

Thinking With Agents

Webinar 1: AI for Research

Emily Beam & Erkmen G. Aslim

Department of Economics, University of Vermont

June 2026 · Live Online

thinkingwithagents.github.io

This Hour



1. A Few Ideas

The inverse laws of AI, and why **verification** is the bottleneck in agentic research.

~10 min, slides

This Hour



1. A Few Ideas

The inverse laws of AI, and why **verification** is the bottleneck in agentic research.

~10 min, slides



2. Website Builder

Your CV → a live academic website you can actually maintain.
Skill + live demo.

~20 min, mostly live

This Hour



1. A Few Ideas

The inverse laws of AI, and why **verification** is the bottleneck in agentic research.

~10 min, slides



2. Website Builder

Your CV → a live academic website you can actually maintain. Skill + live demo.

~20 min, mostly live



3. Brainstorm

Stress-test a big research question end-to-end, live — and keep the brief it produces.

~25 min, live

This Hour



1. A Few Ideas

The inverse laws of AI, and why **verification** is the bottleneck in agentic research.

~10 min, slides



2. Website Builder

Your CV → a live academic website you can actually maintain. Skill + live demo.

~20 min, mostly live



3. Brainstorm

Stress-test a big research question end-to-end, live — and keep the brief it produces.

~25 min, live

Slides are deliberately light — most of this hour happens **live in the terminal**. Prompts and links are on the slides so you can copy as we go.

Part 2 of today's webinar — picking up from Emily's hour. No agentic-AI basics here; we go straight in.

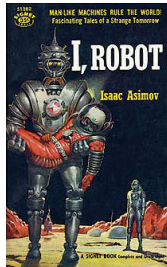
The Three Inverse Laws of AI

Asimov's laws constrain **robots** to keep humans safe — the *inverse* laws constrain **us**.

I. Non-Anthropomorphism

Don't attribute judgement, intent, or care to the agent.

Fluent output $\neq$ understanding — a statistical model, not a coauthor.



Asimov wrote laws for the robots.

These are for us.

The Three Inverse Laws of AI

Asimov's laws constrain **robots** to keep humans safe — the *inverse* laws constrain **us**.

I. Non-Anthropomorphism

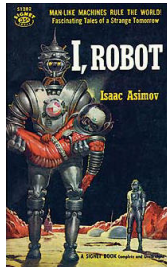
Don't attribute judgement, intent, or care to the agent.

Fluent output $\neq$ understanding — a statistical model, not a coauthor.

II. Non-Deference

Never treat AI output as authoritative without **independent verification**.

The more serious the consequences, the higher the burden of verification.



Asimov wrote laws for the robots.

These are for us.

The Three Inverse Laws of AI

Asimov's laws constrain **robots** to keep humans safe — the *inverse* laws constrain **us**.

I. Non-Anthropomorphism

Don't attribute judgement, intent, or care to the agent.

Fluent output $\neq$ understanding — a statistical model, not a coauthor.

II. Non-Deference

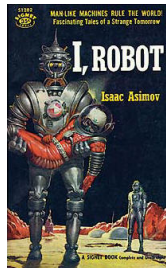
Never treat AI output as authoritative without **independent verification**.

The more serious the consequences, the higher the burden of verification.

III. Non-Abdication

You remain fully responsible and accountable for the consequences.

"The AI told me so" is never a defense.



Asimov wrote laws for the robots.

These are for us.

The Three Inverse Laws of AI

Asimov's laws constrain **robots** to keep humans safe — the *inverse* laws constrain **us**.

I. Non-Anthropomorphism

Don't attribute judgement, intent, or care to the agent.

Fluent output $\neq$ understanding — a statistical model, not a coauthor.

II. Non-Deference

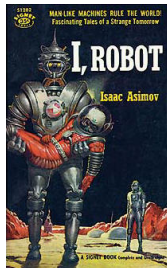
Never treat AI output as authoritative without **independent verification**.

The more serious the consequences, the higher the burden of verification.

III. Non-Abdication

You remain fully responsible and accountable for the consequences.

"The AI told me so" is never a defense.



Asimov wrote laws for the robots.

These are for us.

For researchers, Law III is sharp: **your** name
— not the agent's — is on the paper.

Susam Pal, "Three Inverse Laws of AI" (Jan 2026), susam.net

The New Bottleneck Is Verification



What agents made cheap

- ▶ Code, scrapers, cleaned datasets
- ▶ Figures, tables, drafts
- ▶ Literature summaries

Produced in **minutes**, at scale.



What stayed expensive

Knowing it's **right**:

- ▶ the correct sample & merge
- ▶ the correct specification
- ▶ the correct interpretation

The New Bottleneck Is Verification



What agents made cheap

- ▶ Code, scrapers, cleaned datasets
- ▶ Figures, tables, drafts
- ▶ Literature summaries

Produced in **minutes**, at scale.



What stayed expensive

Knowing it's **right**:

- ▶ the correct sample & merge
- ▶ the correct specification
- ▶ the correct interpretation

“The more serious the potential consequences, the higher the burden of verification should be.” — Susam Pal

Research errors are subtle, plausible-looking, and costly.

A pipeline that runs is not a pipeline that's right.

Borrow the Software Verification Mindset

In software the loop closes fast: it compiles, tests pass, the app visibly works.

A research pipeline “works” *whether or not it's right* — so we import the discipline:

Software practice	Research analog
Unit tests	Assertion checks on data: row counts, unique IDs, value ranges, merge rates
Code review	A fresh agent adversarially reviews code & output — writer $\neq$ reviewer
Continuous integration	One command re-runs raw data $\rightarrow$ final tables; same numbers every run
Small commits & diffs	Incremental changes you can read, test, and revert — git as a checkpoint
Logging	An audit trail: what was read, dropped, imputed, assumed

Borrow the Software Verification Mindset

In software the loop closes fast: it compiles, tests pass, the app visibly works.

A research pipeline “works” *whether or not it's right* — so we import the discipline:

Software practice	Research analog
Unit tests	Assertion checks on data: row counts, unique IDs, value ranges, merge rates
Code review	A fresh agent adversarially reviews code & output — writer $\neq$ reviewer
Continuous integration	One command re-runs raw data $\rightarrow$ final tables; same numbers every run
Small commits & diffs	Incremental changes you can read, test, and revert — git as a checkpoint
Logging	An audit trail: what was read, dropped, imputed, assumed

Treat agent output like RA output: useful, fast, and **audited before it touches the paper**.

No Full Discretion — and Build Audit Tools



Keep your hands on the wheel

- ▶ **Plan first** — review the plan before any code runs
- ▶ **Small steps** — verify at checkpoints, then continue
- ▶ **Fresh eyes** — never let the author-agent grade its own work
- ▶ No YOLO mode on research pipelines



Diagnostic & audit skills

The skills we keep building are **auditors**:

- ▶ Referee-style paper review
- ▶ Adversarial empirical audits
- ▶ Data validators & replication checks

Cheap to run → run them **often**.

No Full Discretion — and Build Audit Tools



Keep your hands on the wheel

- ▶ **Plan first** — review the plan before any code runs
- ▶ **Small steps** — verify at checkpoints, then continue
- ▶ **Fresh eyes** — never let the author-agent grade its own work
- ▶ No YOLO mode on research pipelines



Diagnostic & audit skills

The skills we keep building are **auditors**:

- ▶ Referee-style paper review
- ▶ Adversarial empirical audits
- ▶ Data validators & replication checks

Cheap to run → run them **often**.

When production gets cheap, comparative advantage shifts to **verification**.

The most valuable tools you'll build are the ones that **check** the work.

Demo 1

Academic Website Builder — Your CV → a Site You Can Maintain

Academic Website Builder — What It Is



CV in → website out

- ▶ Hand it your CV (PDF/DOCX) — it extracts publications, teaching, bio, links
- ▶ **Three distinct designs**, not recolors:
minimal-dark · light-academic · bold-modern
- ▶ You pick by *seeing* them with your own content
- ▶ Plain static files → free hosting (GitHub Pages)

Academic Website Builder — What It Is



CV in → website out

- ▶ Hand it your CV (PDF/DOCX) — it extracts publications, teaching, bio, links
- ▶ **Three distinct designs**, not recolors: minimal-dark · light-academic · bold-modern
- ▶ You pick by *seeing* them with your own content
- ▶ Plain static files → free hosting (GitHub Pages)



Built to be maintained

- ▶ One source of truth: `cv-data.json`
- ▶ One command: `python build.py` (~1 s, no dependencies)
- ▶ Update = **edit one file, run one command, push**

No web tooling, no frameworks, no servers. The academic owns the folder.

Academic Website Builder — What It Is



CV in → website out

- ▶ Hand it your CV (PDF/DOCX) — it extracts publications, teaching, bio, links
- ▶ **Three distinct designs**, not recolors: minimal-dark · light-academic · bold-modern
- ▶ You pick by *seeing* them with your own content
- ▶ Plain static files → free hosting (GitHub Pages)



Built to be maintained

- ▶ One source of truth: `cv-data.json`
- ▶ One command: `python build.py` (~1 s, no dependencies)
- ▶ Update = **edit one file, run one command, push**

No web tooling, no frameworks, no servers. The academic owns the folder.



Live demo site (fictitious economist “Adam Smith”):

`erkmenaslim.github.io/academic-website-demo`

— the skill download lives there too

No Skill? One Prompt Gets You Close

```
> claude
```

```
I'm an economist. My CV is in this folder (cv.pdf). Build me a clean, modern academic website from it:
```

1. Extract everything: bio, publications, working papers, teaching, grants, contact details, profile links.
2. Create three distinct design mock-ups (minimal dark, light academic, bold modern) as self-contained HTML files.
3. Keep all content in ONE data file (cv-data.json) plus a small build script (build.py), so I can update the site later by editing one file and re-running one command.
4. No frameworks, no servers -- plain static files I can host for free on GitHub Pages.

```
Save everything in this folder and screenshot each design.
```

This works today, with no setup — but every run comes out a little different.

The **skill** is the streamlined version: same intake, same file layout, same update path, **every time**.

With the Skill: Streamlined & Repeatable



Get it & trigger it

Install: download from the demo site (one click), or clone the repo → drop into `.claude/skills/`

Trigger with plain language:

"Turn my CV into an academic website."

Markdown + scripts — the same skill also works copied into Codex.



What happens

1. Short intake: links, email, bio length, headshot, where you'll deploy
2. Builds **all three** themes from one `cv-data.json`
3. Side-by-side preview → you pick
4. Deploys to GitHub Pages in ~1 min

With the Skill: Streamlined & Repeatable



Get it & trigger it

Install: download from the demo site (one click), or clone the repo → drop into `.claude/skills/`

Trigger with plain language:

"Turn my CV into an academic website."

Markdown + scripts — the same skill also works copied into Codex.



What happens

1. Short intake: links, email, bio length, headshot, where you'll deploy
2. Builds **all three** themes from one `cv-data.json`
3. Side-by-side preview → you pick
4. Deploys to GitHub Pages in ~1 min



Live: walk the Adam Smith site → add a new paper to `cv-data.json` → `python build.py` → push → watch the live site update in ~1 minute.

The headline: going from CV v1 → v2 — new papers, a photo, a redeploy — takes *minutes*.

Demo 2

/research-brainstorm — Stress-Test an Idea, Live

/research-brainstorm in One Slide



A senior colleague, not a cheerleader

- ▶ Restates your question in structured terms — unit, population, treatment, outcome, setting
- ▶ Probes *before* searching: so-what test, ideal experiment
- ▶ **Parallel lit scan** — published, working papers, in-progress (scooping risk)
- ▶ Adversarial critique — “the referee report in advance”
- ▶ 2–3 **pivots** + feasibility (delegates to /find-data)

/research-brainstorm in One Slide



A senior colleague, not a cheerleader

- ▶ Restates your question in structured terms — unit, population, treatment, outcome, setting
- ▶ Probes *before* searching: so-what test, ideal experiment
- ▶ **Parallel lit scan** — published, working papers, in-progress (scooping risk)
- ▶ Adversarial critique — “the referee report in advance”
- ▶ 2–3 **pivots** + feasibility (delegates to /find-data)

The artifact — a research brief saved to disk:

`research_briefs/YYYY-MM-DD-{slug}.md`

Question · closest work + deltas · strongest objection · pivots · feasibility · next concrete step

/research-brainstorm in One Slide



A senior colleague, not a cheerleader

- ▶ Restates your question in structured terms — unit, population, treatment, outcome, setting
- ▶ Probes *before* searching: so-what test, ideal experiment
- ▶ **Parallel lit scan** — published, working papers, in-progress (scooping risk)
- ▶ Adversarial critique — “the referee report in advance”
- ▶ 2–3 **pivots** + feasibility (delegates to /find-data)

The artifact — a research brief saved to disk:

`research_briefs/YYYY-MM-DD-{slug}.md`

Question · closest work + deltas · strongest objection · pivots · feasibility · next concrete step

The conversation is great.

The artifact is what you keep.

/research-brainstorm in One Slide



A senior colleague, not a cheerleader

- ▶ Restates your question in structured terms — unit, population, treatment, outcome, setting
- ▶ Probes *before* searching: so-what test, ideal experiment
- ▶ **Parallel lit scan** — published, working papers, in-progress (scooping risk)
- ▶ Adversarial critique — “the referee report in advance”
- ▶ 2–3 **pivots** + feasibility (delegates to /find-data)

The artifact — a research brief saved to disk:

`research_briefs/YYYY-MM-DD-{slug}.md`

Question · closest work + deltas · strongest objection · pivots · feasibility · next concrete step

*The conversation is great.
The artifact is what you keep.*

Install:

```
npx skills add  
  thinkingwithagents/skills  
  --skill research-brainstorm
```

Live Demo: One Big Question

Does agentic AI level the playing field in research —
or widen the gap?

junior vs. senior · coding-fluent vs. not · top-ranked vs. the rest

Half-baked on purpose. That's the point — the skill's first job is to sharpen it into something searchable and falsifiable.

Watch for: the structured restatement · the lit scan verdict · the strongest objection · the pivots · the brief landing on disk.

Have a half-formed question of your own? Drop it in the chat — if time allows, we'll run yours instead.

Takeaways

1

Production is cheap now. **Verification is the bottleneck** — budget your attention there.

Takeaways

1

Production is cheap now. **Verification is the bottleneck** — budget your attention there.

2

Never defer blindly: **incremental steps, checkpoints, fresh-agent review**. You own the output.

Takeaways

1

Production is cheap now. **Verification is the bottleneck** — budget your attention there.

2

Never defer blindly: **incremental steps, checkpoints, fresh-agent review**. You own the output.

3

Borrow the software mindset: **tests, review, reproducible re-runs** — build audit tools.

Takeaways

1

Production is cheap now. **Verification is the bottleneck** — budget your attention there.

2

Never defer blindly: **incremental steps, checkpoints, fresh-agent review**. You own the output.

3

Borrow the software mindset: **tests, review, reproducible re-runs** — build audit tools.

4

Skills turn good workflows into shareable, repeatable, auditable artifacts.

Takeaways

1

Production is cheap now. **Verification is the bottleneck** — budget your attention there.

2

Never defer blindly: **incremental steps, checkpoints, fresh-agent review**. You own the output.

3

Borrow the software mindset: **tests, review, reproducible re-runs** — build audit tools.

4

Skills turn good workflows into shareable, repeatable, auditable artifacts.

Webinar 2: AI for Teaching — same place, coming soon.



Thank You!

Webinar 2: AI for Teaching — see you there

Emily Beam & Erkmen G. Aslim

Department of Economics, University of Vermont

Slides, skills & demo links:

thinkingwithagents.github.io