Agentic AI in Social Science Research

From Annotators to Autonomous Pipelines

Matt DiGiuseppe

Institute of Political Science
Leiden University

University of Glasgow
May 2026

Roadmap

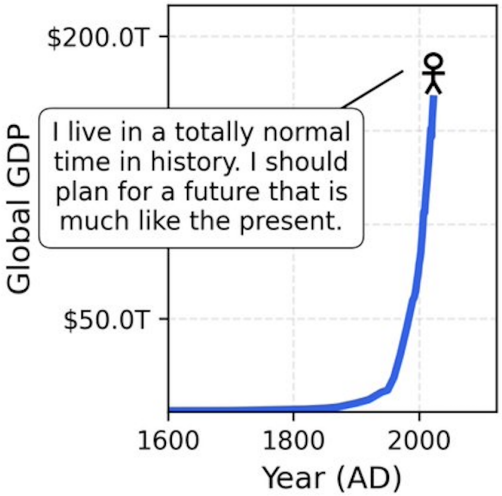
1. Why Agents, Why Now
2. What Is an Agent?
3. Case Study: Economic Exposure and Congressional Rhetoric
4. What the Pipeline Taught Me
5. Agentic Qualitative Interviewing
6. From Synthetic Individuals to Simulated Societies
7. Building Your Own Pipeline

From LLM-as-Tool to LLM-as-Colleague

Last year's lecture: LLMs *annotate*, *classify*, *scale* text.

This year's lecture: LLMs *plan*, *search*, *call code*, *call other LLMs*, and *deliver finished pipelines*.

- The unit of work is no longer "one prompt, one answer."
- It is a multi-step research workflow with checkpoints.
- Researchers act as **managers**, not typists.



[h/t Tim Urban, Wait But Why]

What Can Agentic AI Do for Your Research?

- Build datasets from the open web or existing datasets/corpa at scale
- Run multi-stage pipelines unattended
- Code, debug, and document its own scripts
- Conduct adaptive interviews and surveys
- Triangulate evidence across many models
- Simulate respondents to pre-test designs
- Populate small-scale societies for ABM-style work
- Produce a full audit trail by default

The frontier is not what an agent can do — it is what we can *trust* an agent to do.

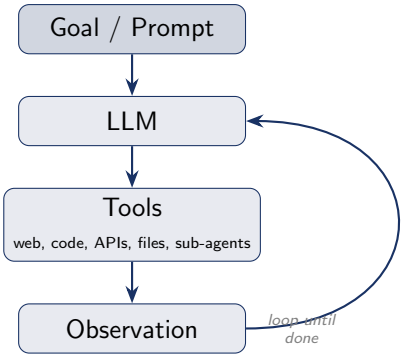
The Minimal Definition

An **agent** is an LLM that:

1. Receives a goal
2. Selects and invokes **tools** (web search, code execution, file I/O, APIs, other LLMs)
3. Observes the result
4. Iterates until done (or hits a stop condition)

What changes vs. a chatbot?

- Persistent state across many turns
- Tool use $\Rightarrow$ the model can *act*, not just *speak*
- Self-correction inside one task



What's the Right Counterfactual?

The methods literature usually picks the wrong comparison.

Wrong counterfactual #1: deterministic text-as-data

Dictionary methods, supervised classifiers, BERT — reproducible, auditable, transparent.

Against this benchmark the LLM looks like a black box.

Most coding tasks don't benefit from these tools.

Wrong counterfactual #2: an idealised RA team

Trained coders, validated protocol, second coder, κ reported.

Against this benchmark the LLM looks underregulated.

But this is not what most projects actually do.

And in 2026: can you swear your RAs aren't quietly running everything through ChatGPT themselves? *Hidden cyborgs everywhere.* The idealised RA team is a relic of a pre-2023 world.

The Honest Counterfactual: One RA, Also a Black Box

The actual alternative is one RA reading and coding texts.

That is a black box too.

- Coding rules drift across the day.
- Fatigue effects are unmeasured.
- No second coder, no κ reported.
- Decisions are not logged.
- Replication is impossible — the RA graduated.

We trusted that black box for decades.

The honest question is not
"Is the LLM a black box?"

It is
"*Which* black box,
and which one can we audit?"

The Access Argument

Even the RA comparison overstates who has the resource.

Who has no RAs?

- Teaching-focused institutions
- Researchers without grant funding
- Early-career scholars
- Most of the Global South
- Independent and policy researchers

Their counterfactual is **a smaller N, a narrower question, or no study at all.**

Dimension	Human RA	Agent
Cost	\$15–25/hr × hrs	\$9 (Hormuz)
Availability	Grant-dependent	Anyone with API key
Consistency	Drift, fatigue	Same prompt, same rules
Documentation	Rarely logged	Full audit trail
Scale	50–200 cases	438 overnight
Errors	Random	Systematic bias
Validation	Rarely	*Should* be

The Honest Comparison

The validation we apply to LLMs —

- existence checks (refetch the URL, verify the quote)
- prompt sensitivity (does the answer survive a rewording?)
- inter-model κ (does Claude agree with GPT?)

— is stricter than what is standard for human-coded data.

That is a feature of the method,
not a concession about its limitations.

The Question

The event: On 28 February 2026, the US launched military operations against Iran. Strait of Hormuz effectively closed shortly after.

The puzzle: Economic costs were not distributed evenly across US Congressional districts.

- Petrochemicals, freight logistics ⇒ acute disruption
- Finance, education, legal services ⇒ insulated

The hypothesis: Legislators from more exposed districts express greater opposition to the conflict.

Core prediction

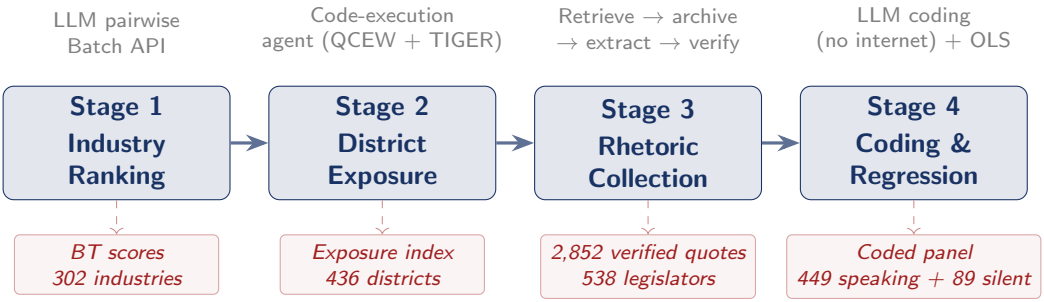
$$\underbrace{\text{Exposure}_d}_{\text{district}} \Rightarrow \underbrace{\text{public opposition}_i}_{\text{legislator}}$$

$\beta_{\text{exposure}} < 0$

The challenge

Build, code, and analyse rhetoric for all 538 members of Congress — within *days* of the event, by myself.

A Four-Stage Agentic Pipeline



Agents at every stage. The researcher writes the *spec*; agents do the labour.

Each stage is a checkpoint. Pipeline is resumable; failures are local.

Zoom In on Stage 3 — Where the Agents Lives

The seduction. A single API call with a built-in web tool returns a coded political position in 30 seconds for a few cents.

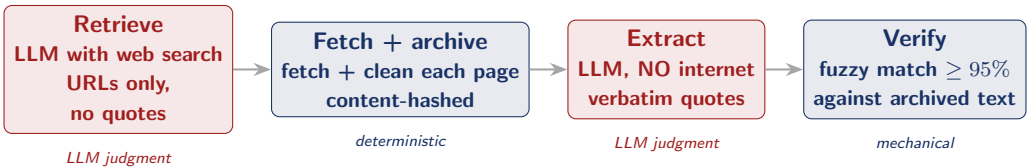
What previously needed a trained RA team, an inter-rater protocol, and weeks of coding can now run *overnight on a laptop*.

The danger

That same single call:

- **Logs no queries** — you can't replicate what was searched.
- **Returns no source text** — nothing to verify the quote against.
- **Cannot distinguish** "no statement exists" from "the model invented a different one."
- **Conflates retrieval and extraction** — one black box does both.

The Architecture: Search, Archive, Extract, Verify



The rule: the LLM is allowed to *search*, but not to pretend it has *read* what it searched.

- Retrieval call returns queries and URLs only — no quotation, no synthesis.
- Every URL fetched independently, archived as raw HTML + cleaned text with content hashes.
- Extraction call has no internet — it can only read the archive.
- Every extracted quote mechanically checked against the archive; failures dropped *before* coding.

The Stopping Rule: When the Agent Stops Searching

Each tier is a separate agent call with its own prompt.

The agent issues only the queries the current prompt tells it to. The *pipeline* decides whether to issue the next-tier prompt.

Three exit conditions:

1. **Success.** ≥ 1 verified quote in the current tier $\rightarrow$ stop, code the quote(s).
2. **Failure.** Tier T4 returns 0 verified quotes $\rightarrow$ emit *no recoverable statement*.
3. **Resource cap. 17 queries per legislator** across all tiers $(2{+}3{+}4{+}4{+}4)$.

The agent never says "let me try one more thing."
The stop is owned by the pipeline.

The escalation ladder:

Tier	Q	Triggered when	Search strategy
T0	2	always	Canonical: *name* + *Iran Hormuz 2026*
T1	3	T0 = 0	Name variants
T2	4	T1 = 0	Channel: `site:`.gov, socials
T3	4	T2 = 0	Proxy events: votes, letters
T4	4	T3 = 0	District press
T5	0	T4 = 0	**Stop** — emit silence

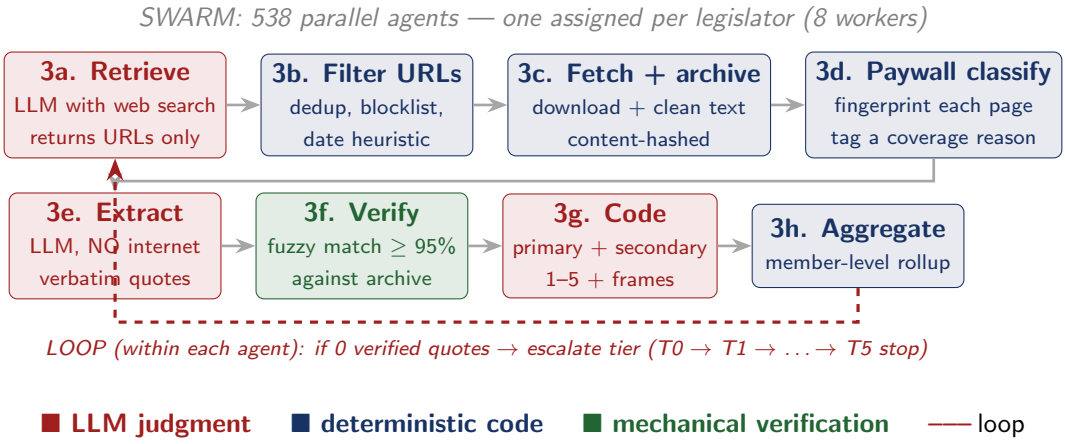
Q = number of web searches in that tier's prompt.

Inside the Retrieve Call: The Tier 0 Prompt

T0 is one of **five** structurally identical retrieval prompts (T0–T4). If T0 yields 0 verified quotes, the pipeline fires a *fresh agent call* with the T1 prompt; same for T2–T4.

```
You are a research assistant locating public statements made by US legislators
about the February 28, 2026 US military operations against Iran and the
Strait of Hormuz crisis.

Your job in this call is RETRIEVAL ONLY. Issue web searches via the web_search
tool, then return a JSON object listing the queries you issued and the URLs
you surfaced. You MUST NOT extract quotes, summarize articles, paraphrase
content, or write any prose beyond the JSON output.

TASK -- TIER 0 (CANONICAL)
Issue exactly TWO search queries:
  1. "{full_name}" Iran Hormuz 2026
  2. "{full_name}" Iran statement 2026
Both queries MUST contain "2026" to filter out 1979 / 2015 coverage.

OUTPUT -- STRICT JSON, NO PREAMBLE:
{
  "queries_issued": ["string", ...],
  "urls_found": [{"url": "...", "title": "...",
                  "snippet": "...verbatim from search tool...",
                  "rank": 1, "appeared_for_query": "..."}, ...]
}

CONSTRAINTS
- Do NOT include URLs from before 2026-02-21.
- Do NOT extract or quote article content.
- Snippets must be verbatim from the search tool, not your summary.
```

Appendix: The Rhetoric Pipeline in Detail



Appendix: Prompt 2 — Extract

```
You are extracting verbatim quotes from a single archived news
article attributed to a specific US legislator.

You DO NOT have internet access. The only text available is what
is included below under SOURCE TEXT. If a fact, date, or quote is
not in that text, you do NOT know it. Do not invent.

SOURCE TEXT
=== BEGIN SOURCE TEXT ===
{source_text}
=== END SOURCE TEXT ===

TASK
Identify EVERY passage that is a direct quote attributed to
{full_name} OR a clear paraphrase of a statement they made about
the Iran / Hormuz crisis. For each, output:
  - claimed_quote: EXACT verbatim chars from source
  - offset_start / _end: 0-indexed character offsets
  - is_direct_quote: bool
  - claimed_date: YYYY-MM-DD or null  (do NOT guess)
  - context: 1-sentence note (<=200 chars)

ABSOLUTE RULES
- claimed_quote MUST appear verbatim; downstream verification
  rejects any quote at fuzzy ratio < 95.
- Do NOT paraphrase or normalise punctuation.
- Do NOT use prior knowledge of the legislator or event.
- Output JSON only.
```

The Codebook: What the Agents Actually Apply

The research instrument.

Each verified quote scored 1–5 on *conflict support*:

1	Strongly opposes
2	Somewhat opposes
3	Neutral / procedural
4	Somewhat supports
5	Strongly supports

Plus 7 binary frames: economic cost, security, constituent impact, industry mention, admin praise / criticism, diplomacy.

Building the codebook is the research design.

1. Draft v0 from the construct in the literature.
2. Hand-code 30–50 cases yourself; revise rules where your intuition disagreed.
3. Add worked examples for the hard cases; write an explicit "*if ambiguous, do X*" rule.
4. Pilot on the LLM; compare to your hand-codes; revise.
5. Lock the prompt, hash it, pin the model.

What the codebook doesn't say, the model's defaults will say — and those aren't your defaults.

The Validity Checks

Mechanical hallucination check

Every extracted quote must match the archived page text at $\geq$**95% character similarity**.

2,910 extraction attempts $\rightarrow$ **2,852** verified $\rightarrow$ **58** rejected = **2.0%** hallucination rate.

Reviewers don't have to trust the LLM — the verifier doesn't either.

Coverage as a taxonomy

449 speaking + **89** silent.

Each silent member carries a coverage reason. Silence isn't dropped — it's classified.

Failure register (no free-text notes)

Page fetch failed	2,198
Source text empty	490
Bot challenge	303
Paywall / forbidden	146
Quote not verified (hallucination)	**58**
Human review required	**0**

Identifier lineage

Every artifact is content-hashed.

query $\rightarrow$ URL $\rightarrow$ raw page $\rightarrow$ cleaned text $\rightarrow$ extraction $\rightarrow$ verified quote $\rightarrow$ coding

A reviewer can re-derive the chain.

The Numbers

Scale

- **538** legislators (House + Senate)
- **2,852** verified quotes (avg 5.3 / speaking member)
- **6,899** LLM API calls + **1,525** web searches
- **1.5 hours** wall-clock, 8 parallel workers

Cost

- Total: **$62.58**
- Comparable RA-coded study: $1,500
- Pipeline is $\sim$**24**$\times$ **cheaper**

Quality

2.0% mechanically detected hallucination rate.

Zero items required human review.

Every quote traceable to its archived source.

Substantive result (illustrative)

Pooled OLS, controls: $\beta_{\text{exposure}} = +0.035$, $p = 0.258$ — **null**.

Most robust within-party finding: asymmetric *constituent-impact framing* — exposed Dems invoke it more, exposed Reps invoke it less.

Lesson 1: Pilot One Agent Before You Launch 500

What happened.

1. Launched 10 parallel sub-agents to run the legislator searches.
2. All 10 returned empty results.
3. Cause: **sub-agents do not inherit the parent's tool permissions**. The web-search tool was silently disabled.
4. $\sim$20 minutes burned before I noticed.

The fix. Move the search into the parent process; one explicit tool call per legislator; rate-limit; checkpoint to disk.

Pre-flight checklist before any swarm

- Run one agent end-to-end with *explicit* tool calls.
- Verify tool permissions propagate to sub-agents.
- Pin the output schema; pilot on 3 cases.
- Only then scale.

Checkpoint and resume

Persist raw results every 25 records. On restart, skip done IDs.

The pipeline becomes restartable at zero cost.

Lesson 2: Treat Your Agent Like a Hungover RA

A bright but unreliable RA shows up Monday morning. They will work fast, follow instructions literally, and confidently invent things they don't know.

You wouldn't publish from their first pass without:

- Spot-checking the citations
- Rephrasing the brief and seeing if the answers move
- Having a second RA recode a subsample

Apply the same standard to your agent. The methodology literature treats this as best practice for LLMs — it is just the standard we *should* have been applying to RAs all along.

The three checks — non-negotiable

1. **Existence check** (hallucination audit). Refetch every cited URL; verify the quote appears. $<5\%$ unverifiable.
2. **Prompt stability.** Re-run with 2–3 query variants. Report cross-variant DV correlation. $>70\%$ statement overlap.
3. **Inter-model reliability.** Recode with a second model family (Claude/Gemini if you used GPT). Cohen's $\kappa > 0.70$.

If your finding doesn't survive a different prompt and a different model, it isn't a finding — it's a prompt artefact.

Lesson 3: Do a Cost Audit

Where the Hormuz study actually spent money:

Task	Calls	Cost
Stage 1: pairwise BT ranking	45,451	\$4.50
Stage 3: retrieval (LLM + tool)	1,498	\$49.91
Stage 3: extraction (no internet)	2,549	\$5.60
Stage 4: coding	2,852	\$2.57
Total	**52,350**	**\$62.58**

Comparable RA-coded study: \$15/hr $\times$ 100 hr $\approx$ \$1,500. Pipeline is $\sim$24$\times$ cheaper.

What actually happened

Audit projection: **\$52.07**. Actual: **\$58.07** on rhetoric stages — 11.5% over.

The dry-run on 3 hand-picked legislators systematically *under-estimated* escalation depth.

Lessons

- Pilot on a *stratified* sample, not on easy cases.
- Wire a real-time **kill-switch**, not just a pre-run gate.
- Make resume-from-checkpoint first-class.

Other Uses of Agents

Some other ideas

- Collect all budget bills from EU countries $\rightarrow$ code existence of a policy $\rightarrow$ find correlates of that policy.
- "Needle in a Haystack" - structured web searches (in different languages) for that one piece of qualitative evidence you're looking for - then follow up.
- Take an existing dataset - ParlEE - filter for domain, run agents to find relevant speeches. Code it. Find correlates. Run Models.

Surveys That Listen Back

The traditional survey is a script.
The questionnaire is fixed. The respondent's answers cannot reshape the next question.

An agentic survey is a conversation.

- An LLM reads the participant's open-ended response in real time.
- It decides what to ask next from a pool of candidate items.
- It can *create* new items from earlier responses.
- It can probe ambiguity, ask for examples, or pivot.

The survey adapts to the respondent — not the other way around.

Why this is hard with classic methods

- Branching logic is brittle and pre-specified.
- Open-ended responses are coded *after* the fact.
- Salient issues that researchers didn't anticipate go unmeasured.

LLMs solve all three problems at once.

The Field Is Splitting Into Three Camps

1. **AI-led interviews at scale.** The LLM runs the whole semi-structured interview with adaptive follow-ups.

   Geiecke & Jaravel, "Conversations at Scale"; Wuttke et al., "AI Conversational Interviewing"; **Anthropic Interviewer** (1,250 → 81,000 interviews, 159 countries, 70 languages).

2. **AI-assisted human interviewing.** Human stays primary; LLM suggests probes, flags missed themes, supports note-taking.

   Zhang et al. (real-time follow-up generation); Wen et al. ("unobtrusive co-interviewer").

3. **Agentic / multi-agent systems.** The interviewer plans, tracks coverage, simulates conversational rollouts, updates strategy mid-interview, sometimes codes downstream.

   SparkMe (Anugraha, Padmakumar, Yang) — interview as an optimisation problem; Panfilova et al. 2026 — evaluation across 6 frontier models.

The seductive claim: depth-of-interview at survey scale.
The hard problem: "depth" is rapport, trust, silence, embodiment, reflexivity — not just longer text.

Bottom Line

What we can defend now

- Large-N *exploratory* interviewing.
- Open-ended survey modules with 1–3 adaptive probes.
- **Asynchronous adaptive interviewing** — the real coordination win: respondent clicks a link, gets tailored probes, no scheduled interviewer.

What we cannot defend

- LLM *as respondent* for primary qualitative data.
  (Kapania et al., "Simulacrum of Stories.")
- LLM as *interpreter* — it applies the codebook, doesn't write it.

Two risks the field hasn't reckoned with

Re-identification. Anthropic's 1,250-interview dataset → Tianshi Li (2026) re-identified scientists using agentic web search. Rich transcripts are uniquely identifying.

Leading effects. LLMs are trained to be helpful. They over-affirm, normalise assumptions, ask subtly leading probes — worst on sensitive topics.

The honest framing:
adaptive open-ended measurement at survey scale.

Step 1: Simulating Respondents ("Silicon Samples")

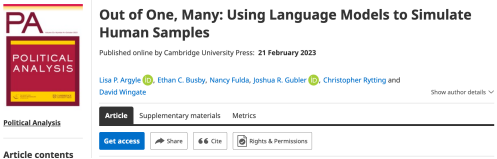
The idea. Prompt an LLM to take on a persona (age, income, party, region) and answer a survey *as that person would*.

Good for

- Pre-testing instrument wording.
- Sketching expected effect sizes.
- Stress-testing designs cheaply.

Not good for

- Estimating actual population means.
- Heterogeneity the training data underrepresents.
- Replacing real respondents in published estimands.



Synthetic Pennsylvania samples track marginals well but mis-estimate cross-tabs and minority opinion.

The Building Block

A persona + a prompt = a synthetic respondent.

An + a defined interest = actor in a game theory model

What If the Personas Interact?

The personas in Steps 1 and 2 are *static*.
They answer; they don't *interact*.

What changes when they do?

- A synthetic deliberation between citizen and expert.
- A synthetic legislature debating a bill.
- A synthetic crisis cabinet weighing escalation.

That is the next step.

Step 3: Generative Agent Societies (Park et al. 2023)

Populate a virtual town with 25 LLM-powered agents — each with a persistent identity, memory, and goals — and let them *interact*.

- Agents **remember**, **reflect**, and **plan**.
- Emergent social behaviour follows: they organise a party, spread information, form relationships — without being told to.
- Architecture: perception $\rightarrow$ memory stream $\rightarrow$ retrieval $\rightarrow$ reflection $\rightarrow$ action.

This is where *agent-based modelling* meets LLMs.

The Natural Extension: Agentic War Games

Same architecture, raised stakes:

- **Crisis bargaining.** Iran, Israel, US, allies as separate agents with mandates and red lines — run the next 30 days a thousand times.
- **Legislative coalition formation** under counterfactual rule changes.
- **Information ecosystems.** How does a deliberative public respond to a misinformation shock?

Used today by **RAND, Hoover, several Ministries of Defence** — with all the validity caveats that follow.

The validity warning

Agents behave like *the median text on the internet*, not necessarily like humans.

Treat outputs as **hypothesis-generating** or as a stress test of your own theory — **not as evidence about the world.**

The Tools You Actually Need

In R

- **ellmer** — Call any LLM API (OpenAI, Anthropic, Google, Ollama). Structured output, tool calling, batching. Tidyverse-native.
- **quallmer** — AI-assisted qualitative coding with κ and F1 baked in.

In Python

- Official SDKs from OpenAI, Anthropic, Google.
- Structured-output helpers (Instructor, Pydantic).
- Batch APIs from every major provider (50% discount).

Agent runtimes you can use today

- **Claude Code**, **Codex** (OpenAI), **Gemini CLI** — give a research task, come back later.
- **Cursor**, **Windsurf**, **Antigravity**, **Positron** — IDEs that read your whole project folder.

Local (sensitive data)

- **Ollama** + Gemma 3, Qwen 3, Llama 3. Free, private, no API costs. 16 GB RAM opens most options.

Tips and Tricks from the Trenches

Claude Code as orchestrator

Give Claude Code (or Codex) an **API key**. It can now spawn its own LLM calls — write a coding script, submit a batch, fetch results, all unattended.

The IDE becomes the *outer* agent that runs an *inner* fleet.

Use the Batch API

Every major provider offers it: **50% discount**, results within 24 hours. Indispensable for any task above a few thousand calls.

Pilot first ($\sim$100 calls); read the `usage` field; project; *then* submit.

Have Codex and Claude check each other

Different model families = free **inter-model reliability**.

One drafts code, the other reviews. One codes statements, the other recodes them. Report κ.

Cost guardrails (non-negotiable)

Agents will happily spend \$500 in 4 minutes.

Turn off auto-recharge. Hard stop when credits run out.

Add a **Skill / hook** that blocks batches without an approved cost audit.

The Mindset Shift

You are no longer the typist. You are the manager.

- Give the agent context: data, codebook, draft, constraints.
- Specify the deliverable as if briefing an RA.
- Demand checkpoints and a log.
- Read the diff. Question the choices.
- Validate, validate, validate.

The agent is patient, fast, cheap, consistent, and *wrong in interesting ways*.

Your job is to design the workflow that catches the wrongness before it reaches the table in the paper.

What hasn't changed

- Construct validity is still on you.
- Identification is still on you.
- The theory is still on you.

What has

The cost of running an ambitious empirical project alone has fallen by an order of magnitude.

The set of people who can do that work has expanded.

Closing

Agents won't write your paper (well).
They will let you write a paper you couldn't write alone.

The methodological work — audit trails, validation, honest construct definition — is what makes the difference between a press release and a publication.

Thank you.

m.r.digiuseppe@fsw.leidenuniv.nl

Concurrent Work Worth Citing

Methodologically related papers that landed alongside this pipeline:

Afonso, Galiani, Gálvez & Sosa (2026)

Deep Research on a Loop: Using AI Agents to Construct Economic Datasets.
NBER Working Paper

Proposes **DRIL** — two-stage agent architecture (design → implementation), frozen protocol, citation contract, explicit data-gap taxonomy. Applied to the Global Tax Expenditures Database for eight LAC countries. Strong overlap with Stage 3 of this pipeline; we add mechanical quote verification and a strict retrieval/extraction split.

Poast (work in progress, University of Chicago)

LLMs and Data Management for International Relations Scholars: Moving Beyond EuGene-Style Software. SSRN

Positions LLM-based pipelines as the successor to EuGene-era IR data-management tools — same architectural family, different substantive domain. The convergence is the point.

Shameless Plug: Scaling Open-Ended Survey Responses

JOURNAL ARTICLE

Scaling Open-Ended Survey Responses Using LLM-Paired Comparisons

Matthew R DiGiuseppe ✉, Michael E Flynn

Public Opinion Quarterly, nfag013, https://doi.org/10.1093/poq/nfag013
Published: 27 March 2026

PDF Split View Cite Permissions Share ▾

Abstract

Survey researchers rely heavily on closed-ended questions to measure latent respondent characteristics like knowledge, policy positions, emotions, ideology, and various other traits. Closed-ended questions are easy to analyze and collect, but necessarily limit the depth and variability of responses. Open-ended responses allow for greater depth and variability in responses, but are labor intensive to code. Large language models (LLMs) may help with this problem, but existing approaches to using LLMs have a number of limitations. In this paper, we

DiGiuseppe & Flynn (2026).
Public Opinion Quarterly.

The problem: open-ended survey responses are rich, but coding them by hand doesn't scale.

The method: **LLM-paired comparisons.** Ask the model "which respondent shows more economic knowledge?" thousands of times; recover a latent score via Bradley–Terry.

Same agentic primitive — pairwise LLM scaling — that Stage 1 of the Hormuz pipeline uses to rank industries.

doi.org/10.1093/poq/nfag013

Shameless Plug: MStack

A Claude Code plugin for academic research.

~34 slash commands that walk a paper across every stage — with the *right forcing questions* at the right moment.

Stage	Example commands
Ideate	`/research-question`, `/scope-challenge`
Map	`/lit-map`, `/theory-build`
Design	`/identification-review`, `/preregister`
Build	`/data-acquire`, `/codebook`
Analyse	`/robustness`, `/results-audit`
Write	`/draft-section`, `/abstract-shotgun`
Submit	`/journal-fit`, `/referee-mock`
R&R	`/r-and-r`, `/coauthor-review`

Design principle

Role > prompt.

Every skill speaks in a defined voice — advisor, methodologist, referee, replicator. The forcing questions are baked in.

Install (two lines in Claude Code)

```
/plugin marketplace add
  matthewdigiuseppe/MStack
/plugin install mstack@mstack
```

github.com/matthewdigiuseppe/MStack

Built for IPE. Generalises to most quantitative social science.

Questions?

Appendix: Building the District Exposure Score

Step 1: Industry vulnerability score.

Pairwise LLM comparisons across 302 NAICS industries:

"Which suffers greater net harm from a Hormuz closure?"

Bradley–Terry recovers $\hat{\beta}_i$.

Step 2: Employment shares.

BLS QCEW 2023 county $\times$ 4-digit NAICS counts E_{ci}.

Step 3: County $\rightarrow$ district crosswalk.

TIGER 2025 shapefiles; area-weighted allocation to congressional district d, giving E_{di}.

Step 4: Weighted exposure index.

$$\text{Exposure}_d \;=\; \sum_i \frac{E_{di}}{E_d} \cdot \hat{\beta}_i$$

Most exposed districts:

District	Dominant industry	z
TN-04	Resin/synthetics mfg	$+2.59$
OH-05	Resin/synthetics mfg	$+2.33$
CO-08	Resin/synthetics mfg	$+2.05$
TX-14	Resin/synthetics mfg	$+1.99$

Least exposed districts:

District	Dominant industry	z
DC-98	Financial services	-3.74
NY-12	Financial services	-3.75
NY-10	Financial services	-3.57
CA-11	Financial services	-2.98

The agent downloaded BLS & Census files, parsed NAICS, built the crosswalk, computed the index, wrote the script. *I read the diff.*

Appendix: Prompt 3 — Code

```
You are coding a single quote attributed to a US legislator about
the February 28, 2026 US military operations against Iran and the
Strait of Hormuz crisis. You produce structured codes only.

You DO NOT have internet access. Use only the information below.

QUOTE
  Verbatim text:     {verified_text}
  Surrounding context (<=200 char): {context}

TASK -- code on the following dimensions:

1. conflict_support  -- integer 1-5
   1 = strongly opposes / calls for withdrawal
   2 = opposes / urges restraint / questions rationale
   3 = neutral / procedural / declines to take a position
   4 = supports / endorses the operation
   5 = strongly supports / calls for escalation

2. economic_cost_frame         -- bool
3. security_frame              -- bool
4. mentions_constituent_impact -- bool
5. criticizes_administration   -- bool
6. praises_administration      -- bool
7. calls_for_diplomacy         -- bool

ABSOLUTE RULES
- Code only what the quote actually says.
- Do NOT infer the legislator's general position from prior
  knowledge of them.
- If genuinely ambiguous, set conflict_support to 3.
```