Ben Golub
Modern AI For Economics Research

On Thursday, December 18, Benjamin Golub joined Markus' Academy for a conversation on "Modern AI for Economics Research: An Overview of Tools." Benjamin Golub is a professor of economics and computer science at Northwestern University and Co-Founder of Refine.ink.

A few highlights from the discussion. Each section of the summary corresponds to a separate part of the conversation, published separately on YouTube.[1]

Talk overview:

- The first part is a tutorial for how to use Cursor, an AI-native environment that bakes LLMs directly into the editing workflow so that models can read your repository, edit multiple files, run "agent" tasks, or explain code in-context
- The second part presents Ben Golub and Yann Calvó's start-up Refine.ink, an AI tool to generate referee-style feedback to academic papers, identifying errors in math and empirical strategy, clarity problems, and consistency issues
- The third part walks through some best practices when prompting LLMs. Much of Golub's advice is also provided by Goldsmith Pinkham (2024)

Markus' Introduction

- In the future static PDFs might be replaced by interactive AI-dashboards that enable researchers to "talk to papers". Literature reviews might become living knowledge graphs
- Theory might begin modelling LLM-based agents with heterogeneous personalities, bounded rationality, and data-driven expectation formation
- Experiments may begin to study "synthetic subjects", allowing for running thousands of AI-based pilots before costly human experiments
- Future econometrics may rely even more heavily on unstructured data, AI scanning for instrumental variables, and double machine learning
- Peer review may become AI-augmented, with tools such as Refine.ink supporting referees and editors

[Video 1] AI-native environments: Cursor

- Cursor, built on VS Code, is an AI-native development environment that connects to GitHub repositories containing LaTeX, code, and bibliographic files, replacing or complementing Overleaf-based workflows
- Within the environment one can then access all of a project's documents. One can then assign tasks to LLMs and, through retrieval-augmented generation, attach files or snippets directly, improving performance
- The value of Cursor is that it is a much better orchestrator of different tools than chatbots, which are confined to your browser. Often with chatbots it can feel like managing a small bureaucracy with different tabs

[1] Summary produced by Pablo Balsinde (PhD student, Stockholm School of Economics)

- Within Cursor, Claude Opus 4.5 is best for arduous and simple tasks, like searching for papers and writing BibTeX entries. GPT 5.2 is best for math-heavy tasks
- Rather than attempting everything in a single pass, decompose technical tasks into staged prompts. For example when writing proofs: (1) first ask the model to assimilate the existing work, (2) then ask it to generate background notes, (3) then to design a proof strategy, and (4) finally ask it to produce the proof
- For those that prefer single-shot prompts, one can provide high-level instructions to orchestrate, for example telling the model to "orchestrate a good software engineering team for different tasks."
- Cursor is largely similar to GPT Codex and Claude Code. The key benefit is that it is model-agnostic, allowing for mixing and matching models to tasks
- If you are comfortable with the basic computer terminal but not an expert in software deployment, it might be best to grant Cursor permission to run basic command-line steps like installing packages

[Video 2] Refine.ink

- Faster AI-assisted research can reduce the slow, deep engagement that typically surfaces mistakes, increasing the chance that errors persist into drafts.
- Generic chatbots cannot guarantee logical consistency, creating demand for a dedicated, end-to-end audit layer.
- Refine is an AI-based academic reviewing service that produces deep referee-style reports, identifying issues such as errors in logic, clarity problems, and consistency issues. It can produce a referee report at the level of a better-than-average PhD student
- Specialized orchestration and domain-tuned workflows make Refine's commentary systematically deeper and more coherent than generic chatbots
- For example, Refine points out definition inconsistencies, flags clustering choices, and questions an IV's logic.
- Testimonials from Omer Tamuz and Drew Fudenberg report that Refine detects subtle mathematical errors and inconsistencies that would take chatbot-assisted human experts many hours to uncover
- Refine works for both theory and empirical papers (as well as in fields such as applied mathematics, computer science, and physics). For empirical work, the main current limitation is that tables and figures often cannot be appropriately parsed, an issue the tool's creators are actively working to improve
- It is best to ask Refine to give at most 10 comments on your paper. Use it before you circulate or submit papers; when you care about correctness
- Privacy is contractually enforced: uploaded papers are neither used for training nor exposed to external model providers
- Top journals are piloting Refine as a complement to human peer review before publication with ethical use enhanced by disclosure in referee reports.

[Video 3] Heuristics for working with AI

- Treat LLMs as a brilliant but limited junior assistants: strong intuition, algebra, and coding skills, but weak understanding of higher-level project objectives and norms
- Post-training biases models towards overconfident answers rather than acknowledging ignorance. It also biases (especially chatbots) toward narrow, local task completion rather than cross-domain discovery

- Models' "tunnel vision" leads them to execute the immediate instruction, ignoring the broader goal, even by sacrificing important checks or commenting out other code
- Prompt for rigor, not just content: pair instructions with explicit behavioral constraints on rigor, notation discipline, and stylistic consistency—especially by specifying the role and audience (e.g., "a senior probabilist writing for Econometrica")
- Ask it to reason step-by-step, even if you later request a concise final output. Ask it to explain things back to you in the process
- Decompose tasks to improve reasoning quality. Ask for a plan, and then implement it in small steps
- Don't fight the model. After a wrong answer, editing the prompt or starting a new chat is better than arguing within the same chat: the wrong answers can poison the model, inducing it to reconcile previous errors with later content
- Ask for "handoff reports", key summaries of what has been done and next actions. Then feed these reports to the next chat. Example: "Write a handoff to a junior colleague: include all essential information to continue the task."