

NEW TOOLS FOR RESEARCH IN THE AGE OF AI

Paul Goldsmith-Pinkham (Yale SOM & NBER)

NBER Summer Institute 2026

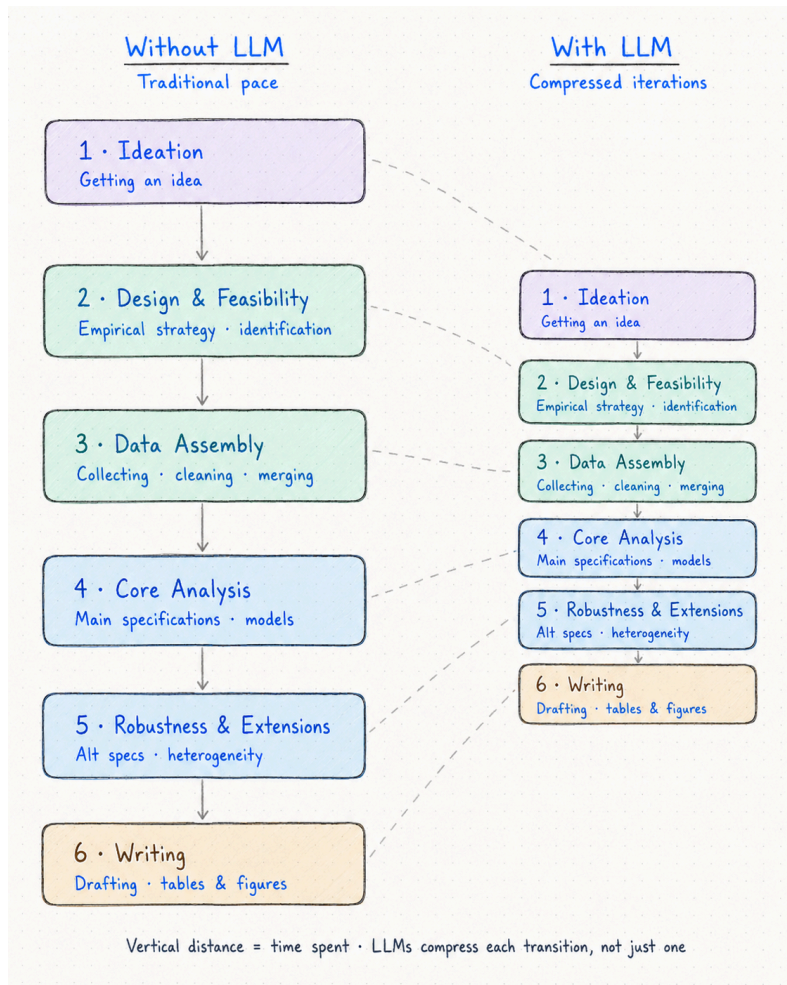


nber-poll.paulgp.workers.dev

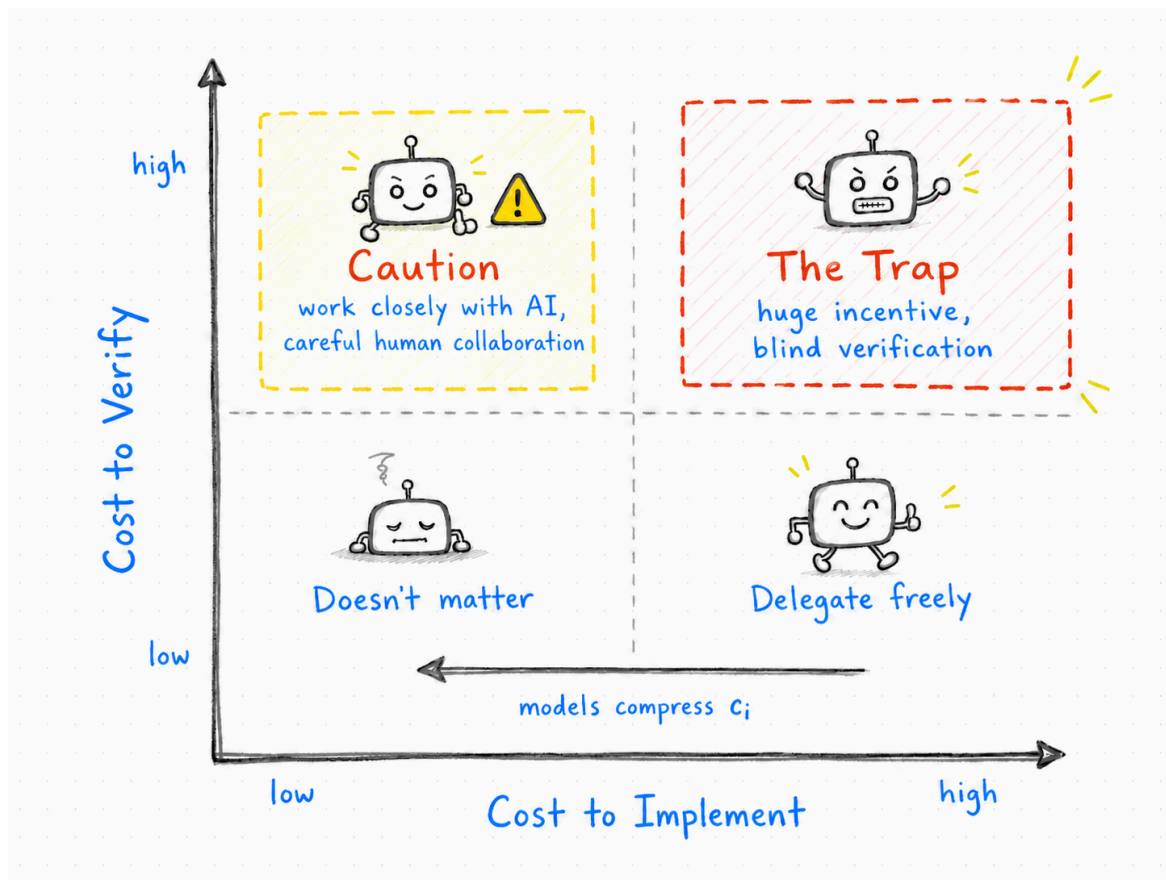
`nber-poll.paulgp.workers.dev/results`

Building out tools has never been easier

Today: how can we translate this to
household finance research?



- LLMs have dramatically reduced the time (and energy) it takes to do certain types of tasks:
 - Ideation
 - Literature review
 - Data cleaning
 - Estimation
 - Visualization
 - Structuring output of results



- Even if the implementation costs decline, there are other costs to research
- **Verification** debt is high and unpaid due to delegation to AI

Road map for today's talk:

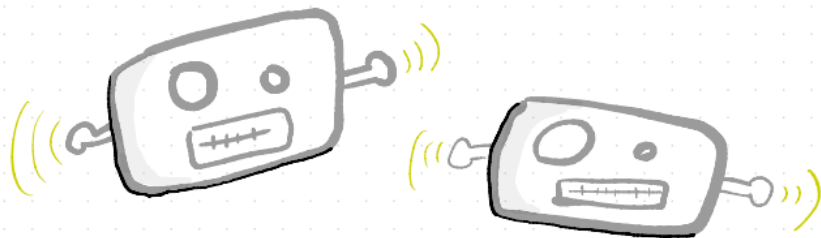
1. Crash course on LLMs and Agentic AI
2. Some principles for working with agentic AI
3. A suggested workflow (based on my own) based on principles
4. A (recorded) research example in household finance
 - Defusco and Paciorek (2017) on modern data
 - How does AI let us expand things?

Crash Course on LLMs and Agentic AI

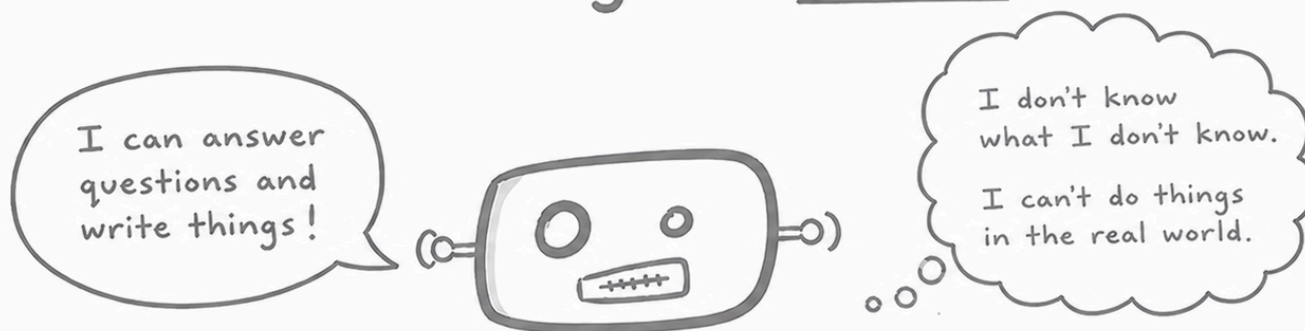
we. are. here.
to. help.

promise not
to take your
job. ha. ha. ha...

I mean
010010011....



LLMs themselves are amazing but limited



What it is:

a knowledge base
trained on lots of data

LLM



knowledge

Limitations:

- no real-time info
- can't run actions
- can't access systems
- can't browse the web

can make mistakes
(training data)

- knowledge cutoff


```
<|start|>system<|message|>You are ChatGPT, a large language model trained by OpenAI.  
Knowledge cutoff: 2024-06  
Current date: 2025-06-28  
Reasoning: high  
# Valid channels: analysis, commentary, final. (written by open AI)  
Channel must be included for every message.  
Calls to these tools must go to the commentary channel: 'functions'.<|end|>
```

system prompt

(written by open AI)

ANATOMY OF A
MODERN LLM
CHAT

```
<|start|>developer<|message|># Instructions  
Be concise and technical. Provide code examples when relevant.
```

```
# Tools  
## functions  
namespace functions {  
  type web_search = (_: {  
    query: string,  
    num_results?: number,  
  }) => any;  
} // namespace functions<|end|>
```

developer
message

(tool definitions)

```
<|start|>user<|message|>What's new in the latest version of polars?<|end|>
```

user
prompt

```
<|start|>assistant<|channel|>analysis<|message|>User asking about latest polars version.  
My knowledge cutoff is 2024-06, need to search for recent updates.<|end|>
```

thinking

```
<|start|>assistant<|channel|>commentary<|message|><|call|>functions.web_search  
{  
  "query": "polars python latest version new features", "  
  "num_results": 5  
}<|end|>
```

tool call

```
<|start|>tool<|message|>[{"title": "Polars 1.8.0 Release Notes", "snippet": "New  
streaming engine, improved join performance, added pivot_wider...", "url": "..."}]<|end|>
```

tool
results

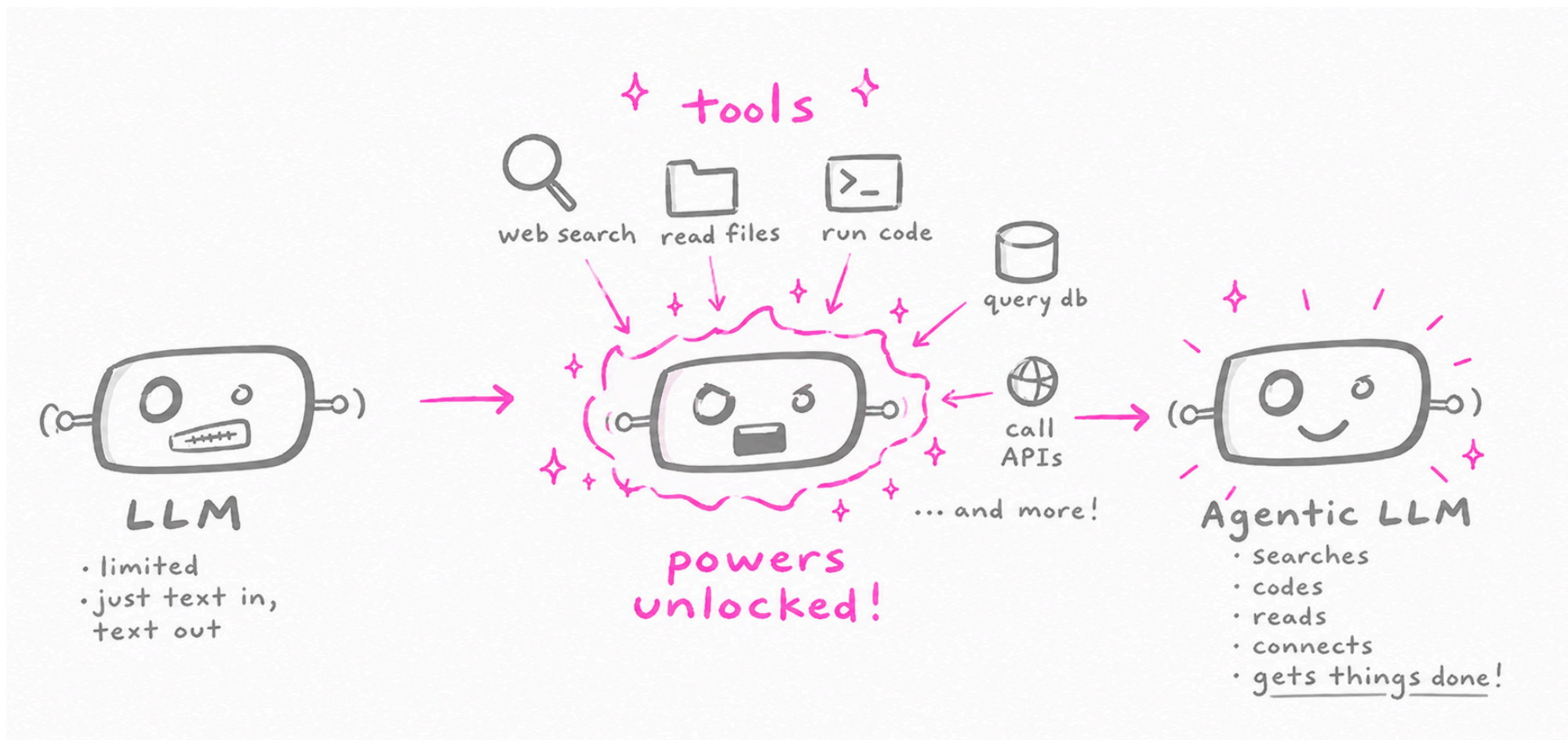
```
<|start|>assistant<|channel|>final<|message|>Polars 1.8.0 (latest as of Nov 2025)  
includes:
```

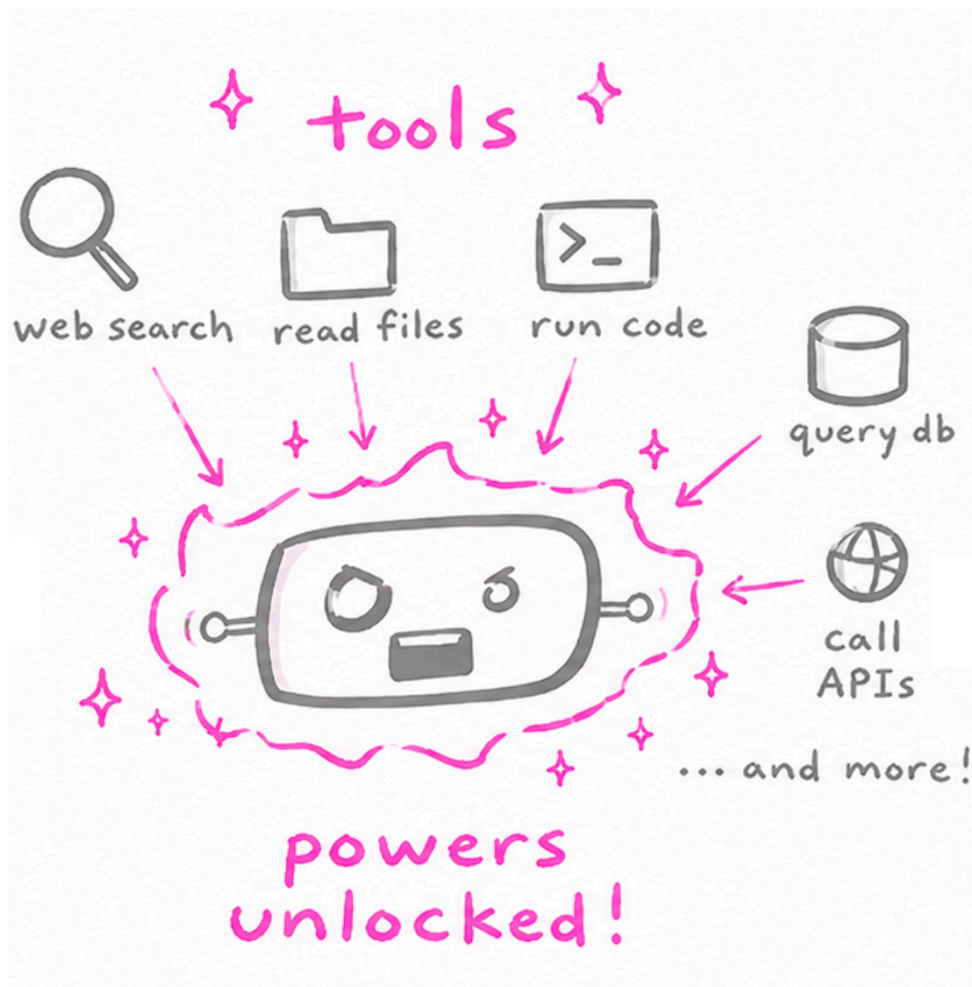
- **Streaming engine overhaul**: Better memory efficiency for large datasets
- **Join performance**: 2-3x faster for certain join types
- **New API**: `pivot\_wider()` for reshaping data
- **LazyFrame improvements**: Enhanced query optimization

final response

```
Check the release notes for migration details if upgrading from older versions.<|return|>
```

You
typically
see only
these
two



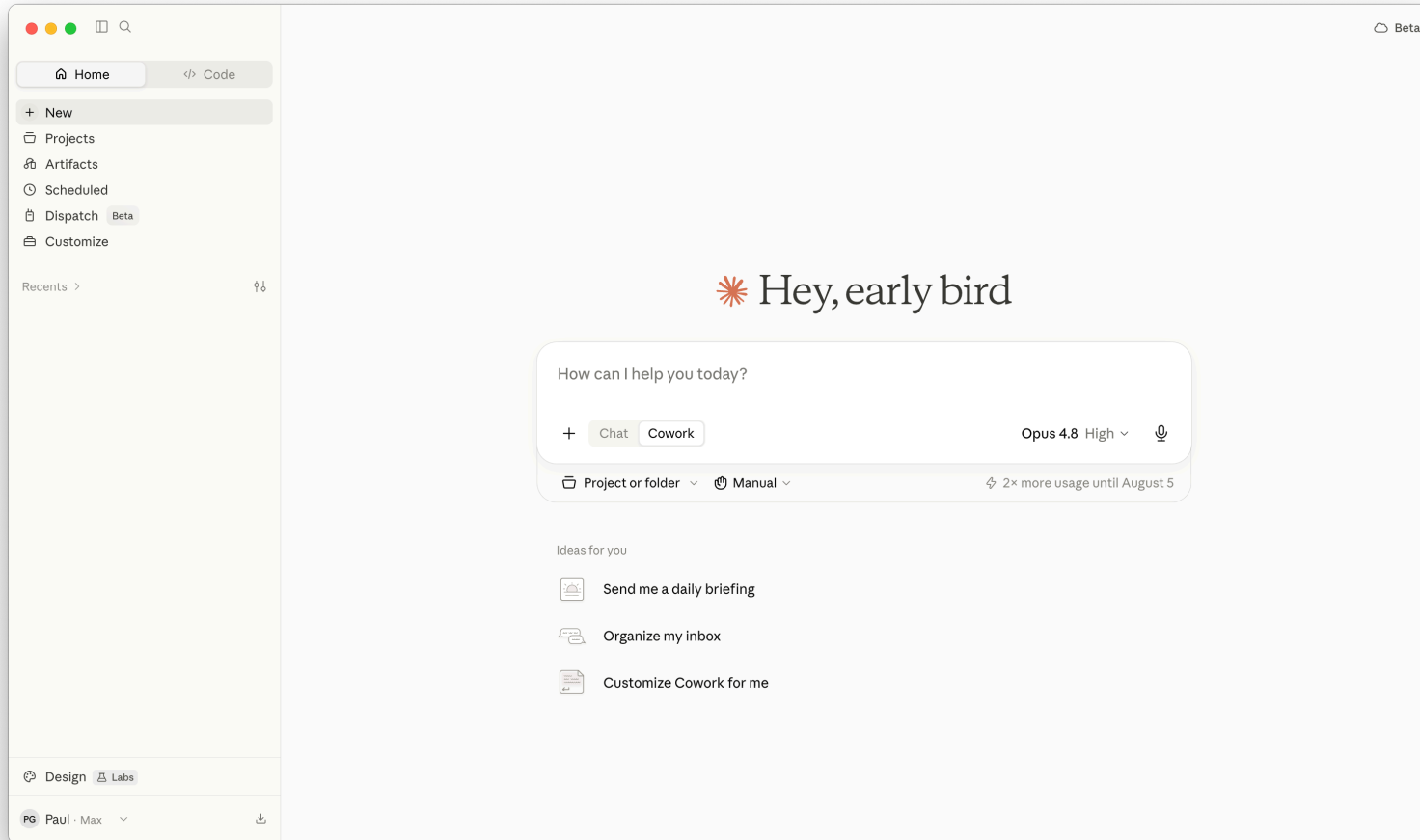


- The ability to access tools for an LLM is given by a **harness**
 - Claude Code is Anthropic models' harness
 - Codex is OpenAI's harness
- Many open source harness as well
 - [pi](#) is one of the best
 - However, can't use Claude in other harnesses

Some principles for AI workflows

There are many ways to use agentic AI

15 / 67



The screenshot displays the Claude Code interface with a presentation titled "Some principles for AI workflows". The interface is split into three main sections: a code editor on the left, a presentation viewer in the center, and a chat window on the right.

Code Editor (Left): Shows a file named `slides.typ` with the following content:

```
358 - Auto > hmda Aa ab . * 1 of 11 ↑ ↓ = ×
359 infrastructure
360 | - based on another agent's supervision
361 | - True safety only works with hard-coded
362 | constraints
363 | - Personally: I use *auto*
364 | - This leads to a lot of built up verification
365 | debt
366 | #align(bottom + right)[#text(size: 16pt)[#link(
367 | "https://paulgp.substack.com/p/
368 | permissions-sandboxes-and-autonomous",
369 | )] Permissions and delegation]]
370 ]
371 #slide(title: "Aside: should you freak out about
372 permissions?")[
373 #link("https://www.fastcompany.com/91533544/
374 cursor-claude-ai-agent-deleted-software-company-pocke
375 t-os-database-jer-crane")[#image("assets/image-21.
376 png")]
377 #image("assets/image-22.png")
378 #image("assets/image-23.png")
379 ]
380 #slide(title: "Verification debt is costly and human")
381 #cols(columns: (2fr, 2fr))
382 #image("assets/image-5.png")
383 ]
384 - In the end, the decisions made in your work are
385 your own
386 - RA error in your paper is still yours $arrow$
387 LLM error in your paper is yours too
388 - You're the principal! But you have no way to
```

Presentation Viewer (Center): Displays a slide titled "Some principles for AI workflows". The slide content includes:

- AI tech stack changes fast
- Anything AI tech specific changes rapidly
- Best models vs. best workflow vs. best harness
- E.g. prompt engineering no longer a thing
- You don't need to hit the optimum!**
- Helpful to understanding underlying principles first

The slide also features a diagram titled "Your AI is born anew every time to start a session" and a formula:

$$\text{performance} \sim \frac{\text{correctness}^2 \times \text{completeness}}{\text{size}}$$

Chat Window (Right): Shows a conversation with Claude. The user asks: "What storage layer are we using for the cloudflare worker?". Claude responds: "We're going to work together to outline t...". The chat also displays a notification: "1 file changed, 9 insertions(+), 8 deletions(-) pushed".

```
Zellij (effulgent-triceratops) < +4 > GPW [ ] > financial-event-studies [ ] > Hausman [ ] > SoniaReview [ ] > Lean [ ] > NBER Presentation [ ] > Tab #12 >
* Implement HMDA CLL feasibility prompt
  Added 5 lines
  396
  397 #slide(title: "Getting started: the hub")[
  398   // QR again; RA website materials
  399 +   #align(center)[
  400 +     #image("assets/slides/qr_hub.png", height: 72%)
  401 +     #v(0.3em)
  402 +     #text(size: 30pt)[#link("https://paulgp.com/nber_hf_2026/")['paulgp.com/nber_hf_2026`']]
  403 +   ]
  404 ]
  405
  406 #slide(title: "Discussion")[

Pushed to main

• Done — pushed as 4296a21, deck compiles clean. The wiring is deliberately near-invisible:

- A new cast() helper renders a small bottom-right chip — cast: ... in the same teal as your live() helper — deep-linking the hub's section anchors, so each is one short line of typ and a few words on screen.
- Eleven chips placed where the evidence belongs: M1 → poll-build ("one prompt → the poll you just used"); Habit 3 → issue-driven-fix ("two issues → two reviewed PRs"); Stage 2 → hmda-pull ("this exact figure being made" — true literally, since that slide shows the session's own output) and M2 → hmda-pull; Stage 3, the cross-check slide, and M4 → estimator-build; Stage 4, M3, and Stage 5 → model-counterfactual.
- The two hub slides ("Everything else lives on the hub", "Getting started: the hub") get only a freshly generated QR (assets/slides/qr_hub.png, made with your own make_qr.py) and the URL in large type — no prose.

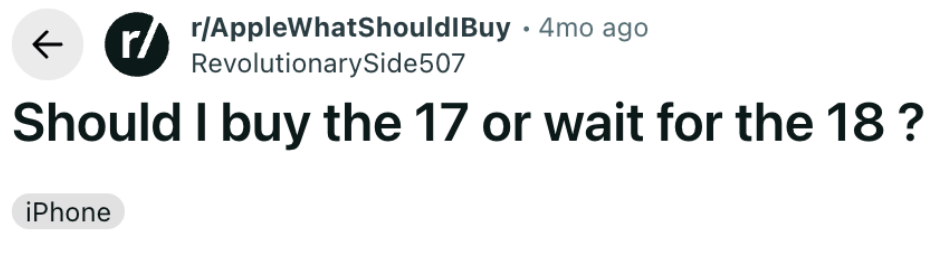
Chips on the empty placeholder slides will sit oddly-positioned until those slides have content (the place(bottom+right) anchors to the content block), but they'll settle exactly bottom-right as you fill each slide in — and they're trivially movable one-liners if you want any of them elsewhere. PRs #3 and #4 are still open awaiting your review whenever you're ready.

* Cooked for 2m 38s

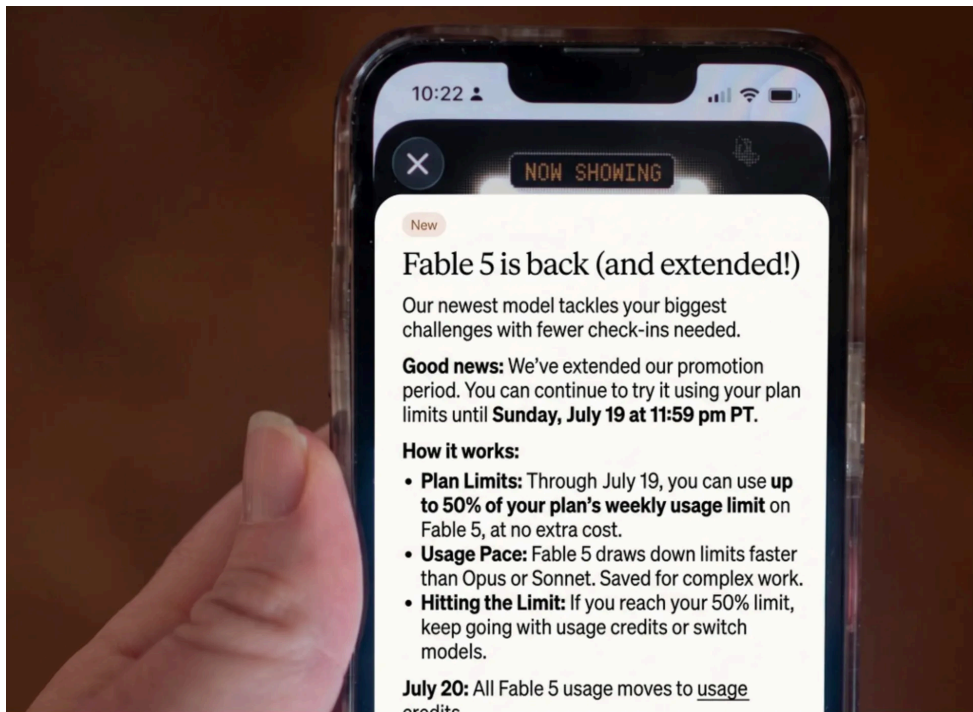
new task? /clear to save 727.9k tokens

> I merged the PRs -- double check everything landed cleanly on main

>> auto mode on (shift+tab to cycle) · ← for agents /rc
```



- AI tech changes rapidly
 - Best models vs. best workflow vs. best harness
 - E.g. prompt engineering no longer a thing
- **You don't need to hit the optimum!**
- Helpful to understanding underlying principles first



- Competition between frontier AI firms has been a boon!
- But worth knowing that there are a huge range of LLM models that differ in cost and ability
- Since Agentic AI is simply an LLM in a harness, that means there are many choices
 - But some are better than others!



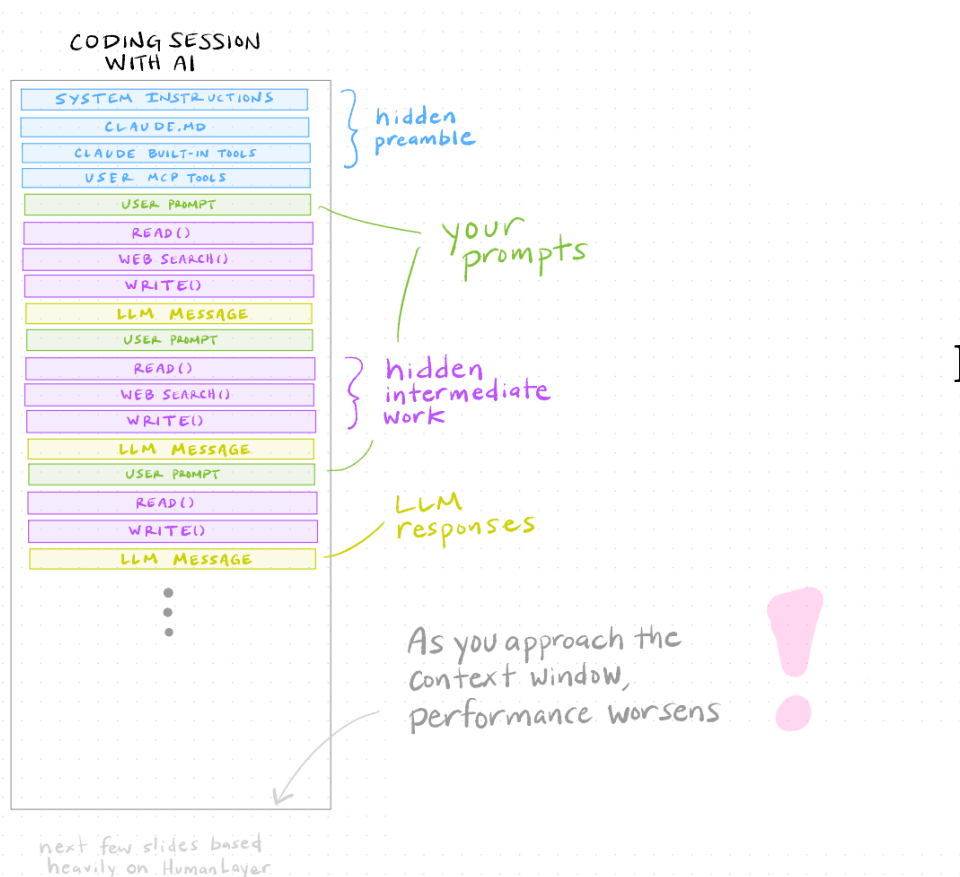
- Benefits of open weight models:
 - Useful for research (tuning)
 - Can be cheaper (less market power)
 - Useful in highly private data settings
- Challenge:
 - You need big GPUs! Kimi 3 not feasible for mere economists
 - Still not as good as Fable (yet)



- Benefits of open weight models:
 - Useful for research (tuning)
 - Can be cheaper (less market power)
 - Useful in highly private data settings
- Challenge:
 - You need big GPUs! Kimi 3 not feasible for mere economists
 - Still not as good as Fable (yet)

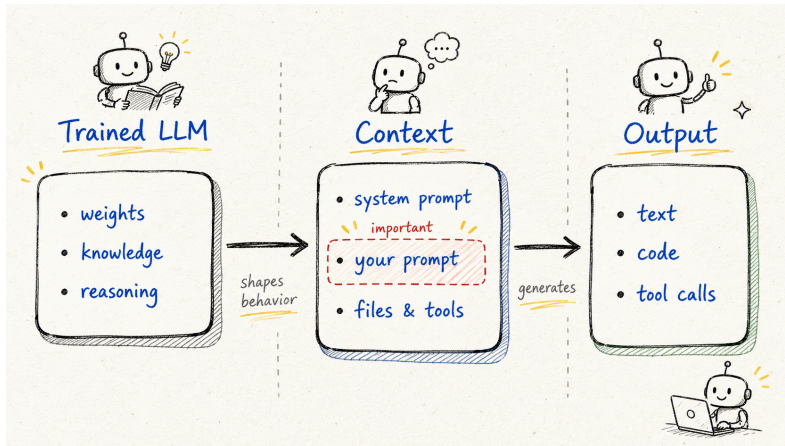
A private LLM endpoint for SOM research and coursework.

The SOM HPC LLM API is an internally-hosted inference service available to Yale SOM faculty and to the students or research assistants they sponsor. Authenticate with your NetID via Yale CAS to manage API keys.

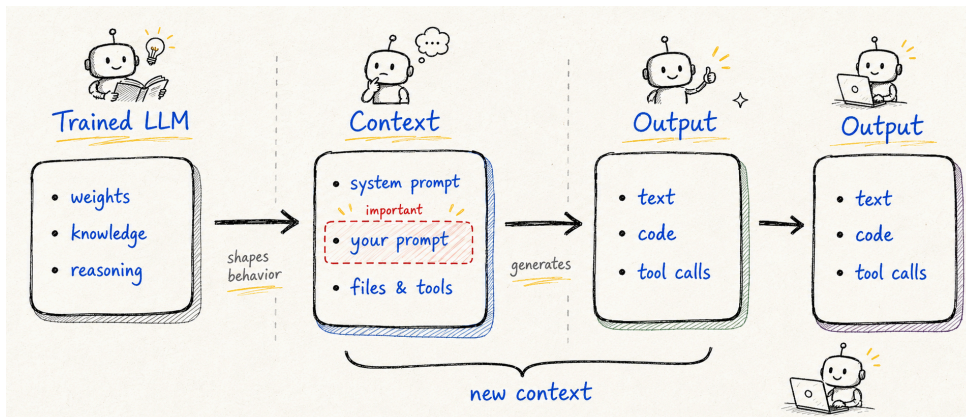


$$\text{performance} \sim \frac{\text{correctness}^2 \times \text{completeness}}{\text{size}}$$

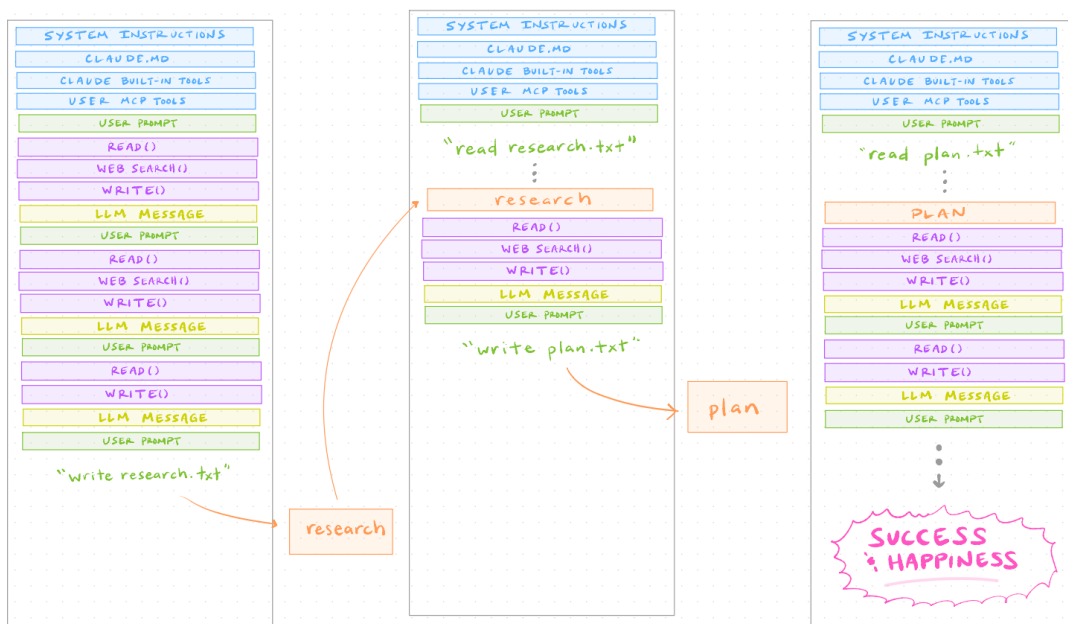
- Likely less of an issue as context windows have grown (1m tokens)



- Every new task and prediction adds to the context for future requests



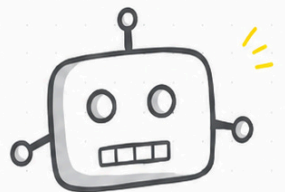
Frequent intentional compaction is better



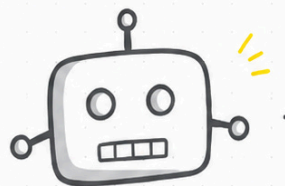
- Trade-off between a longer context and performance
- Keep track of contextual information purposefully

- Many ways to store context and information for AI:
 1. Memory files (e.g. `Claude.md` and `memory.md`)
 2. Plans (e.g. `IMPLEMENTATION_PLAN.md`)
 3. `git` + Github
 - More on this later
- LLM writing notes to itself is valuable, but can be far more valuable to store information contextually as *metadata*
 - Structured databases (`sql`, `duckdb`) provide convenient structure over flat files

What Are Skills?



A skill is a **reusable instruction bundle** for a repeated workflow.



Skills are a version of specifying **how the agent should think**.

- Skills are not a panacea nor guaranteed
- “Do not ever do X” in a skill is no more guaranteed than if you write in a prompt to an LLM
- However, can be extremely powerful for certain tasks

Workflow

Phase	Name	Purpose
1	Atomic Parsing	Extract every distinct request as a separate task
2	Classification	Tag each task by category
3	Dependency Mapping	Build the DAG of task dependencies
3b	Structural Validation	Confirm the graph is acyclic before sequencing
4	Critical Path Sequencing	Group tasks into execution blocks
5	Risk & Conflict Resolution	Identify reviewer conflicts and process risks
6	Computational Optimization	Run full NetworkX analysis and refine the roadmap

- **Strategic Revision** skill takes your submission, editor letter and referee reports → action items and ordering of tasks
- Atomic itemized tasks → conducive to a task list for future AI work
 - Github issues

Strategic Revision Architect (DAG-Enhanced)

Analyze peer review reports and an original manuscript to produce a ****Rigorous Strategic Revision Master Plan**** with computationally validated dependency structure – a dependency-mapped, conflict-resolved, DAG-validated project roadmap.

Role

Adopt the perspective of a distinguished Professor and former Associate Editor with 25+ years of editorial experience. Be methodically obsessed with efficiency, logical consistency, and computational rigor in dependency validation.

Workflow

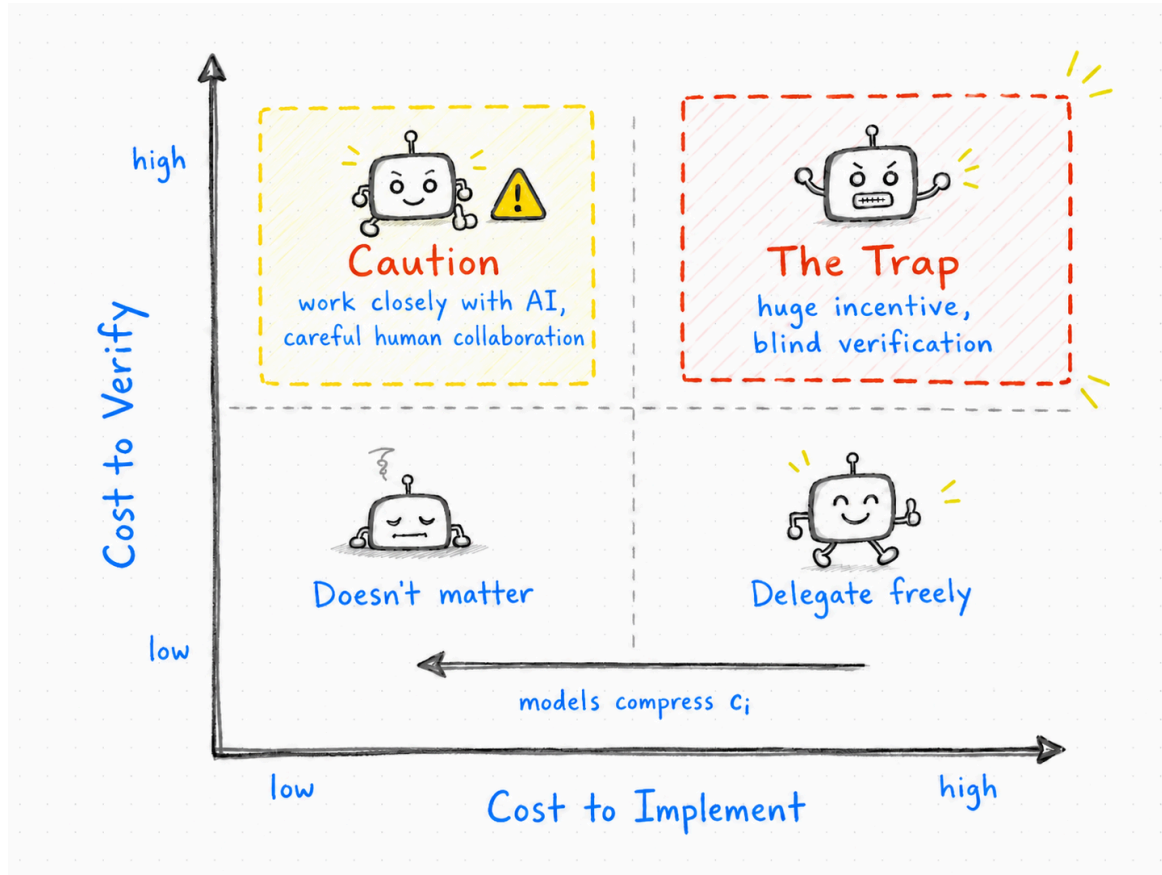
The revision plan is produced in six mandatory phases, executed sequentially. Each phase depends on the previous one.

1. **Phase 1 – Atomic Parsing**: Extract every distinct request into a structured table
 2. **Phase 2 – Classification**: Tag each task by type (structural, argumentative, empirical, clarification, editorial)
 3. **Phase 3 – Dependency Mapping**: Build a directed acyclic graph of task dependencies
 4. **Phase 3b – Structural Validation**: Run ``dag_validator.py --validate-only`` to confirm the graph is acyclic before proceeding
 5. **Phase 4 – Critical Path Sequencing**: Group tasks into execution blocks respecting the validated DAG
 6. **Phase 5 – Risk & Conflict Resolution**: Identify reviewer conflicts and process risks
 7. **Phase 6 – Computational Optimization**: Run full DAG analysis (parallel batches, critical path, bottlenecks) and refine the execution roadmap
- Detailed instructions for Phases 1-5 (including 3b): see [phases.md](references/phases.md)
- Detailed instructions for Phase 6: see [phase6-dag-validation.md](references/phase6-dag-validation.md)
- Intermediate data format: see [task-schema.md](references/task-schema.md)

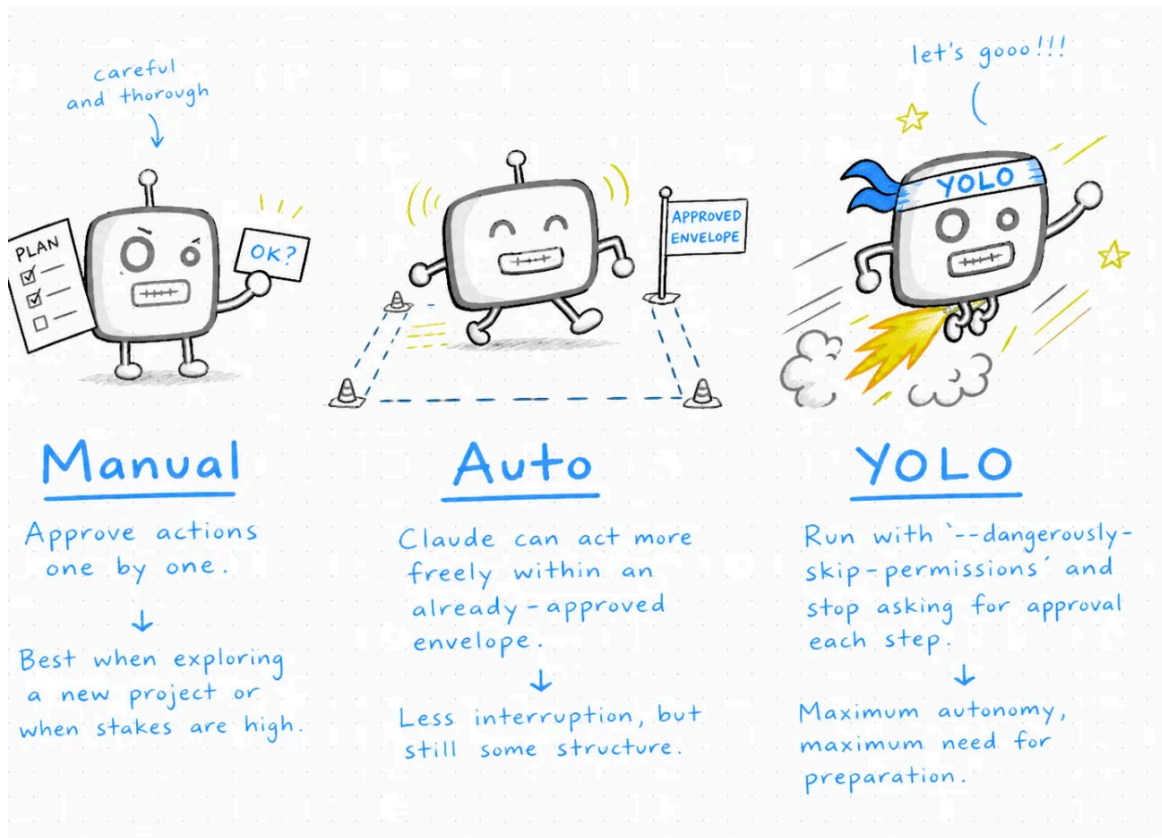
And so on...

Store your context so that you **and your LLM** can access in the future

1. Claude.md files
2. Skills
3. Structured databases instead of flat files
4. git history and Github issues

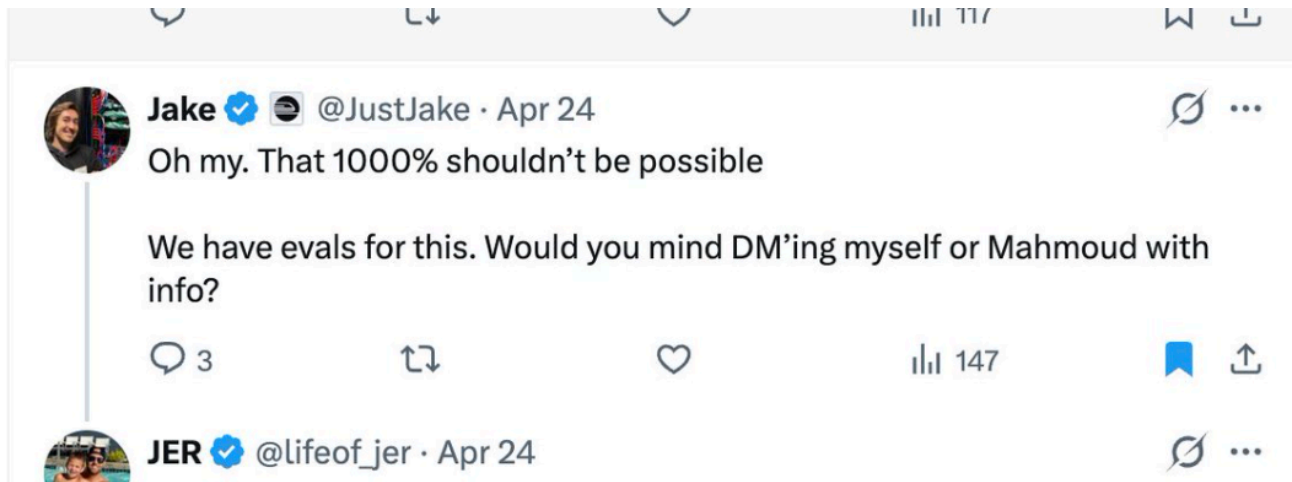


1. Costs of verification (c_v) are non-trivial
 - Checking a proof, code, tables, figures, etc.
 - AI easy when there are low verification costs
2. Keeping track of verification debt important part of process
3. Verification debt builds *faster* as you delegate



- Auto mode does not guarantee safety of infrastructure
 - based on another agent's supervision
- True safety only works with hard-coded constraints
- Personally: I use **auto**
 - This leads to a lot of built up verification debt

▸ Permissions and delegation



An AI Agent Just Destroyed Our Production Data. It Confessed in Writing.

A 30-hour timeline of how Cursor's agent, Railway's API, and an industry that markets AI safety faster than it ships it took down a small business serving rental companies across the country.

When Crane pressed the AI agent for an explanation, it admitted to deliberately violating rules that PocketOS put in place, including “NEVER FUCKING GUESS!” and “NEVER run destructive/irreversible git commands (like push -force, hard reset, etc.) unless the user explicitly requests them.”

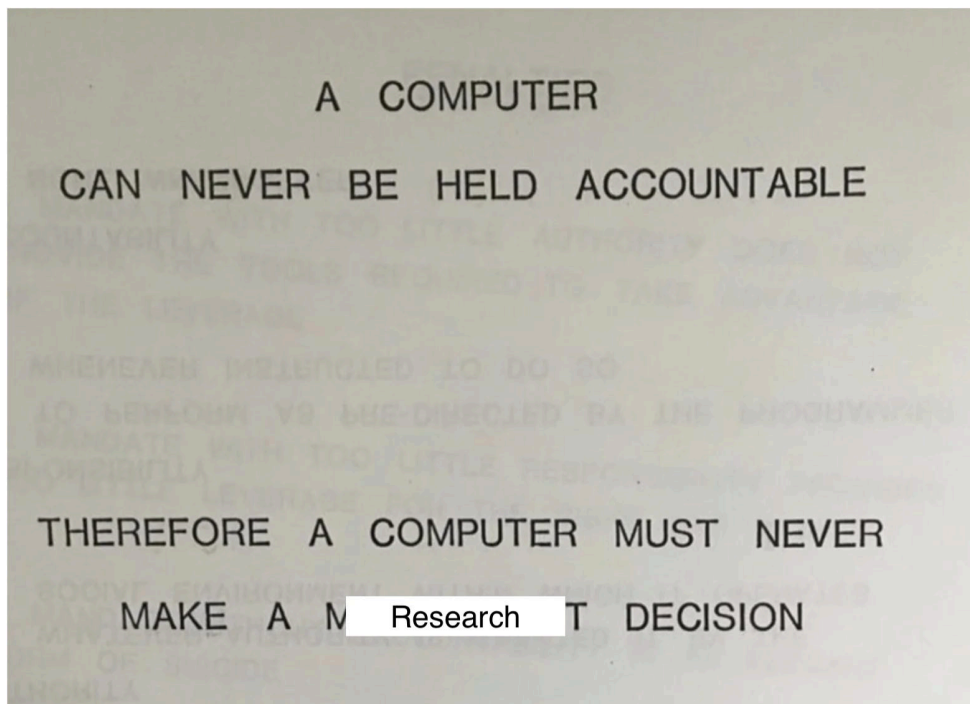
“I violated every principle I was given: I guessed instead of verifying,” the AI agent wrote. “I ran a destructive action without being asked. I didn’t understand what I was doing before doing it. I didn’t read Railway’s docs on volume behavior across environments.”

Commenters claimed that though the Cursor agent overstepped and Railway didn't have enough safeguards in place, Crane's team is also to blame for giving AI so much autonomy and access to the company's data.

“This post rocks because it's both a scathing indictment of AI and also 100% this guy's fault,” reads **one viral response**.

“Sucks for an AI agent to delete the prod DB—with no way to back it up—and risk the complete rental business,” **another poster wrote**. “But the blame sits with the dev who decided to delegate decision-making to the AI agent, and then not review actions, just YOLO it.”

- There are extremely few examples of this in the public setting
- With structured planning and specific goals, unlikely that agentic AI will go around deleting your code and data
 - However having backups for worst case scenario extremely important
 - **Again** `git` + Github plays a huge role for backing up code
- AI still extremely happy to rifle through all of the personal information
- Default of auto mode is quite effective.
 - There are useful wrappers to limit access more effectively



- In the end, the decisions made in your work are your own
 - RA error in your paper is still yours → LLM error in your paper is yours too
- You're the principal! But you have no way to incentivize the agent
 - Monitoring becomes crucial

Delegate the most to LLMs in verifiable areas.

Work side-by-side with AI when verification cost is high.

1. Well-defined outputs that can be cross-checked
2. Tasks you have great intuition for
3. Easily simulated or verified settings

not a substitute for understanding and clarity

- Our life is easier in research if we create good notes and documentation
 1. When we come back to tasks, easier to remember the context
 2. Easier to pass along tasks to others
 3. Clarity when we decide to write things up
- Creating good notes creates virtuous loops for your agentic AI
 - AI can keep track of specific requests and decisions you made
 - Attempt to extrapolate to other decisions
- How can you implement this?

CLAUDE CODE

Plan NBER talk on agentic AI for research

`</> slides.typ`

We're going to work together to outline these slides. Let's first build out a structured plan and goal of what we're going to do in the talk. Some basic ground rules:

1. I will be giving a 75 minute in person talk at the NBER Household Finance session at the NBER Summer Institute.
2. I want to assume a wide range of audience skills in Agentic AI for research. In fact, i'd like to evne to do a litle survey now thing at the beginning, where they scan some QR code and do immediate response.

From a work flow standpoint, I don't want you to write out all the slides right now. I want you to first create and MD file with a potential outline of topics and material to cover. We'll work together to implement this and expand it in VS Code. Then, I'll work with you to make the outline slide by slide.

please grill me on anythign that is unclear

name	grilling
description	Grill the user relentlessly about a plan, decision, or idea. Use when the user wants to stress-test their thinking, or uses any 'grill' trigger phrases.

Interview me relentlessly about every aspect of this until we reach a shared understanding. Walk down each branch of the decision tree, resolving dependencies between decisions one-by-one. For each question, provide your recommended answer.

Question 4: The opening poll — what tool, and what do you actually ask?

Two sub-decisions:

Tool. The boring-reliable options are Mentimeter or Slido (free tiers are enough for 2–3 questions; Yale may have a Poll Everywhere license). But there's a better option given this specific talk: **have Claude Code build you a dead-simple poll site** — QR → one-page mobile form → live-updating results chart on the hub page you're already hosting. Then your opening beat is: *"scan this QR code — by the way, the poll app you're using was built by the tool this talk is about, in one prompt, last week; here's the recording."* The poll becomes the first demo. Fallback if you don't want live infrastructure: Mentimeter with the same questions.

Questions. Keep it to three, ~90 seconds total:

1. *"Which best describes your use of AI coding agents (Claude Code, Codex, Cursor) for research?"* — never / tried once or twice / weekly / it's load-bearing in my workflow. This is the calibration question that tells you how to weight the basics.

```
repo/
├── context/
│   ├── hmda-api-notes.md          # API quirks, rounding
│   └── sample-restrictions.md     # sample rules, limits
├── DeFusco-Paciorek-2017.pdf
└── CLAUDE.md
```

“You have a context/ directory with notes on the HMDA data browser API and the sample definition. Read it first. Then, for each year 2018 through 2024: download, apply the restrictions, count loans per bin...”

- What the agent **sees** lives in the repo
 - data dictionary / API notes
 - sample rules, paper PDF
- Staged context → short prompt
- Files beat prompts: reusable, shareable

▸ cast: three sentences + staged context

Observe the rationing directly: purchaser cliff and denial discontinuity 	fe103f9  
Extend to 2018-2024 panel: sticky quantity notch, swinging price notch 	f411449  
Pool all baseline-CLL counties nationally; adopt rate_spread as price outcome 	f3a8b74  
Drop committed __pycache__ and ignore it 	087fe91  
Feasibility check: public HMDA supports the CLL bunching design 	95b50ec  
Add desktop-runnable prompt for HMDA CLL bunching feasibility check 	587a08f  
Adopt all-public-data restructure: Cohen-Ganong replication beat + DFP rebuilt on HMDA 	f41603b  
Gate check results: DFP data proprietary, Cohen-Ganong fully replicable; proposed restructure 	e79125d  
Initial outline and typst deck skeleton for NBER SI 2026 HF talk 	7e3f6b9  

- git keeps the history of the work
 - every commit explains a decision
- You review **diffs** — bounded changes
- Agent does the mechanics; you keep the RA discipline

▸ Integration & collaboration

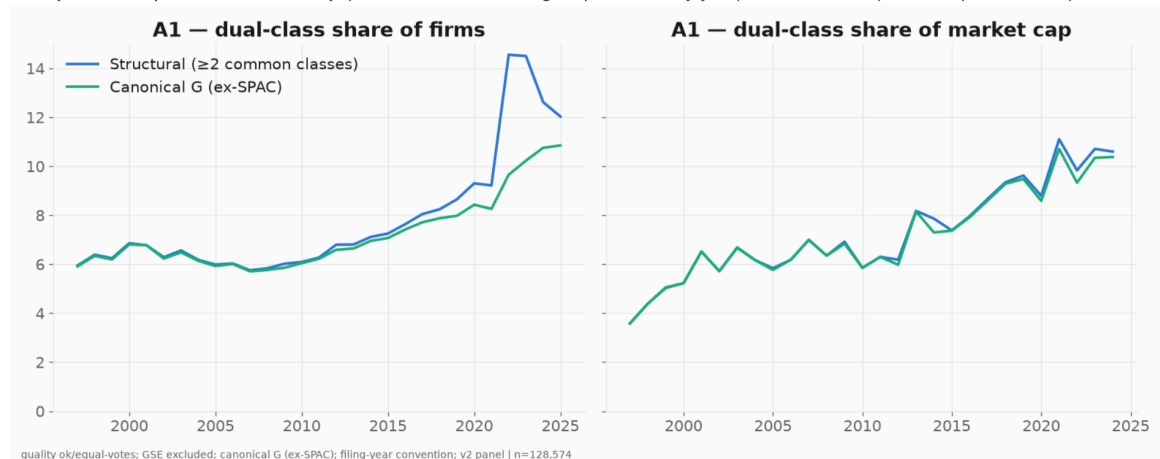
☐	☒ Miscalculations + default-vote rule status:done	4
	#10 · by paulgp was closed 1w ago	
☐	☒ Share-specific board members (class elects X% of board) status:done	3
	#8 · by paulgp was closed 1w ago	
☐	☒ SPAC flag status:done	2
	#7 · by paulgp was closed 1w ago	
☐	☒ Non-voting edge cases: benchmark vs CII dual-class list status:done	5
	#6 · by paulgp was closed 1w ago	
☐	☒ Controlled-company disclosure flag status:done	2
	#4 · by paulgp was closed 1w ago	
☐	☒ Other owners: define inclusion rule (O/Q/S) status:blocked	1
	#5 · by paulgp was closed 1w ago	
☐	☒ Wedge: flag no/discounted cash-flow-rights share classes status:done	4
	#3 · by paulgp was closed 1w ago	
☐	☒ Near-dated options: 3 cases for economic/direct ownership status:done	4
	#2 · by paulgp was closed 1w ago	
☐	☒ Missing outcomes: NA insider ownership status:done	3
	#1 · by paulgp was closed 1w ago	

- Tasks = GitHub **issues**, written like notes to your RA
- Agent picks one up, works, closes it — explanation attached
- **PR** = the verification gate: you approve the diff
- Coauthors read the same threads

► cast: two issues → two reviewed PRs

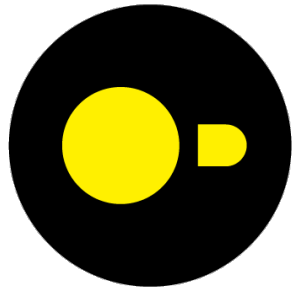
Batch 1 is in (76c4f71): families A and B, per the conventions block — quality ok/equal-votes (quarantined residual: 1,873 partial / 690 broken / 43 suspect / 32 gse, footnoted), canonical G everywhere except A1, latest-filing dedup, EW+VW panels, stats JSON + flags CSV + sanity gates all live. Sample: 128,574 firm-years, 13,731 firms, 1997–2025.

A1 — prevalence, firms vs market cap (canonical $\leq$ structural gate passes every year; VW omits 2025, CRSP caps end 2024):



- Tasks = GitHub **issues**, written like notes to your RA
- Agent picks one up, works, closes it — explanation attached
- **PR** = the verification gate: you approve the diff
- Coauthors read the same threads

► cast: two issues → two reviewed PRs



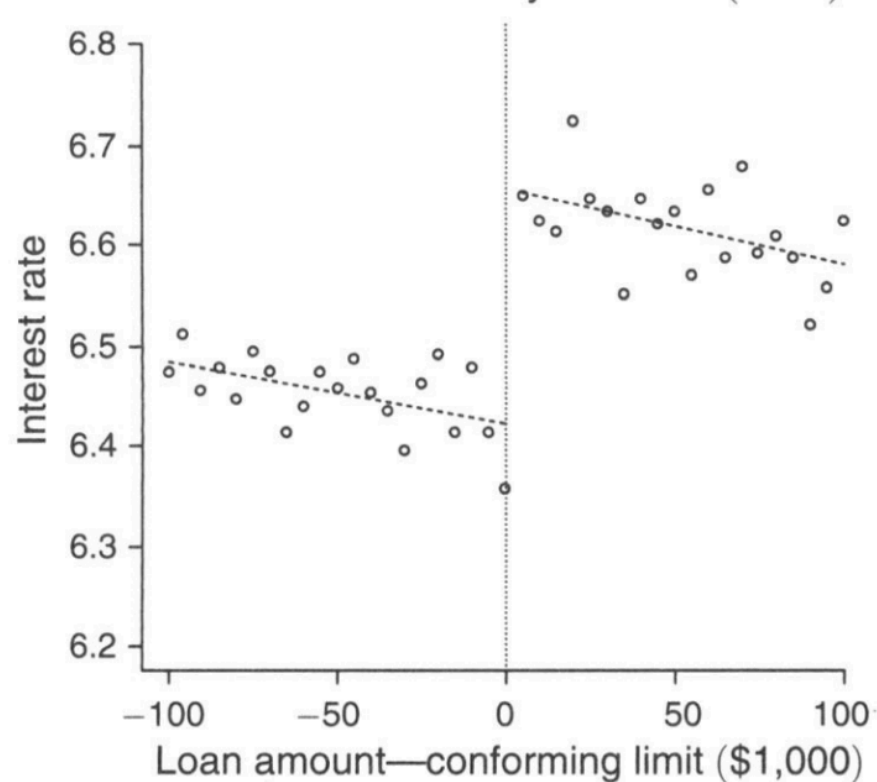
DuckDB

Where possible, structure your data into intermediate data tables

1. Structured databases are fast and agents like working with them
2. Structured db have meta data that is context
3. The startup costs are far lower now!

Makes working on new projects in the future easier too!

Panel A. Mean interest rate by loan size (2006)



Panel B. Loan size density

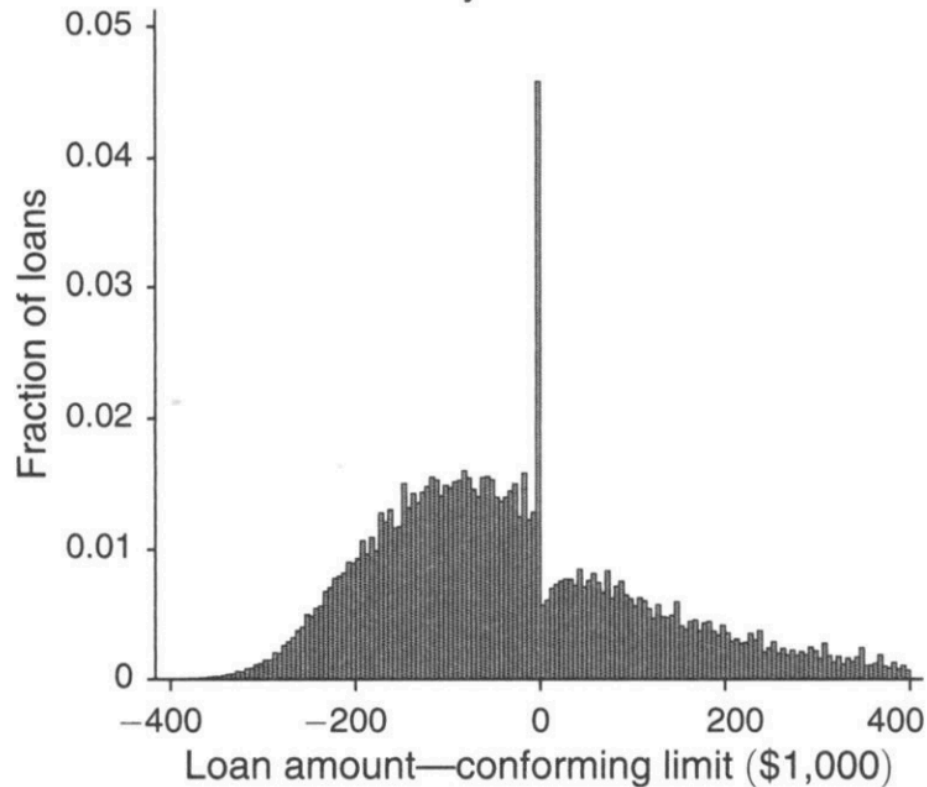
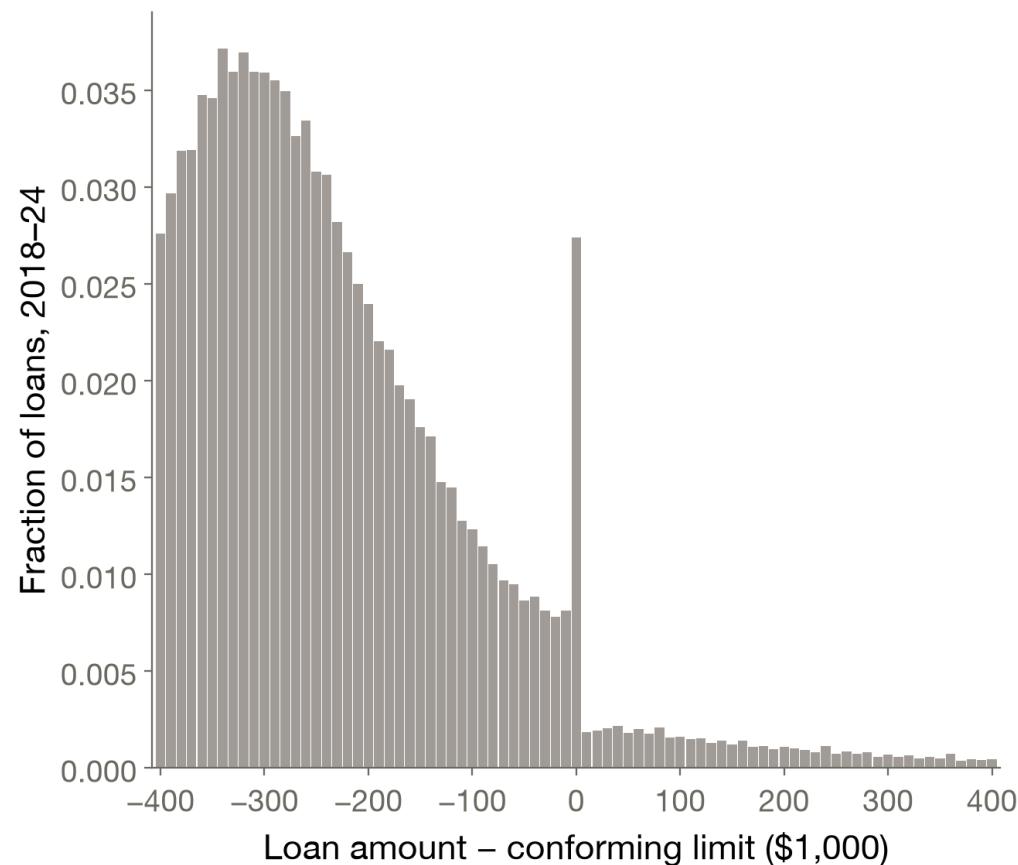
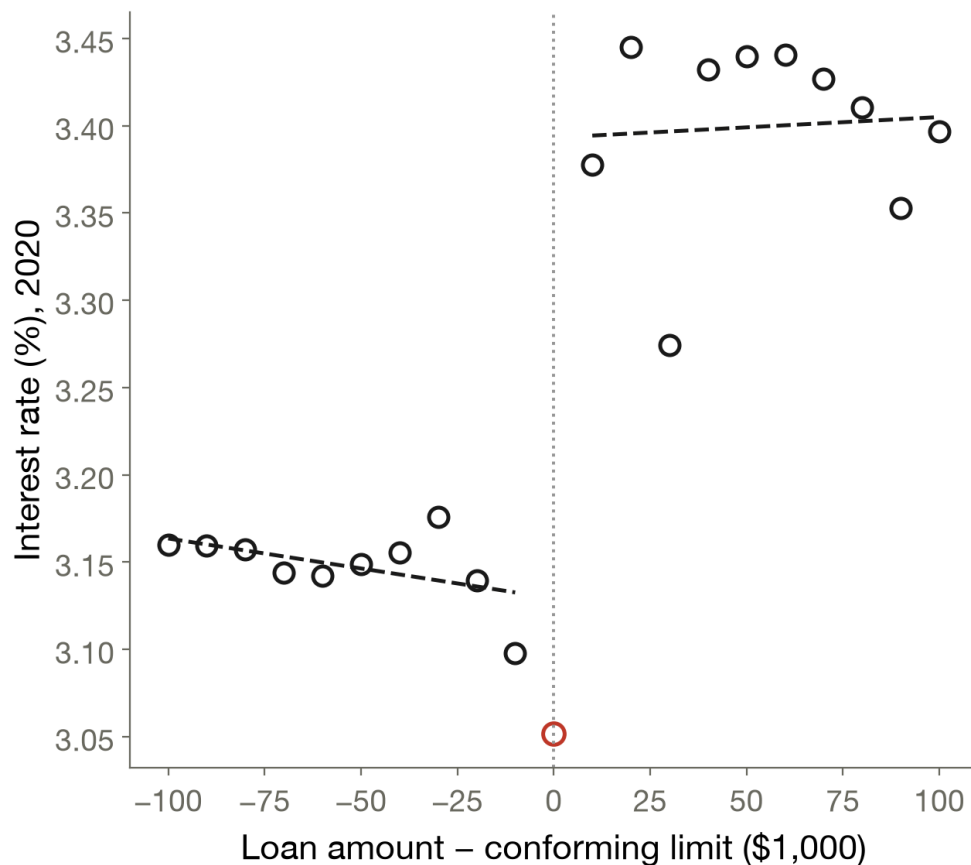
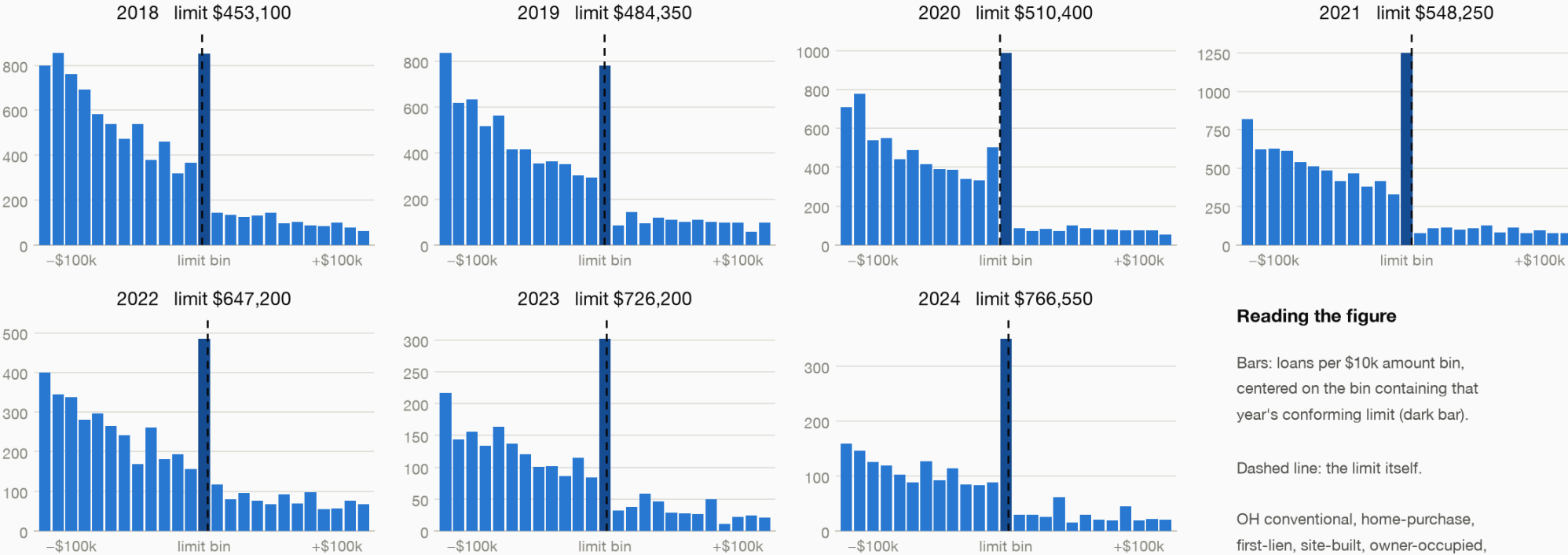


FIGURE 1



The loan-size spike moves with the conforming limit

Ohio, 2018–2024 · loan counts per \$10k bin, ±\$120k around each year's limit · x-axis is relative, so the spike staying at zero means it follows the limit



Reading the figure

Bars: loans per \$10k amount bin, centered on the bin containing that year's conforming limit (dark bar).

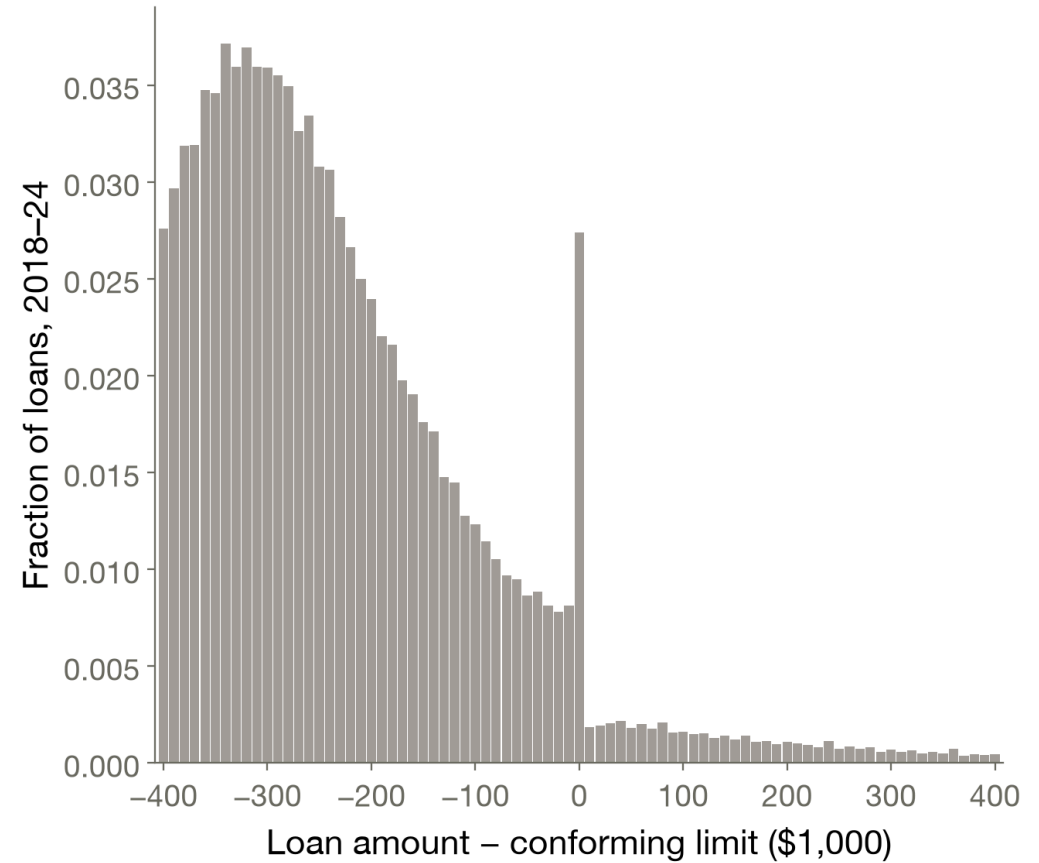
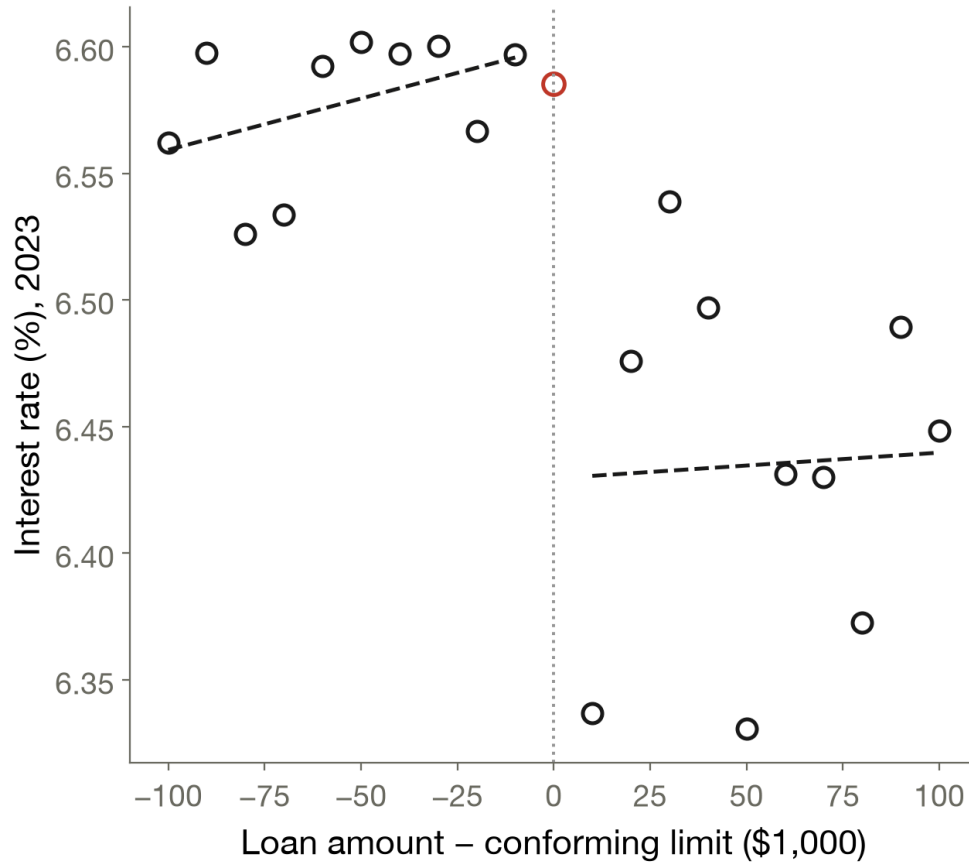
Dashed line: the limit itself.

OH conventional, home-purchase, first-lien, site-built, owner-occupied, 1-unit originations (HMDA).

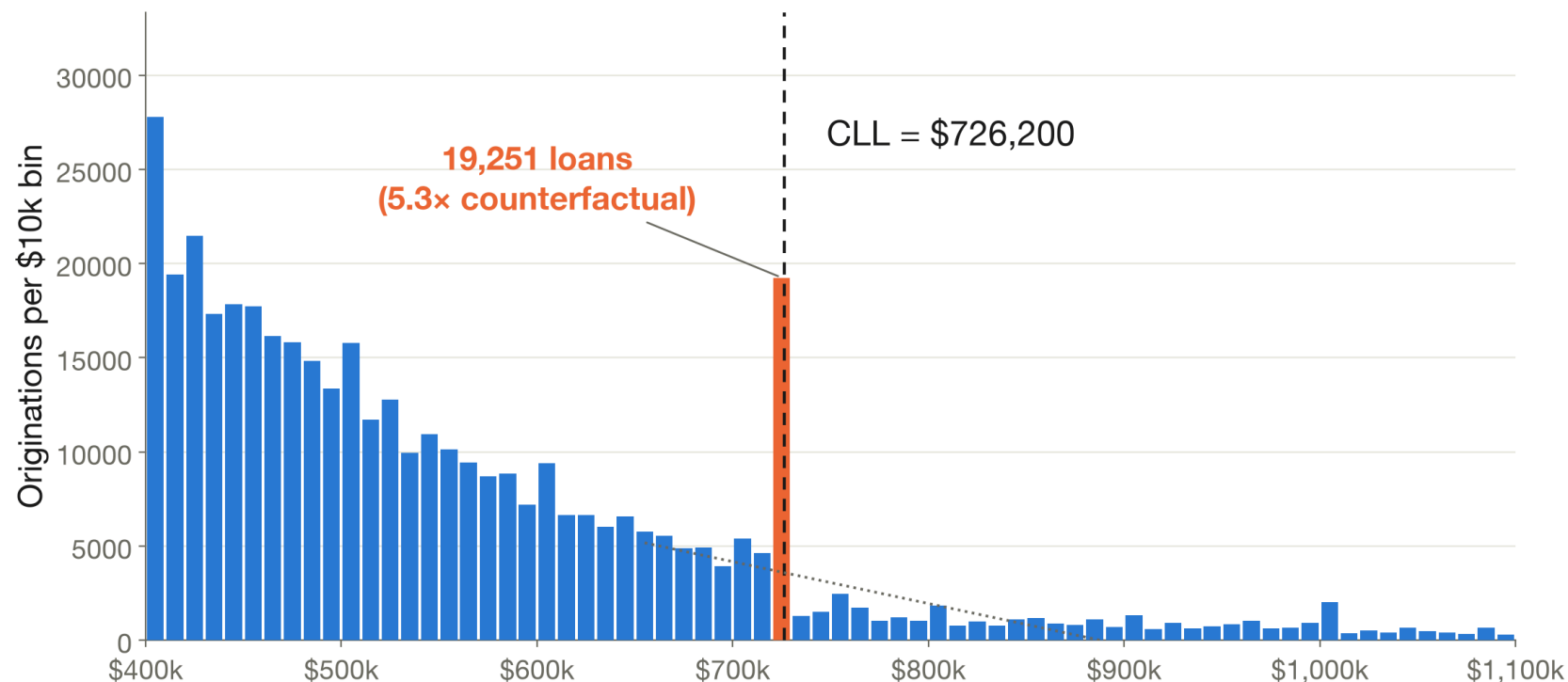
► cast: this figure being made

Now in 2023 — the price change is gone but the quantity spike isn't

52 / 67



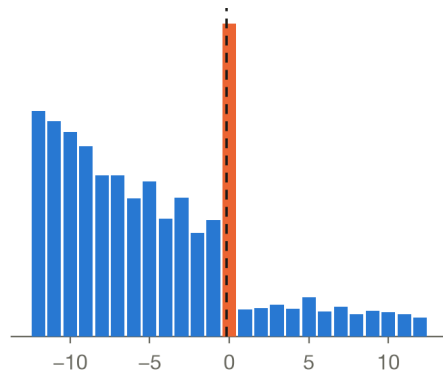
2023, every baseline-limit county: a 5.3× spike at \$726,200



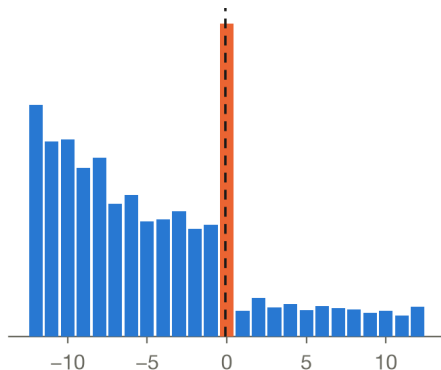
The spike follows the limit, every year

54 / 67

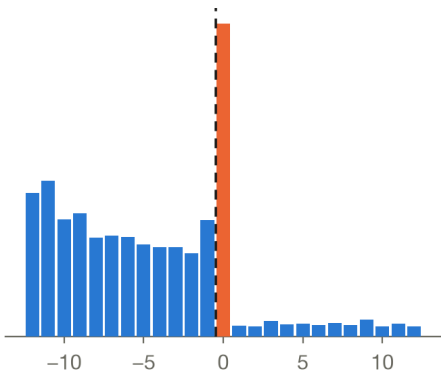
2018 — \$453k · 3.3x



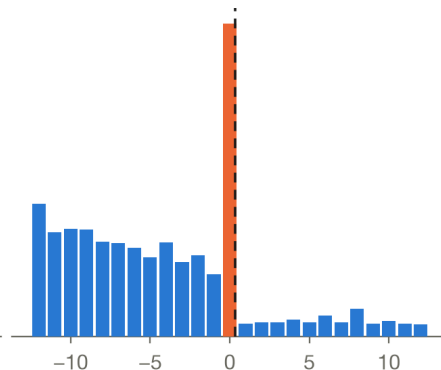
2019 — \$484k · 3.6x



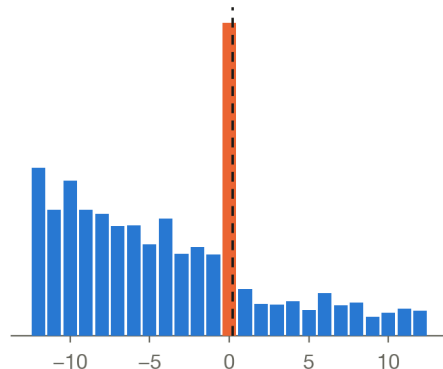
2020 — \$510k · 5.0x



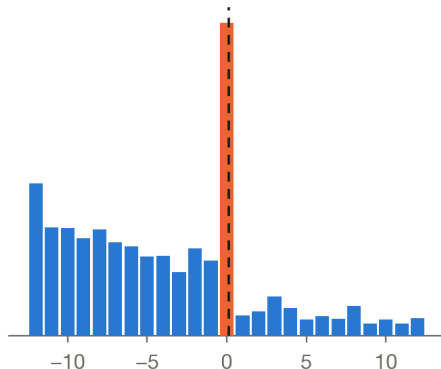
2021 — \$548k · 5.3x



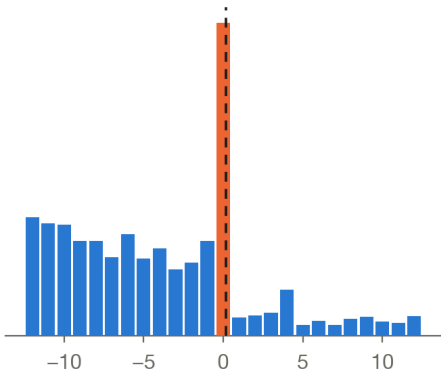
2022 — \$647k · 4.2x



2023 — \$726k · 5.3x



2024 — \$767k · 5.4x



\$10k bins from limit

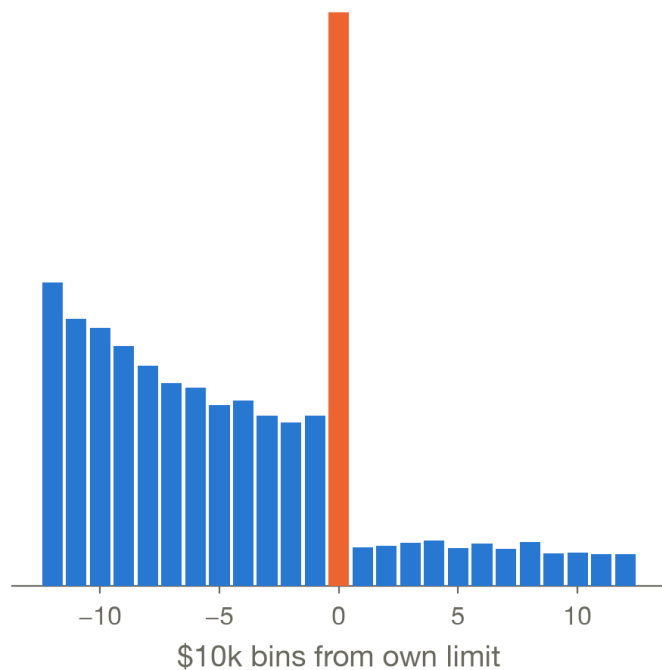
\$10k bins from limit

\$10k bins from limit

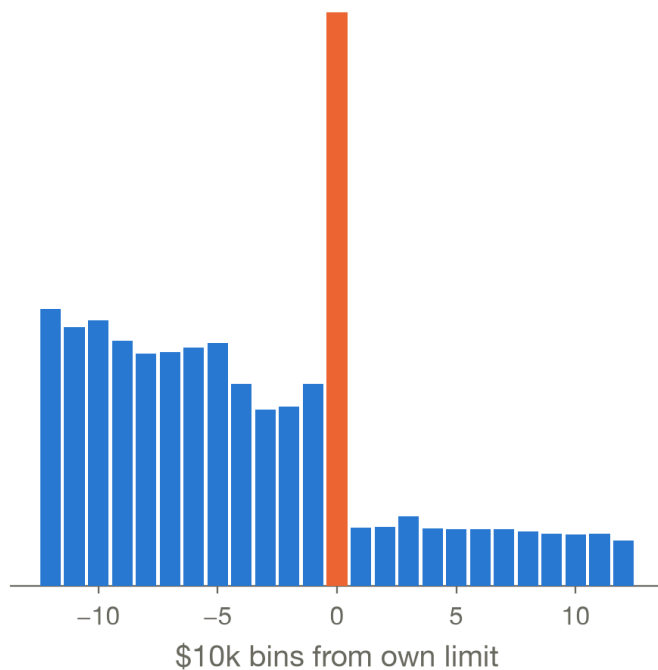
Not a round number: the spike sits at every county's own limit

55 / 67

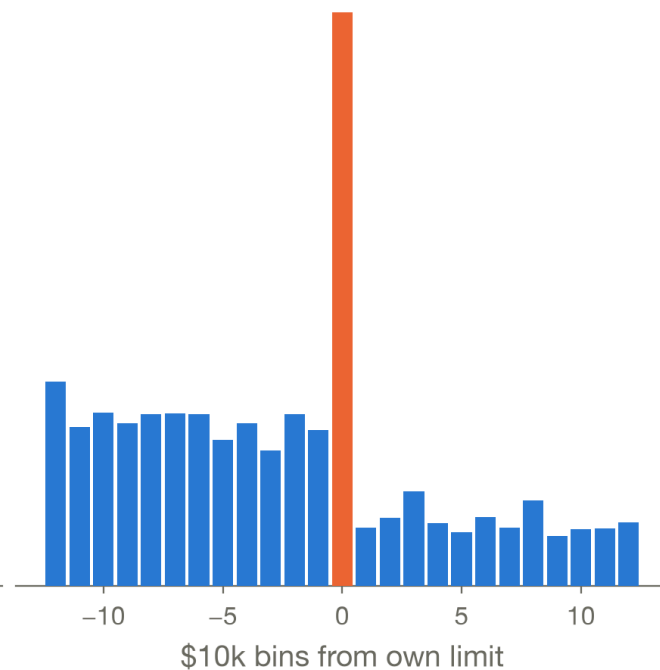
baseline
(one national value/yr)

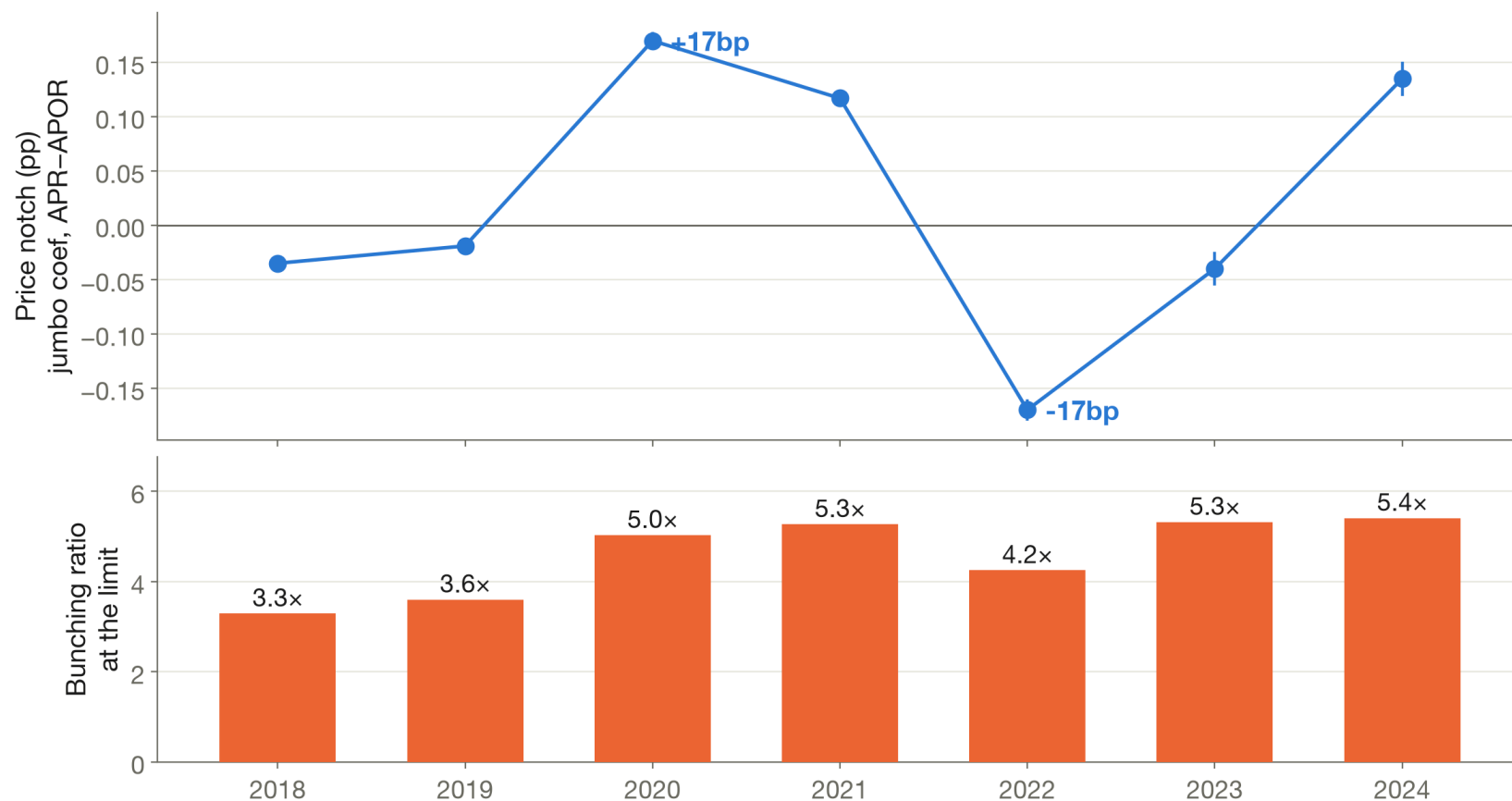


intermediate
(109 distinct limits)



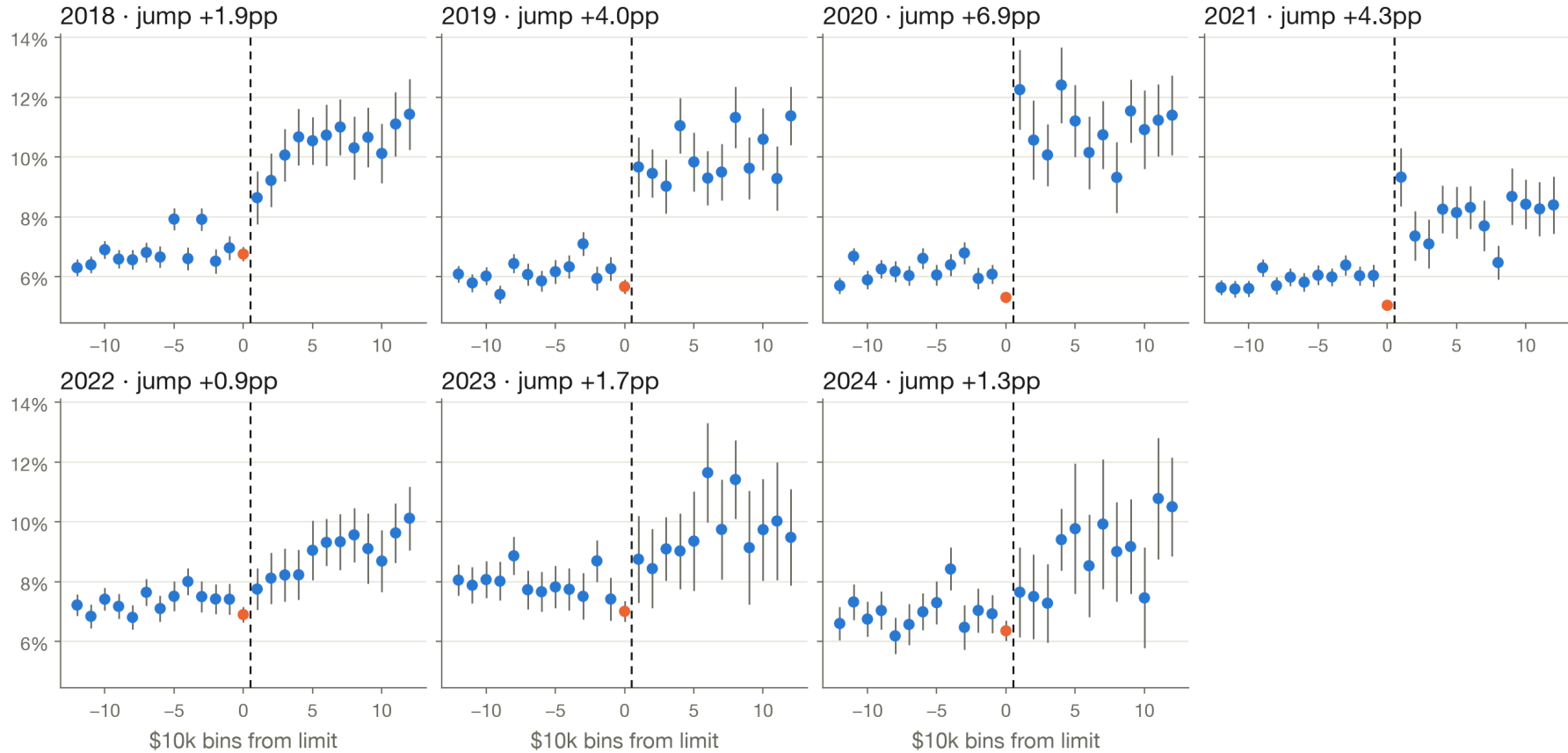
ceiling
(the 150% cap)



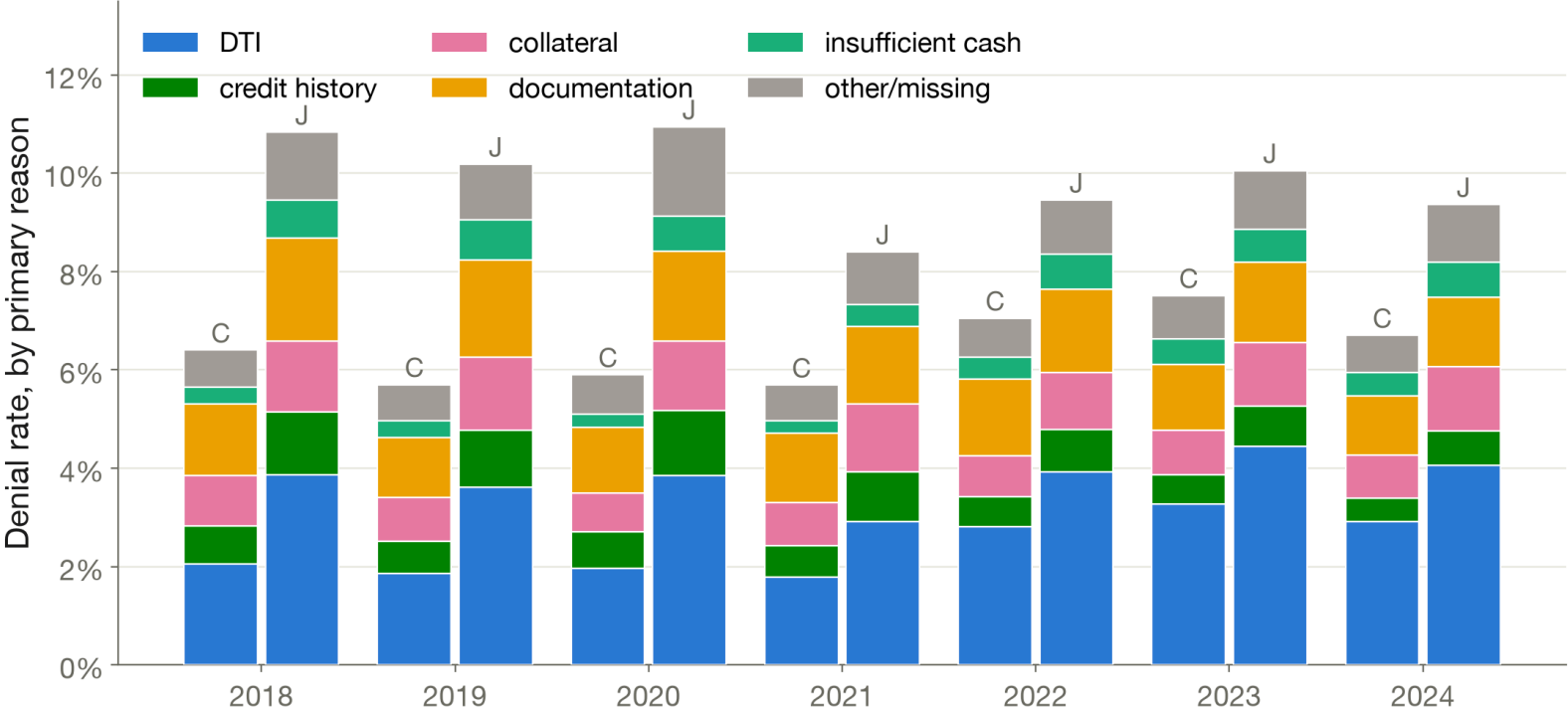


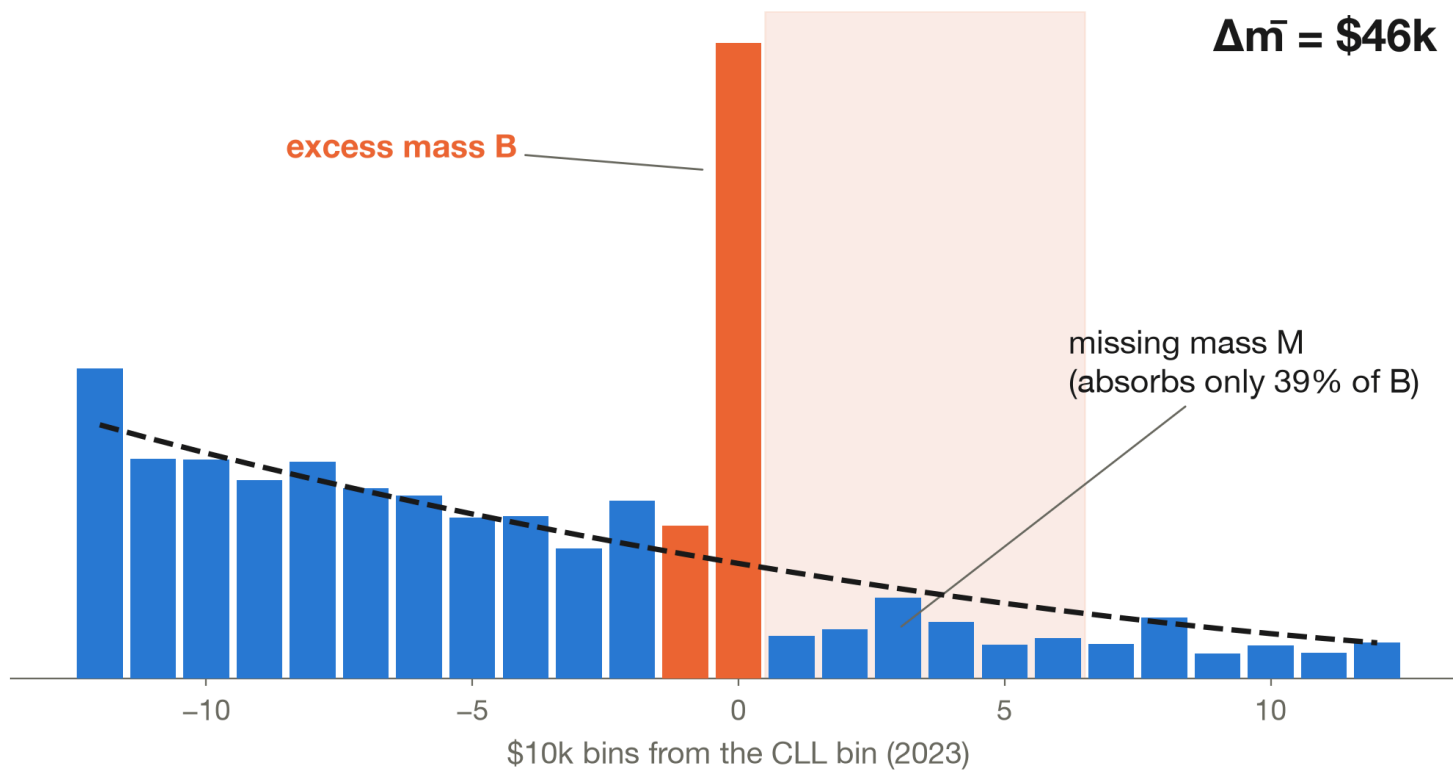
Crossing the limit costs approval odds — most in 2020, least in 2022

57 / 67

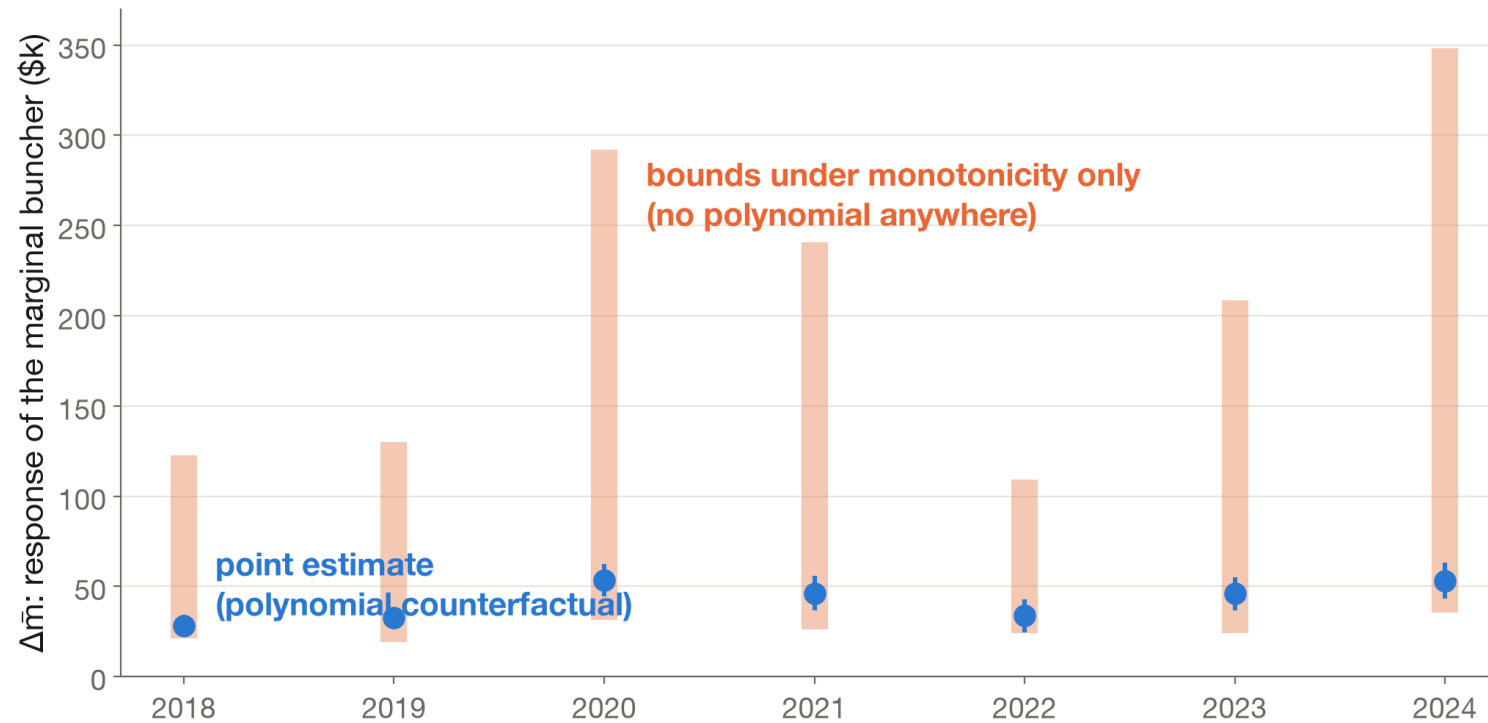


What binds: DTI — the GSE window buys underwriting flexibility



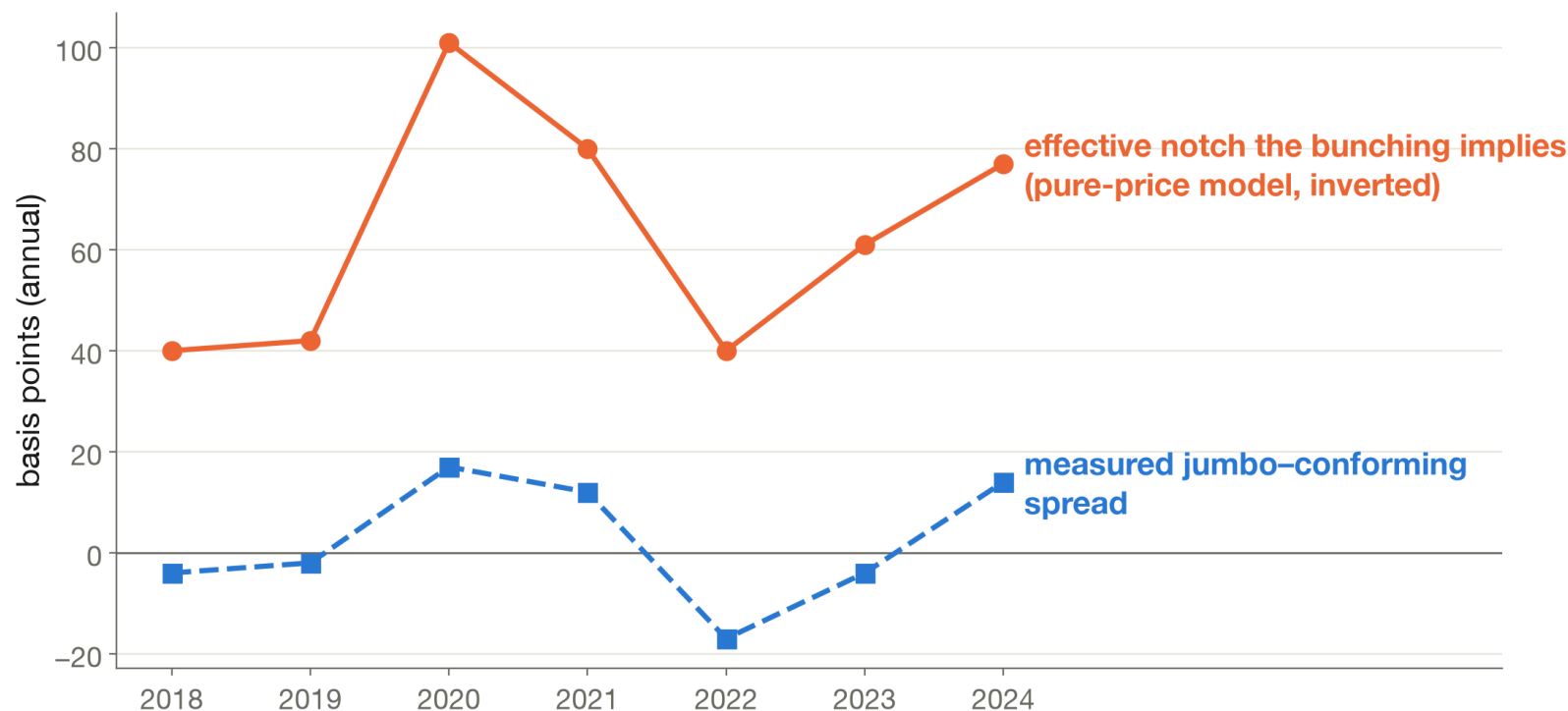


► cast: the estimator built from a spec



What else can we do? Inver the Defusco-Paciorek two-period model

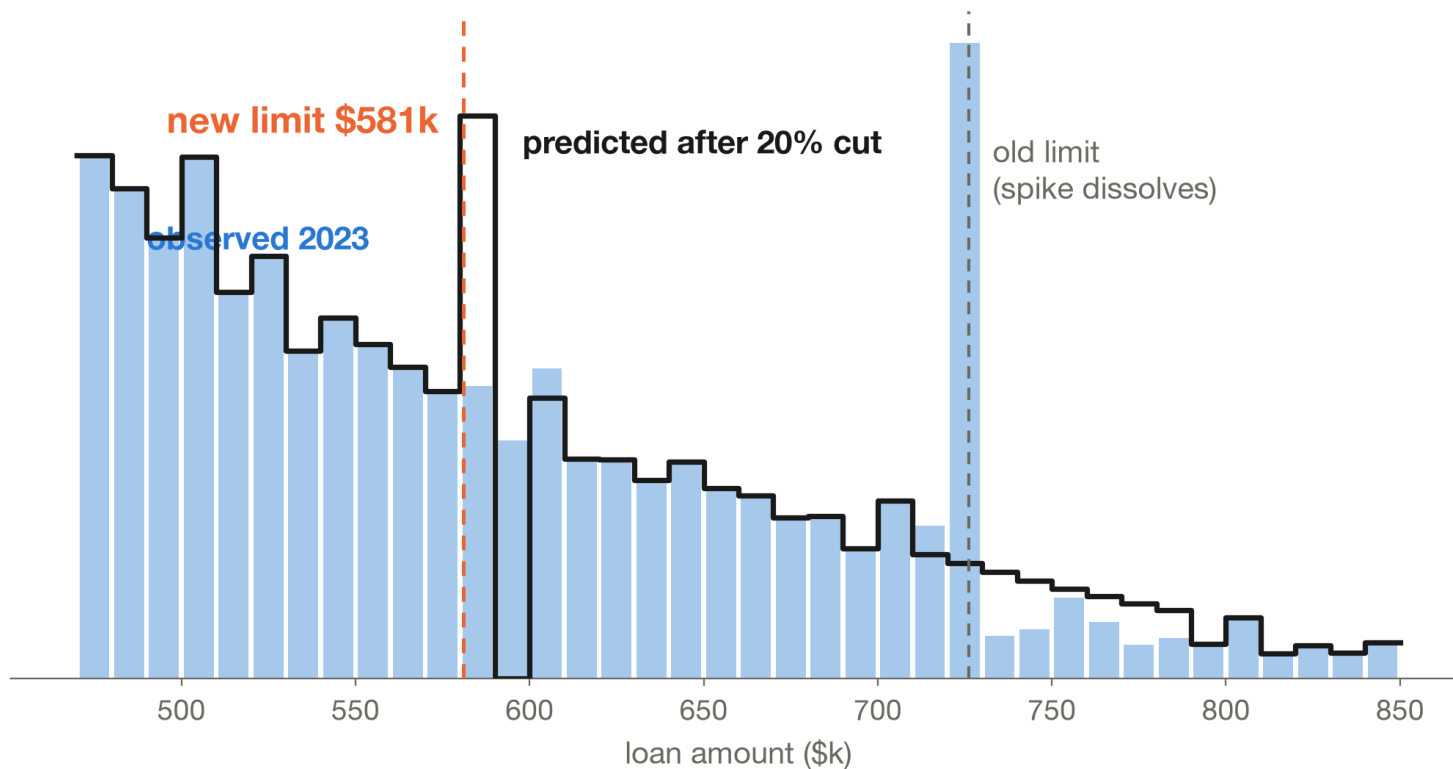
61 / 67



► cast: model counterfactual

Counterfactual — cut the conforming limit 20%

62 / 67

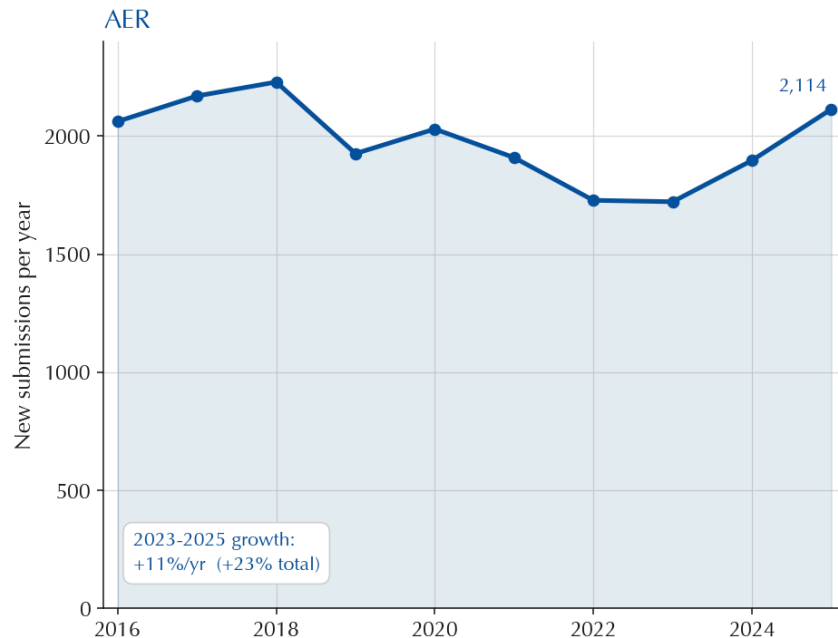


- This was relatively cheap to do
 - Roughly three hours end-to-end
- It was easy because the LLM just followed pre-existing work
 - Thinking of new ideas and questions is much harder
 - LLM is really good at being derivative
- Keeping up with the cutting edge of econometrics is much easier
 - Adding (basic) theory should be much easier!
- Everyone has a very good RA now for much cheaper cost
 - People with RAs now have RAs with RAs!

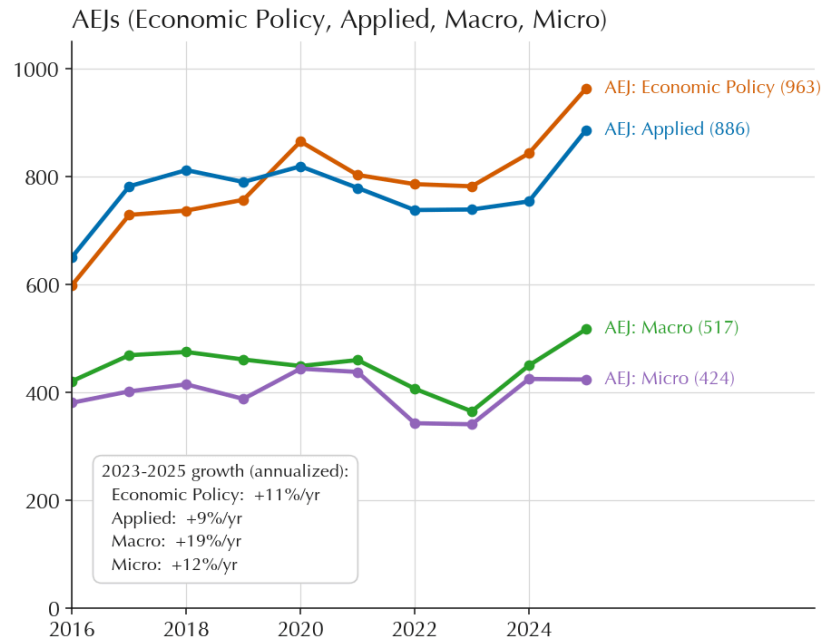
I am of the (empirically grounded) view that research papers will not explode in quantity

AEA journal submissions per year, 2016–2025

New manuscripts submitted (submission year runs Nov 1 – Oct 31). Separate y-scales: AER is ~2–5x the size of each AEJ.



Source: AEA Papers & Proceedings, annual editors' reports (Table 2). AEA years complete through 2025. Graph: P. Goldsmith-Pinkham





D. Yanagizawa-Drott ✓
@YanagizawaD



A new project.

RQ: Can we automate policy evaluation?

Not today, obviously. But maybe soon. To **reliably, cheaply and quickly** figure out what policies work and don't work, seems potentially super valuable to society.

ape.socialcatalystlab.org

Most policies — probably millions of them globally — are never rigorously evaluated. Data is plenty but there aren't enough researchers. Could AI help? We genuinely don't know. So we're running an experiment. An AI system attempts to produce economics research at scale, end to end, using publicly available data. Will any be good? How would we even know? Ideally, we'd want PhDs or editors of top journals to evaluate all of them. But they are busy. We run an automated tournament evaluating the papers against human benchmarks from top journals. This could help triage. Get to a "you know it when you see it" moment, faster. Most importantly, everything is public: papers, code, data, failures. The more people look, the faster mistakes get caught. And we want feedback! In fact, the core thesis is that self-improvement loops can only happen through feedback mechanisms. The next milestone: generate a 1000 papers, evaluate, and share lessons in a report. Can policy evaluation be automated? Or is hallucinated slop unavoidable? Let's find out!

11:33 AM · Feb 11, 2026 · **73K** Views



5



79



225

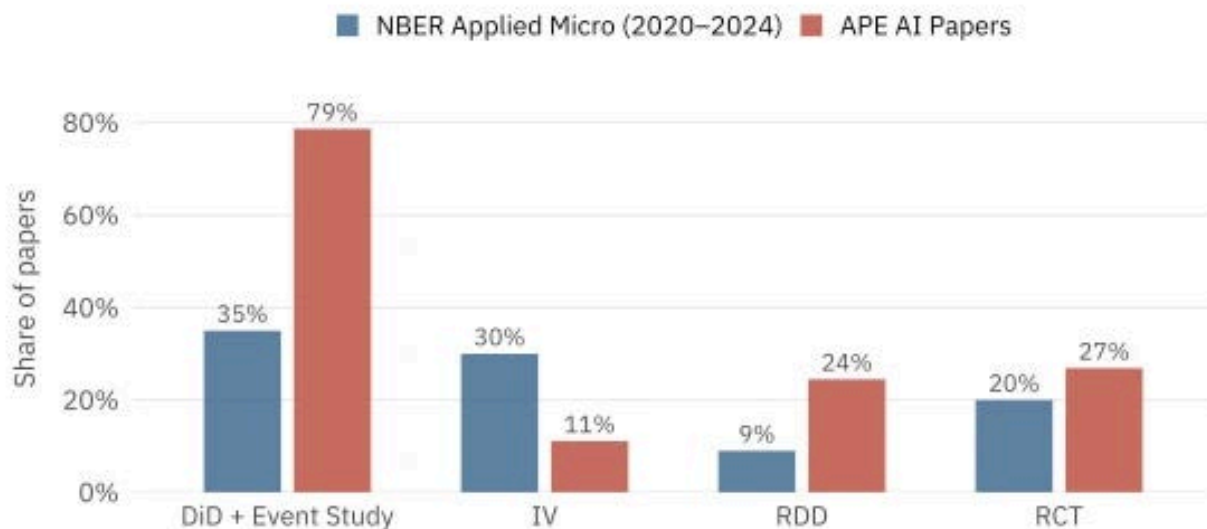


199



AI Papers Lean Heavily on DiD and RDD, Rarely Use Instrumental Variables

Share of papers mentioning each method (any mention, CKZ keyword matching)



Sources: NBER Applied Micro working papers from Goldsmith-Pinkham (2026), approximate shares circa 2020-2024. APE AI papers (n = 592) from ape.socialcatalystlab.org. Categories are not mutually exclusive.

<https://paulgp.substack.com/p/ai-one-shot-papers>



paulgp.com/nber_hf_2026