stress.** Change the prompt wording, the model, and the order of examples. Report what moves.
- 6 **Horse race.** Against dictionaries, FinBERT, and simple controls in the economic test.



Eisfeldt and Schubert: score a small random subset yourself and make sure each version of the prompt performs on it before scaling to the full data. Document these choices the way you would document a difference-in-differences design.

Module 6

AI in teaching

What I built with Claude Code for my undergraduate course, and how you can build the same in an afternoon

- Four tools I use every semester
- A study site and a student picker
- Other teaching uses

Four teaching tools, all built with Claude Code

Question banks

Multiple-choice and calculation questions from my slides and notes, in my format, with answer keys and explanations. I check every key.

A self-quiz study site

A single web page with quizzes and flashcards by lecture, so students test themselves before exams. Later grew into a study hub with writing and presentation modules.

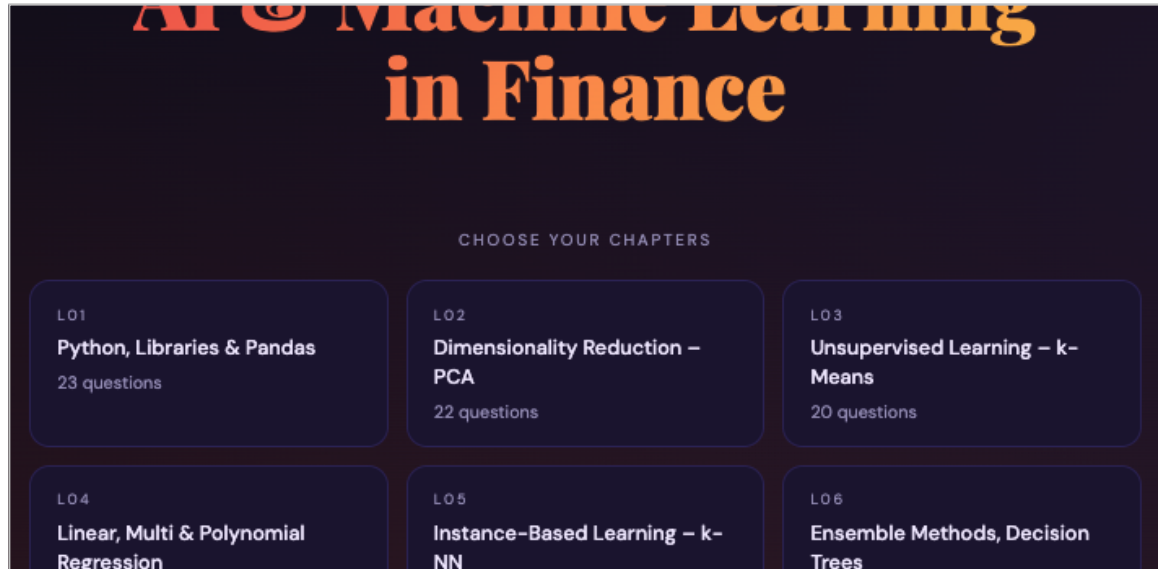
A random student picker

Calls on students at random for in-class questions, so participation is spread across the room and not only the front rows.

Multiple exam versions

Four versions of each exam with shuffled question order and answer options, with answer keys generated and checked by a script. Much harder to copy from a neighbor.

Two classroom tools, each built from one prompt



Self-quiz study site

- One HTML file: quiz and flashcard modes, a lecture filter; works on phones, no login
- **Prompt:** "Build a single-file study website from questions.json with quiz and flashcard modes, a lecture filter, and explanations after each answer."



Random student picker

- Picks at random without repeats; can lower the odds for students already called
- **Prompt:** "Build a one-page picker that reads a roster file, picks at random without repeats until everyone has been called, and runs offline."

Rosters are student records. The picker runs offline in the browser, so the roster never leaves my laptop; do not paste it into a cloud chatbot.

Other teaching uses worth trying

Trial-run an exam

Ask a model to answer as a student would. Misread questions show up before students see them.

Course evaluations

Summarize themes, pull concrete suggestions, keep the kind comments for your file.

Notes to slides

Loose notes into LaTeX or Markdown slides; check derivations for typos.

Table to figure

Turn a table from a paper into a clean figure for class in minutes.

Course chatbot

A Project loaded with your syllabus and transcripts so students can ask about covered material.

Simulations

Interactive cases, e.g., a loan negotiation where the model plays the borrower.

Module 7

Learning On Your Own

Resources and papers

Where to keep learning (all free)

Resource	Author	Best for
An AI Guide for Economists (adrienmatray-ai.com)	Adrien Matray	Setup, the core loop, failure modes, CLAUDE.md
Claude Code for Economists, five videos and posts (paulgp.substack.com)	Paul Goldsmith-Pinkham	Demos: data, EDGAR, big data, writing
AI for Professionals Who Don't Code (Claude Blattman site)	Chris Blattman	Prompts, voice files, skills, deep research
MixtapeTools on GitHub (scunning1975/MixtapeTools)	Scott Cunningham	Referee 2 audits, bibliography checks, decks
Trust, but Verify: toolkit and course (johnmbarrios.com)	John Barrios	The economics of verification
Generative AI for Economic Research; AI Agents for Economic Research	Anton Korinek	Use cases with ratings
Generative AI and Finance, Annual Review of Financial Economics	Eisfeldt and Schubert	Literature, methods, best practice
Deep Learning for Economists, NBER WP 32768	Melissa Dell	Methods foundations

Reading list: published LLM papers, 2023 onward (1 of 2)

Area	Paper
Survey	Eisfeldt and Schubert (2025), "Generative AI and Finance," Annual Review of Financial Economics
Methods	Korinek (2023), "Generative AI for Economic Research," Journal of Economic Literature
Methods	de Kok (2025), "ChatGPT for Textual Analysis? How to Use Generative LLMs in Accounting Research," Management Science
Methods	Huang, Wang, and Yang (2023), "FinBERT: A Large Language Model for Extracting Information from Financial Text," CAR
Methods	Liu et al. (2024), "Lost in the Middle: How Language Models Use Long Contexts," TACL
Firm value	Eisfeldt, Schubert, Taska, and Zhang (2026), "Generative AI and Firm Values," Journal of Finance
Labor	Brynjolfsson, Li, and Raymond (2025), "Generative AI at Work," Quarterly Journal of Economics
Labor	Eloundou, Manning, Mishkin, and Rock (2024), "GPTs are GPTs," Science
Theory	Cai (2026), "Generative AI and Data Quality," Review of Financial Studies
Culture	Li, Mai, Shen, Yang, and Zhang (2026), "Dissecting Corporate Culture Using Generative AI," Review of Financial Studies

Reading list: published LLM papers, 2023 onward (2 of 2)

Area	Paper
Asset pricing	Lopez-Lira and Tang (forthcoming), "Can ChatGPT Forecast Stock Price Movements?" Journal of Financial Economics
Trading	Cheng, Lin, and Zhao (2025), "Does Generative AI Facilitate Investor Trading? Evidence from ChatGPT Outages," JAE
Analysts	Bertomeu, Lin, Liu, and Ni (2025), "The Impact of Generative AI on Information Processing: Evidence from the Ban of ChatGPT in Italy," JAE
Disclosure	Blankespoor, deHaan, and Li (2026), "Generative AI in Financial Reporting," JAR
Disclosure	Bernard, Blankespoor, de Kok, and Toynbee (2026), "Using GPT to Measure Business Complexity," The Accounting Review
Intermediaries	Bradshaw, Ma, Yost, and Zou (2026), "Generative AI Use by Capital Market Information Intermediaries," JAR
Framework	Cao, Chen, Ma, and Srinivasan (2026), "Generative AI in Capital Markets," JAR
Banking	Kazinnik (2026), "Bank Run, Interrupted: Modeling Deposit Withdrawals with Generative AI," Management Science
Executives	Kothari and Ferris (2026), "Strategic Generosity: The Business of Political Contributions," Review of Corporate Finance
Credit	Wu, Dong, Li, and Shi (2025), "Unleashing the Power of Text for Credit Default Prediction," EJOR
Science	Stokel-Walker and Van Noorden (2023), "What ChatGPT and Generative AI Mean for Science," Nature



mjha91.github.io
mjha@gsu.edu

Thank You