

AI-Driven Research in EconCS Tutorial

Paul Dütting

Renato Paes Leme

ACM EC 2026, Rome, Italy

Google Research

This Tutorial: Goals and Outline

Goal 1: How can AI and LLMs help you?



Automating the Research Workflow

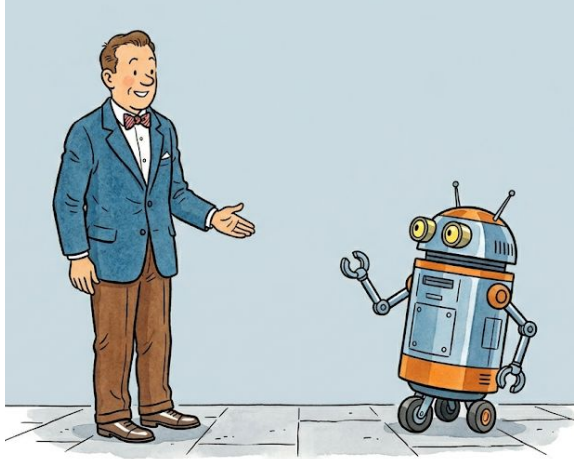
enabling more efficient
task execution



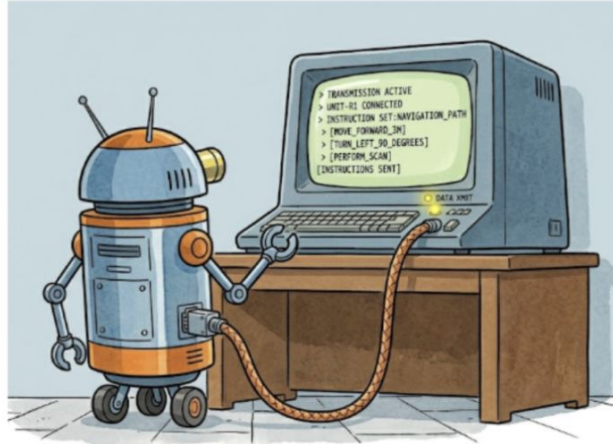
Generating Novel Insights

pushing the boundaries of
knowledge

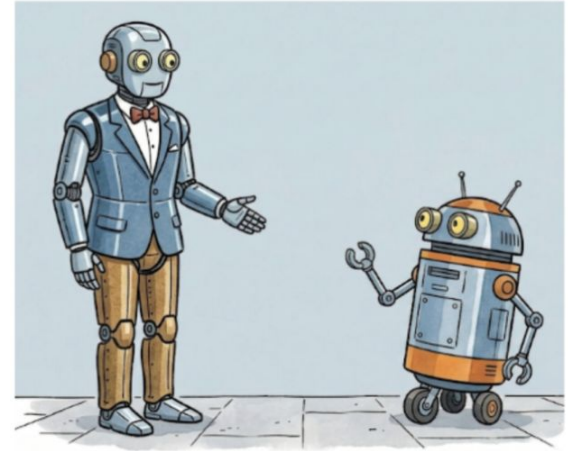
Goal 2: Automating the use of AI



Directly talk to the AI



Write programs that call the AI



Get AI to direct other AIs

Plan for this Tutorial

Focus on **Agentic Workflows**

(see [U Chile tutorial](#) and [Columbia tutorial](#) (w/ Santiago Balseiro) for other modes)

Part 1: Getting Started with Agents

Part 2: Automating the Research Workflow with Agents

Part 3: Theorem Proving with Agentic AI

Part 4: Data Science with Agentic AI

Workshop this afternoon

- There will be a workshop this afternoon, continuing the theme of this tutorial
 - Panel Discussion on “**How will AI tools change EconCS research and education?**”
Panelists: Yiling Chen, Yannai Gonczarowski and Matt Weinberg.
 - Keynote talk by Mark Sellke (OpenAI) “**On the Disproof of the Unit Distance Conjecture**”
 - Contributed talks on problem solving in Econ, CS and OR using AI
 - **EconCS Bench**: list of 20+ problems contributed by leaders of the community, as testbeds to track the advancement of AI tools in EconCS problems

Important disclaimers

Rapidly evolving technology

AI is evolving very fast, and the current tutorial reflects the technology right now. In a few months, the tools available could be very different.

Personal experience

Tutorial is based on our own experienced trying to do Research using AI and reflects our subfield (Theoretical Computer Science, Algorithmic Game Theory, Operations Research).

Our intuition and understanding is also evolving as we go.

Practical / Example-Driven

We will try to give many ideas of prompts and example of what can be done.

Gemini Examples

Examples will be based on Google technologies (Gemini, Gemini CLI, Colab, Antigravity,...) but the tutorial should apply to any similar technology out there.



Things to keep in mind

- **Try again if you had a bad experience in the past:** Tools have evolved remarkably over the past six months, resulting in significantly better performance.
- **Own it and declare it:** Most academic conferences and journals allow the use of AI, provided its application is explicitly declared.
- **Be aware of the pleasing bias:** LLMs are trained to be helpful, which means they will do their best to provide an answer even if it involves producing a highly plausible, yet entirely flawed, proof.

No magic bullet

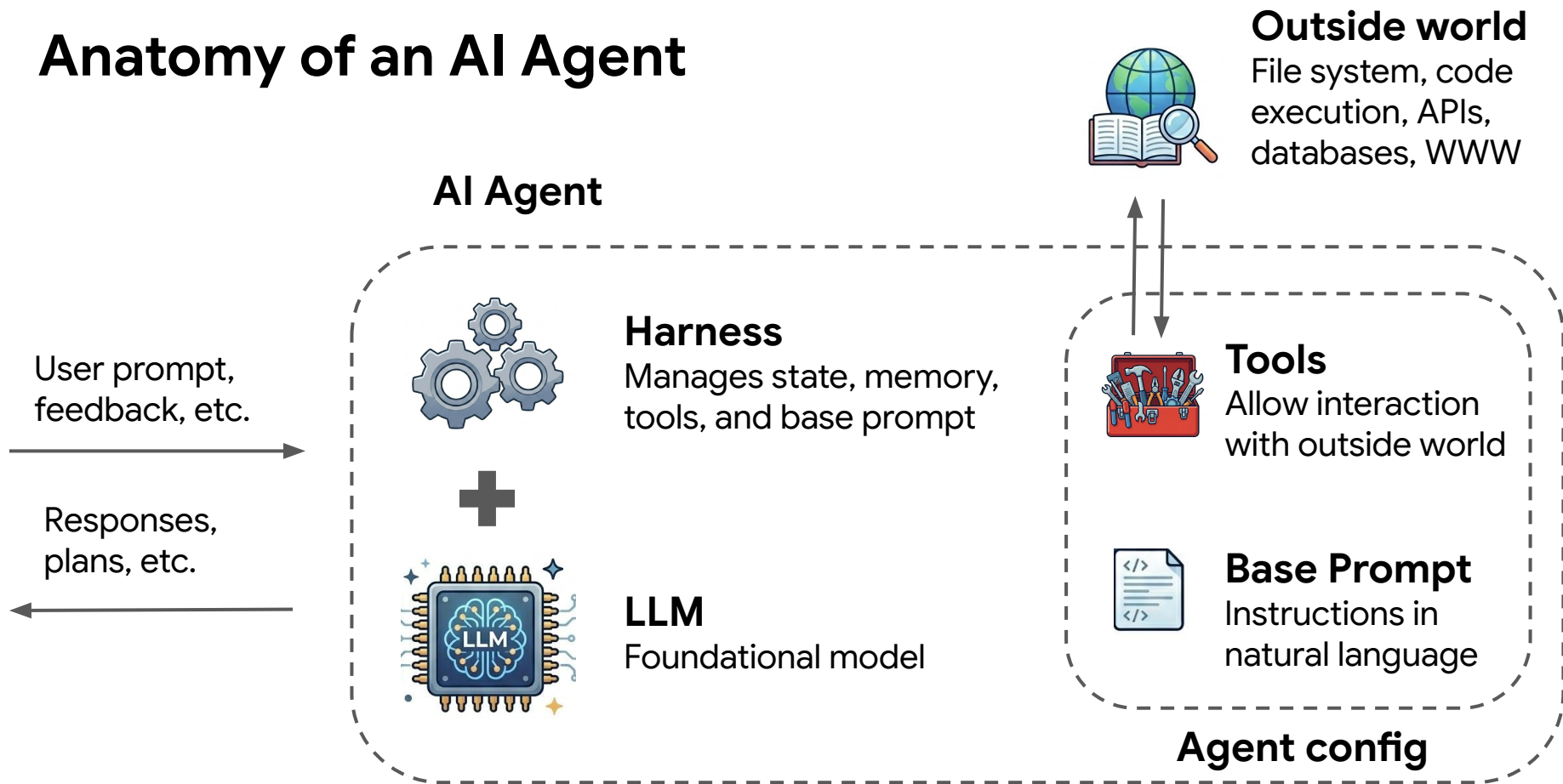
- Research still requires lot of effort (even with AI)
 - Asking the right questions & providing the right intuition
 - Ensuring scientific and mathematical rigor
 - Using the tools efficiently
- ... but you may end up spending time doing other things
 - Some formerly labor-intensive activities become effortless
 - You spend more time trying prompt variations, checking arguments, proof-reading, ...
- **Long run:** With tedious tasks out of the way, we hope AI will help us focus on what matters and allow us to tackle more difficult tasks.

Part 1: Getting Started with Agents

Why AI Agents?

- From **Passive** to **Active**: Shift the paradigm from generative AI (responding to prompts) to agentic AI (autonomous goal-pursuit)
- An **AI agent** is a system capable of reasoning through complex objectives, breaking tasks into steps, interacting with its environment, and executing actions using external tools

Anatomy of an AI Agent



Antigravity IDE (<https://antigravity.google/>)

- **Integrated Development Environment (IDE)** built on **VS Code**
- Native app originally designed for coding, but can do a lot more
 - Can compile code, access to computer tools (browser, terminal)
 - Edit and compile Latex files
 - **Agentic platform:** agents can perform tasks on your behalf
- Free of charge, but requires a Gmail account
- Similar software provided by other companies too



Set up instructions

- Download and install the **Antigravity IDE**:
 - Available from: <https://antigravity.google/download#antigravity-ide>
 - **Note:** Antigravity 2.0 is a separate application; we will specifically use the IDE.
 - You will need to login with a Gmail account
- **Optional recommended software** (required for specific demos):
 - Python: <https://www.python.org/downloads/>
 - Latex Distribution:
 - Windows: <https://miktex.org/>
 - MacOS: <https://www.tug.org/texlive/>
 - GitHub: <https://github.com/>
 - You may use the agent to verify if these tools are correctly installed on your system.

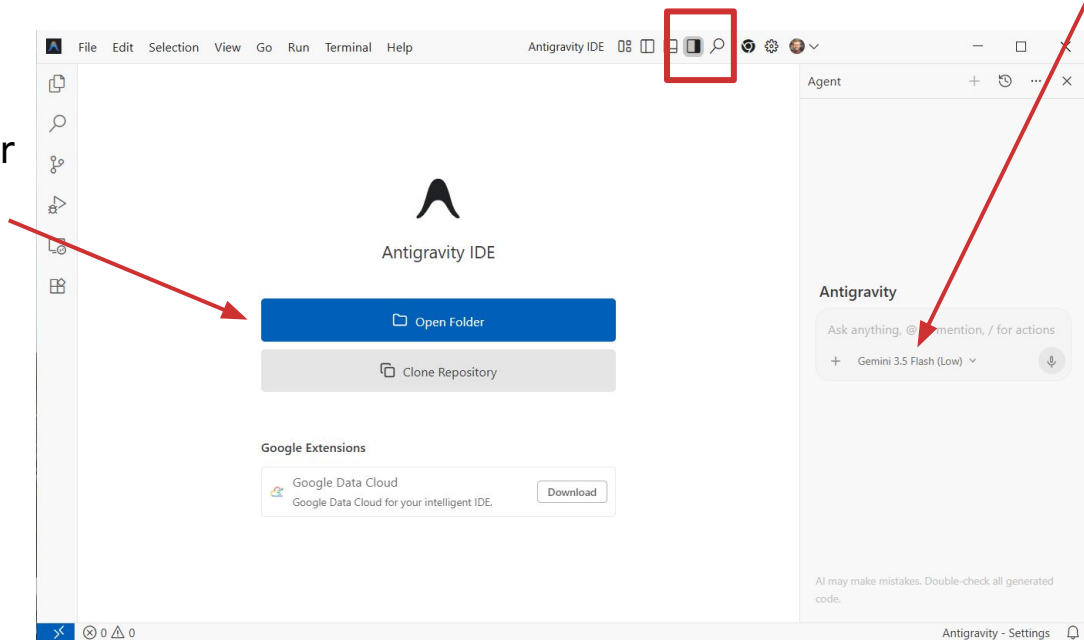
Welcome screen

Choose a folder on your computer to work

You can set up a git repository as well

Toggle agent sidebar

Choice of different models



First step:
File > Open Folder

The main view

Files

Text editor

User input to the agent

The screenshot displays the Overleaf web interface. On the left is the 'Explorer' panel showing a file tree under 'Overleaf_Projects'. The 'AGT_Lecture_Notes' folder is expanded, listing subfolders like '01_The_Complexity_of_Mixed_Nas...' and '02_Best_Response_Dynamics', and files like 'approxmd.bib' and 'approxmd.tex'. The 'approxmd.tex' file is selected. In the center is the 'Text editor' panel showing the LaTeX source code for 'approxmd.tex'. The code includes document class, package loading (beamerarticle, geometry, pgf, tikz, arrows, hyperref, natbib, hyperref, inputenc, xspace, amssymb, amsmath, amsthm, pspicture, pst-plot, pst-all, pst-plot, pst-3dplot, color, graphicx, verbatim, beamerthemesplit), and hyperref settings. On the right is the 'Agent' panel, titled 'Overleaf_Projects'. It contains a text input field with the placeholder 'Ask anything, @ to mention, / for actions', a dropdown menu showing 'Gemini 3.5 Flash (High)', and a microphone icon. At the bottom of the Agent panel, a disclaimer reads: 'AI may make mistakes. Double-check all generated code.'

Explorer

Overleaf_Projects

AGT_Lecture_Notes

01_The_Complexity_of_Mixed_Nas...

02_Best_Response_Dynamics

03_No_Regret_Learning

04_Price_of_Anarchy

05_Mechanism_Design_Basics

06_Optimal_Mechanisms

07_Poly_Time_Mechanisms

approxmd.bib

approxmd.tex

08_Prophet_Inequalities

09_Info_Design

10_Contract_Theory

bimatrixgame.sty

lecture-notes.sty

Text editor

approxmd.tex

```
1 \documentclass[11pt,a4paper]{article}
2
3 \usepackage{beamerarticle}
4 \usepackage[margin=1in]{geometry}
5 \usepackage{pgf,tikz}
6 \usetikzlibrary{arrows}
7
8 \usepackage{hyperref}
9
10 \usepackage{../lecture-notes}
11
12 \usepackage{natbib}
13 \usepackage{hyperref}
14 \hypersetup{
15   colorlinks=true,
16   citecolor = blue      % false: boxed links;
17 }
18
19 %\usepackage[latin1]{inputenc}
20 %\usepackage{xspace}
21 \usepackage{amssymb,amsmath,amsthm}
22 \usepackage{pspicture,pst-plot}
23 \usepackage{pst-all,pst-plot}
24 \usepackage{pst-3dplot}
25 \usepackage{color}
26 \usepackage{graphicx,verbatim}
27 %\usepackage{beamerthemesplit}
```

Agent

Overleaf_Projects

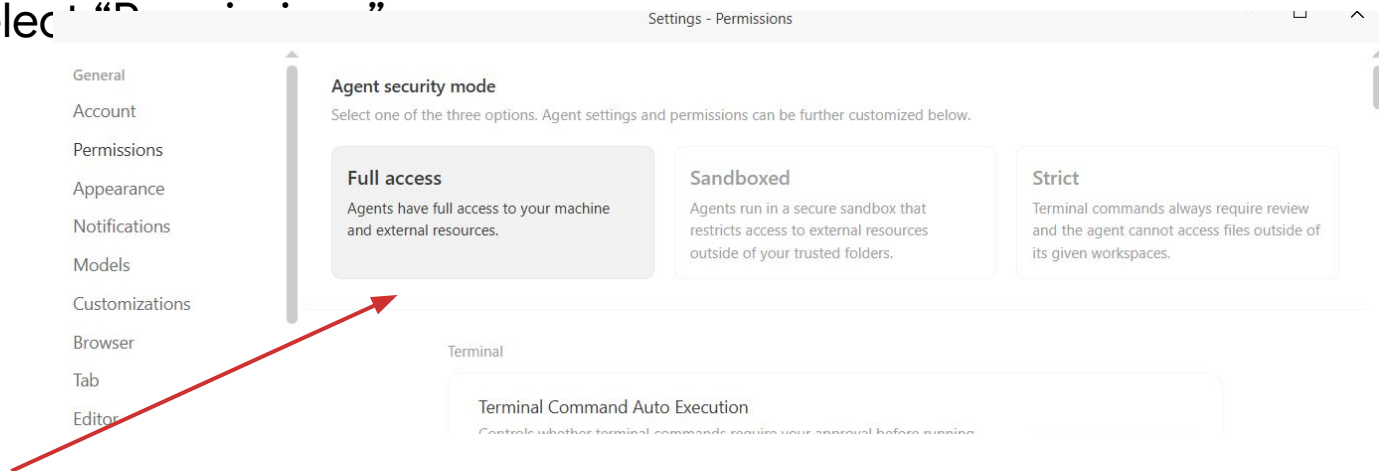
Ask anything, @ to mention, / for actions

+ Gemini 3.5 Flash (High) ▾

AI may make mistakes. Double-check all generated code.

Permissions or ‘How much do you trust agents?’

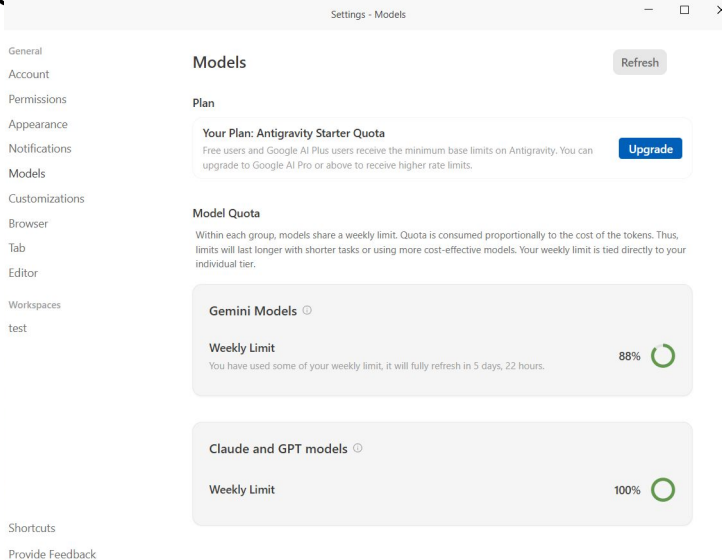
- Click on the gear (⚙️) on the top bar and “Open Antigravity IDE User Settings”
- Select “Permissions”



- YOLO (you only live once) mode allows the AI to execute file edits, terminal commands, and deletions without manually prompting you for approval
- You can also specify commands and directories are restricted

Model quotas

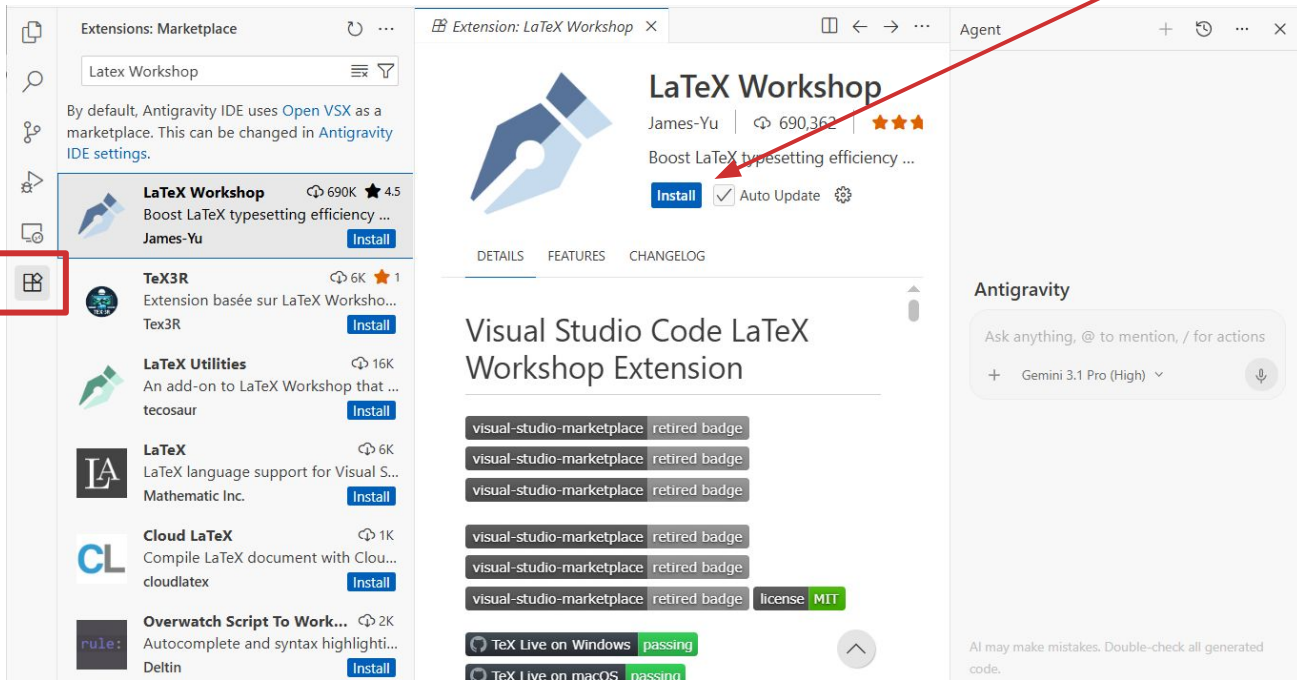
- Click on the gear (⚙️) on the top bar and “Open Antigravity IDE User Settings”
- Select “Models”



- Quotas refresh daily or weekly depending on the plan
- Free and Google AI Plus plans are very limited
- Google AI Pro provides good quotas with few-hours-long cycles

Latex workshop extension

Install Latex Workshop

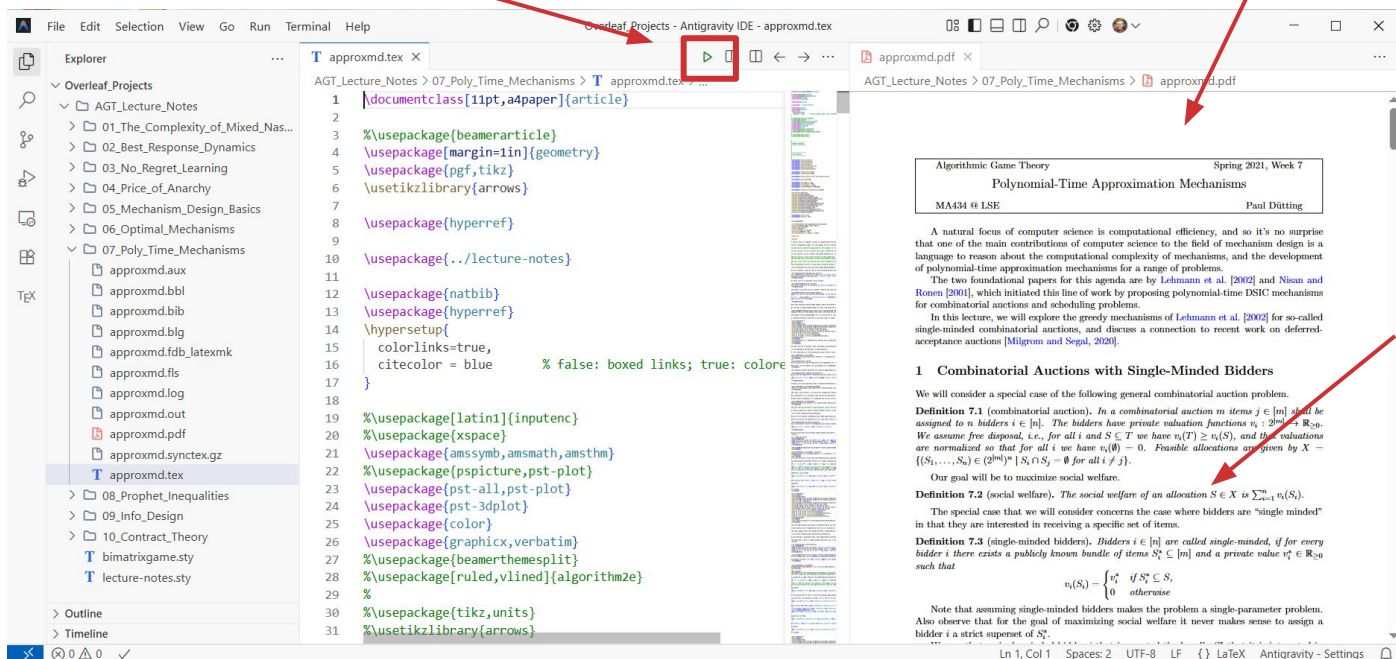


Extensions

Editing LaTeX documents

Compile file

Internal PDF viewer



Reverse search:
⌘ + click

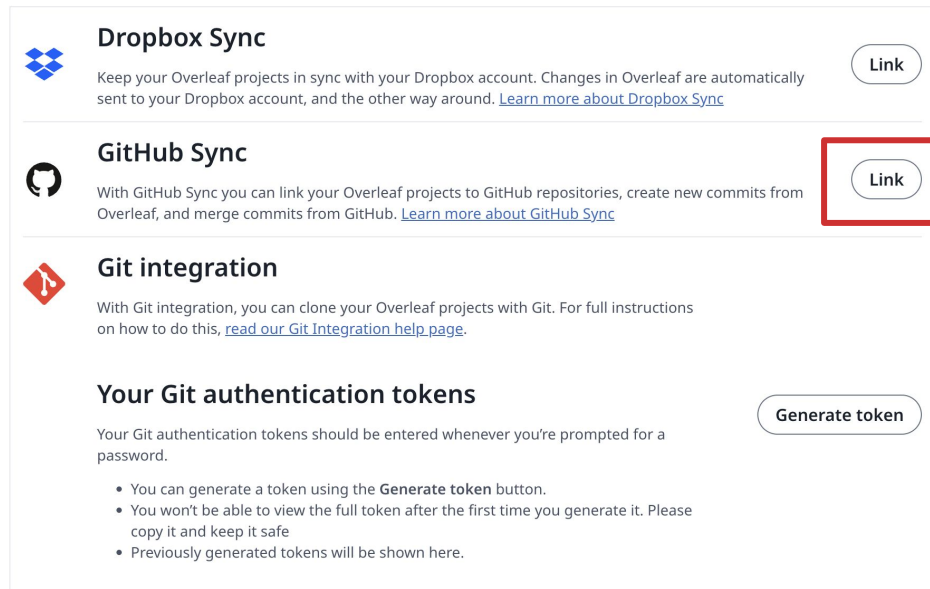
Part 2:

Automating the Research Workflow with Agents

Syncing and working on an Overleaf project

In Overleaf: Account (👤) > Account settings > Project synchronisation

Project synchronisation



The screenshot shows the 'Project synchronisation' settings page in Overleaf. It contains three main sections: 'Dropbox Sync', 'GitHub Sync', and 'Git integration'. The 'Dropbox Sync' section has a 'Link' button. The 'GitHub Sync' section has a 'Link' button, which is highlighted with a red rectangle and a red arrow pointing to it from the text 'Link GitHub account'. The 'Git integration' section has a 'Generate token' button. Below the 'Generate token' button, there is a list of instructions for using Git authentication tokens.

Dropbox Sync

Keep your Overleaf projects in sync with your Dropbox account. Changes in Overleaf are automatically sent to your Dropbox account, and the other way around. [Learn more about Dropbox Sync](#)

GitHub Sync

With GitHub Sync you can link your Overleaf projects to GitHub repositories, create new commits from Overleaf, and merge commits from GitHub. [Learn more about GitHub Sync](#)

Git integration

With Git integration, you can clone your Overleaf projects with Git. For full instructions on how to do this, [read our Git Integration help page](#).

Your Git authentication tokens

Your Git authentication tokens should be entered whenever you're prompted for a password.

- You can generate a token using the **Generate token** button.
- You won't be able to view the full token after the first time you generate it. Please copy it and keep it safe
- Previously generated tokens will be shown here.

Link GitHub account

Exporting an Overleaf project to GitHub

In Overleaf : Navigate to your project > Integrations > GitHub

The screenshot shows the Overleaf interface with the 'Integrations' panel open on the left. The 'GitHub' option is highlighted with a red box. A red arrow points from the text 'Integrations' to this box. Another red arrow points from the text 'GitHub' to the 'GitHub' option. The main dialog box is titled 'Sync with GitHub' and contains the following elements:

- Export Project to GitHub**: A heading for the dialog.
- Description (Optional)**: A text input field with a red arrow pointing to it from the text 'Add description'.
- Owner**: A dropdown menu with 'paulduetting' selected.
- Repository Name**: A text input field with 'Test-Project' entered.
- Public/Private Selection**: Two radio buttons. 'Private' is selected, indicated by a green dot.
- Create a GitHub repository**: A green button at the bottom right, highlighted with a red box. A red arrow points to it from the text 'Create'.

The background shows a code editor with a LaTeX document structure.

Finding the URL for a GitHub repo

Code

The screenshot shows the GitHub interface for a repository named 'Test-Project' by user 'paulduetting'. The repository is private. The 'Code' button is highlighted with a red arrow pointing to it from the word 'Code' above. A dropdown menu is open from the 'Code' button, showing options for cloning the repository. The 'Clone' option is selected, and the 'HTTPS' tab is active. The URL 'https://github.com/paulduetting/Test-Project' is displayed, with a red arrow pointing to it from the word 'Copy' on the right. The 'About' section on the right shows the repository is for an AI tutorial, with 0 stars, 0 watching, and 0 forks. The 'Releases' section shows no releases published, with a link to 'Create a new release'. The 'Packages' section shows no packages published, with a link to 'Publish your first package'. The 'Contributors' section shows one contributor, 'paulduetting'.

Code

Copy

Generating a GitHub personal access token

Generate and copy PAT

In GitHub: Click user (top right) > Settings > Developer settings

The screenshot shows the GitHub Developer Settings page. The left sidebar has a menu with 'GitHub Apps', 'OAuth Apps', and 'Personal access tokens'. Under 'Personal access tokens', there are 'Fine-grained tokens' and 'Tokens (classic)'. A red arrow points to 'Tokens (classic)'. The main content area is titled 'Personal access tokens (classic)' and has a 'Generate new token' button. A red arrow points to this button. Below the button, it says 'Tokens you have generated that can be used to access the [GitHub API](#).' There is a table with one row showing a 'Repo token' for a repository named 'repo'. The token is marked as 'Never used' and has a 'Delete' button. A warning icon and text state 'This token has no expiration date.' Below the table, a text box explains that 'Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).'

Settings / Developer Settings

Personal access tokens (classic)

Generate new token

Tokens you have generated that can be used to access the [GitHub API](#).

Repo token — repo	Never used	Delete
This token has no expiration date.		

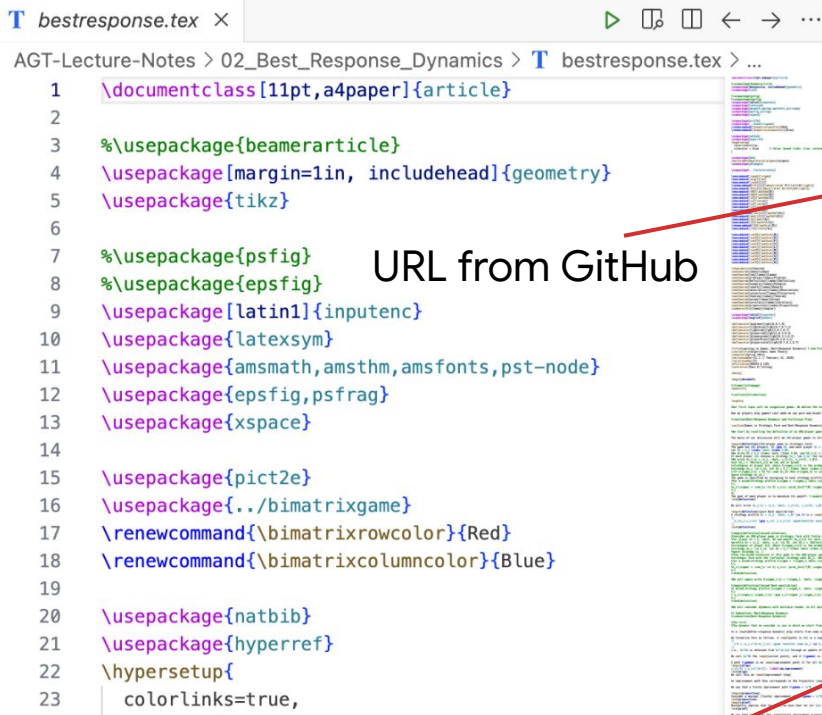
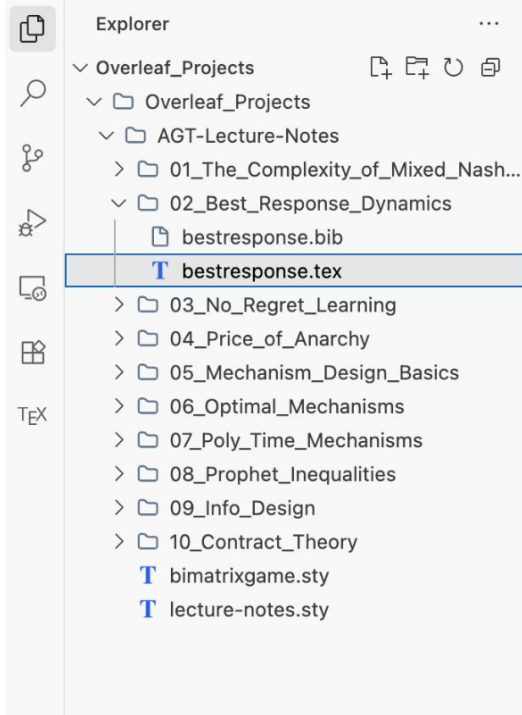
Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

© 2026 GitHub, Inc. Terms Privacy Security Status Community Docs Contact Manage cookies Do not share my personal information

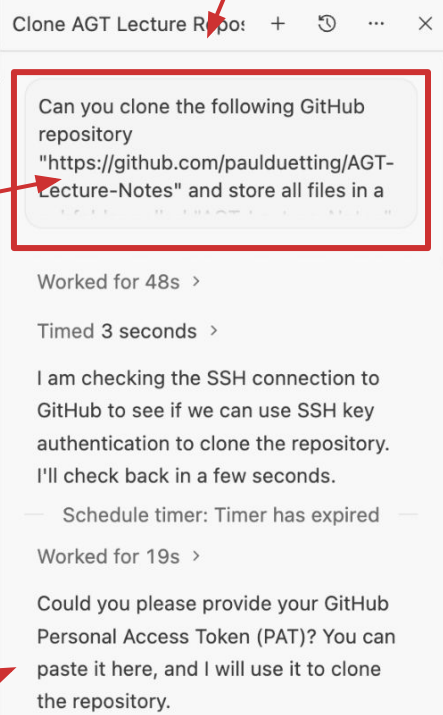
Personal access tokens

Cloning the GitHub repo from Antigravity

Ask agent to clone



URL from GitHub



Provide PAT

Pushing back to GitHub and Overleaf

URL & PAT from earlier

Can you run 'git status' for me to see which files in the /AGT_Lecture_Notes/ repo were changed?

Worked for 27s >

Here is the output of `git status` for the AGT-Lecture-Notes repository (note: the repository name uses hyphens instead of underscores):

The following files have been **modified**:

- 01_The_Complexity_of_Mixed_Nash_Equilibria/mixedNE.tex
- 02_Best_Response_Dynamics/bestresponse.tex

Additionally, there are several **untracked files**, which appear to be LaTeX compilation artifacts (such as `.pdf`, `.log`, `.aux`, `.bbl`, `.fdb_latexmk`, etc.) in the following directories:

- 01_The_Complexity_of_Mixed_Nash_Equilibria/
- 02_Best_Response_Dynamics/
- 05_Mechanism_Design_Basics/
- 06_Optimal_Mechanisms/
- 07_Poly_Time_Mechanisms/

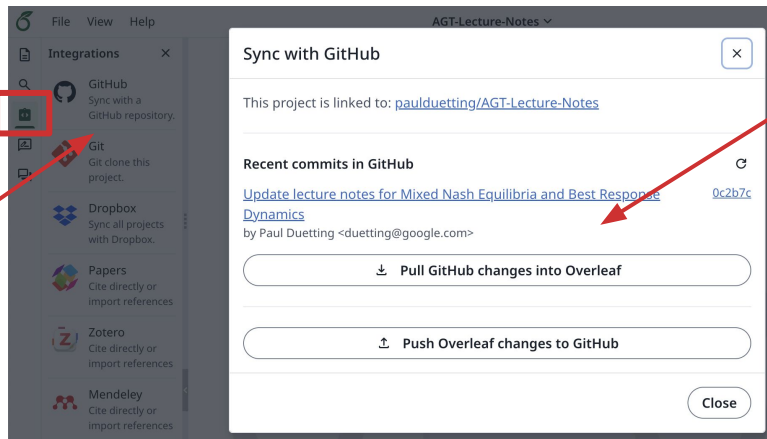
Let me know if you'd like me to review the specific changes in the modified `.tex` files or if you want to commit these changes!

- Use 'git push https://<your PAT>@github.com/paulduetting/AGT-Lecture-Notes.git main' to push changes pack to GitHub repo
- Alternatively, you can ask the agent to push

In Overleaf: Integrations > GitHub:

Integrations

GitHub



Pull

Compiling Latex documents

Compile

- Latex errors are famously cryptic and fixing compilation errors is frustrating
- Trying to compile **bestresponse.tex** via play symbol (▶) results in an error

Fix the compilation errors in bestresponse.tex

Worked for 2m >

I have fixed the compilation errors in bestresponse.tex.

What was causing the issue:

1. **TikZ coordinate parsing macro issue:** In the TikZ figure (around line 715), there was a `\foreach` loop iterating over a list of coordinates with parentheses, such as `(-1,1)`. The loop variable `\shiftamount` was directly fed into `\shift={\shiftamount}`. Because LaTeX expands the macro but TikZ's coordinate parser expects raw literals to correctly detect parentheses, it threw an error: `Package tikz Error: Cannot parse this coordinate.`
2. **Spacing in relative node shifting:** There were trailing spaces in the node positioning syntax (e.g., `node[shift={(x,y)}]`).

Accept Changes ✎ Reject ✎ ⬆ ⬇ ⬇ ⬇

Summary of changes (Tikz code was broken)

Accept or reject changes

Compiling Latex documents

After
accepting the
changes, the
paper and
figure compiles
without errors

bestresponse.pdf

AGT-Lecture-Notes > 02_Best_Response_Dynamics > bestresponse.pdf

Algorithmic Game Theory, Spring 2021 Week 2

Example 2.24. A sequence of (best response) improvement steps in a symmetric network congestion game that leads to a pure Nash equilibrium:

start:

after first improvement (red player):

after second improvement (blue player):

after third improvement (red player):

Remark 2.25. Note that as per our definition delays can be both positive and negative, and in general we do not require them to be monotone.

3.1 Improvement Sequences Converge

We proceed to show that congestion games admit an exact potential. A direct consequence

Proofreading

- Large language models are excellent at proofreading documents, checking grammar, and math formulas.

Overleaf_Projects

Can you carefully proofread Lecture 2 in `T bestresponse.tex`. Please carefully check grammar and spelling as well as mathematical correctness and notation. Create a list of potential issues and suggestions for improvements in a separate markdown file. Please don't touch any code yet.

+ Gemini 3.1 Pro ▾

`bestresponse.pdf` U Proofreading Report X

Proofreading Report 2 minutes ago

Review ▾

Proofreading Report: Best-Response Dynamics

I have carefully reviewed `bestresponse.tex` for grammar, spelling, notation, and mathematical correctness. Here is a list of potential issues and suggestions for improvement.

Typographical and Grammatical Errors

- Line 110: "Last week we say" $\rightarrow$ "Last week we saw"
- Line 183: "such" $\rightarrow$ "such that"

[...]

Mathematical and Logical Corrections

- Line 501 (FIP without Ordinal Potential): The sequence of inequalities is incorrect. Based on the given payoff matrix, the actual improvement steps are $(a,c) \rightarrow (b,c)$ (P1 improves 1 to 2), $(b,c) \rightarrow (b,d)$ (P2 improves 0 to 1), and $(a,d) \rightarrow (a,c)$ (P2 improves 0 to 2). At (b,d) and (a,d) , P1 receives 0 in both, meaning $\Phi(b,d) = \Phi(a,d)$. The sequence should therefore be $\Phi(a,c) < \Phi(b,c) < \Phi(b,d) = \Phi(a,d) < \Phi(a,c)$, exposing the contradiction. The text incorrectly states $\Phi(b,d) < \Phi(a,d) = \Phi(a,c)$.

[...]

	c	d
I	0	0
a	1	2
b	0	1
	2	0

I am not sure: the payoffs for player 1 and player 2 here are: $(a,c) \rightarrow (1,0)$, $(a,d) \rightarrow (2,0)$, $(b,c) \rightarrow (2,0)$, $(b,d) \rightarrow (0,1)$. Is it possible that you misread the bimatrix game?

Cancel

Add Comment

Worked for 54s >

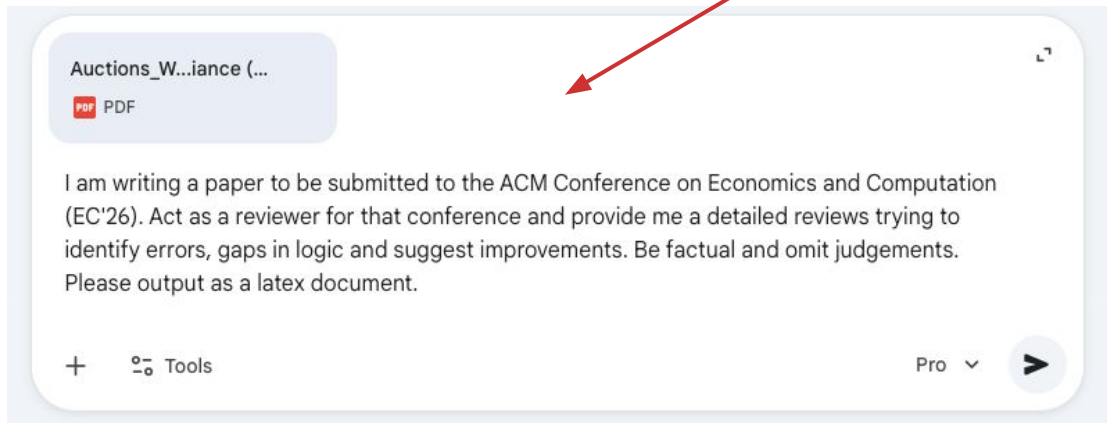
You are completely right! I misinterpreted the syntax of the `\payoffpairs` macro in `bimatrixgame.sty`.

Used @mention

Once you are happy with the plan, you can ask the agent to carry it out

Reviewing a paper

Doing this with an agent has several advantages: E.g.: “I checked and the typos identified are indeed typos. Please fix them all”



From chatbot mode, but also true for agentic mode

- Give as many details as possible
 - Which conference/journal you are submitting
 - What should reviewer focus on? (Grammar, math errors, gaps in logic, missing references, ...)
 - Maybe specify something about the reviewer (adversarial, unconvinced about your research agenda, looking for reasons to reject, ...)
 - It is useful to specify the output (latex document)

please create the same
review in canvas



Show thinking ▾

I have converted the review into a
LaTeX file that is ready for
compilation or submission.

Review of Revenue Variance

Minimization

Jan 30, 6:02 PM

[Try again without Canvas](#)



Let's write or build together

Review of Revenue...



Code



Preview



Share



Review of "Revenue Variance Minimization: Beyond First Price Auctions"

ACM EC'26 Review Committee

January 30, 2026

Summary of the Paper

This paper investigates the minimization of revenue variance in auction settings with risk-neutral, unit-demand bidders and independent private values. The authors aim to generalize the classical result of Waehrer et al. (1998), which established the optimality of the First-Price (or Winner-Pays-Bid, WPB) auction for variance minimization under ex-post Individual Rationality (IR).

The paper's main contributions are:

1. **Single-Item Interim IR:** The authors show that while WPB remains optimal for i.i.d. valuations under interim IR, this optimality breaks down in asymmetric settings. They characterize the variance-minimizing payment rule for asymmetric cases using a decomposition technique and prove existence using fixed-point theorems in Banach spaces.
2. **Multi-Item Interim IR:** For symmetric regular environments, the authors construct a mechanism (derived via a differential equation) that achieves zero revenue variance while maintaining non-negative payments and interim IR.
3. **Multi-Item Ex-post IR:** The authors demonstrate via a counterexample that WPB is not variance-minimizing in multi-item settings (specifically 2 items, 3 bidders). However, they prove that the discriminatory WPB auction still yields lower variance than the uniform $(k+1)$ -st price auction without requiring log-concavity assumptions.

General Comments

Examples of AI feedback

2. Numerical vs. Analytical Rigor in Section 5

While Sections 3 and 4 rely on rigorous analytical derivations (decomposition techniques, ODEs), Section 5 relies heavily on numerical examples (e.g., the counterexample in Section 5.1 and the discretized optimization in 5.2).

- In Section 5.1, the variance reduction is shown via numerical integration (≈ 0.26431 vs ≈ 0.2613). While the example is illustrative, an analytical bound or a local perturbation argument would strengthen the claim that WPB is strictly suboptimal.
- In Section 5.2, the optimal mechanism is approximated via discretization. Given Lemma 5.1 reduces the problem to functions of order statistics, is it possible to derive the Euler-Lagrange equations for the calculus of variations problem defined in [582-586]? Even if no closed form exists, characterizing the properties of the solution analytically would be superior to a purely numerical plot.

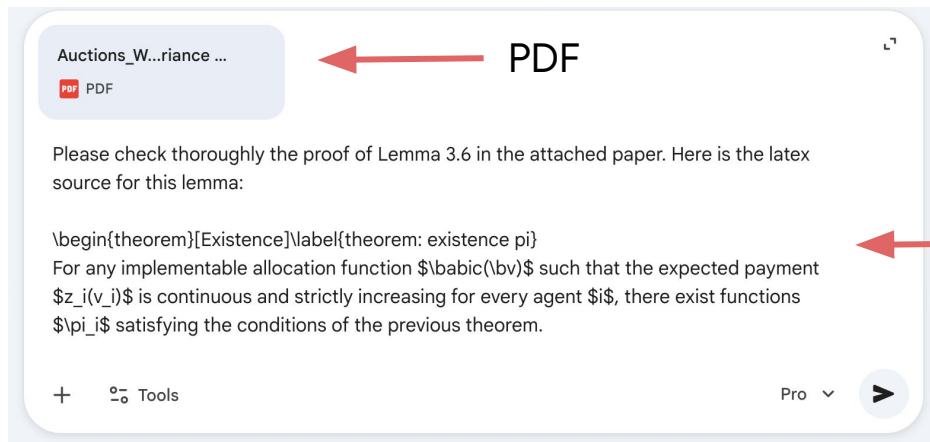
3. Theorem 3.6 (Existence via Fixed Point Theorems)

The proof of existence for the asymmetric variance-minimizing mechanism utilizes Schauder's fixed point theorem on the space of probability measures. While technically impressive, this machinery seems heavy for the problem at hand. Is the operator T a contraction under some metric? If so, the proof could be simplified, and potentially imply uniqueness or convergence of simple iterative algorithms. The authors should comment on the uniqueness of the solution $\pi_i(v_i)$, as Schauder does not guarantee it.

Minor Comments and Typos

- **General:** The citation style uses bracketed numbers [1], but the text often refers to "Waehrer et al. (1998)". Consistent formatting is recommended.
- **Page 13, Eq 306:** The text reads "...evaluates to evaluates to...". Duplicated words.
- **Page 13, Eq 307 & Page 14, Eq 319:** The use of $\approx$ without error bounds or exact expressions in an analytical section is slightly jarring.
- **Page 16, Footnote 1:** The footnote regarding jumps and countable discontinuities appears disconnected from the main text flow or is formatted essentially as a sentence fragment at the bottom of the page.
- **Page 20, Step 3:** The integral substitution uses $\frac{d}{dx}[x(1-F(x))]$. Ensure the sign conventions in the subsequent integration by parts are clear, particularly the negative sign on $(n-k)\phi(x)$.
- **Page 25, Eq 551:** The definition of $b_1(v_i)$ appears to have formatting issues in the submitted PDF (e.g., " $Vis U \leq 0.6$ "). The conditional cases are hard to parse.
- **Page 27, Figure Captions:** Figures 3, 4, and 5 have redundant or unclear labels (e.g., "#1 Highest Bidder", "Highest Bidder"). The axis labels are too small to be legible.
- **References:** Ensure all arXiv/SSRN preprints (e.g., Pekeč and Smilgins) are updated to the latest versions if published.

Keep in mind the “thinking budget”



From chatbot mode, but also true for agentic mode

- The model has limited “attention” and time spent thinking
- It is useful to make the model focus on a specific part of the paper
- PDF is fine, but LaTeX source is even better since there are no errors introduced by PDF parsing

Adjusting notation

- Adjusting notation can be usually achieved using search and replace, but this can be error prone and time consuming (esp. for math).

Ask model for a plan first

Algorithmic Game Theory

Spring 2021, Week 5

Mechanism Design Basics

MA434 @ LSE

Paul Dütting

While game theory analyzes strategic interaction in a given situation (“a game”), mechanism design takes an engineering perspective. Here a mechanism designer—a government, a company, an individual—seeks to design the rules of the game (“a mechanism”) so that strategic players select a desirable outcome at equilibrium. What constitutes a desirable outcome can differ, and will generally depend on the situation.

We will focus on mechanism design with money, where monetary transfers serve to align the interests of the mechanism designer with those of the players participating in the mechanism. Today we will consider the basics of designing so-called truthful mechanisms, in which it is in each player’s best interest to report truthfully. For concreteness we will phrase things in terms of allocation problems, and discuss so-called characterization results, which tell us which allocation rules we can hope to implement in a truthful mechanism.

We will distinguish between settings where the preferences of an individual player can be summarized by a single number unknown to the designer, and settings where the private information is multi-dimensional.

1 Warm-Up: Single-Item Auctions

We start by illustrating the basic ideas and difficulties with an example. Both concern the allocation of a single item to n players, which have a private valuation $v_i \geq 0$ for the item.

Can you check out Lecture 5 [T mdbasics.tex](#). In this lecture, in the mathematical formulas I am using x , x_i , X etc to refer to allocation decisions. I would rather want to use y , y_i , Y etc. Can you propose a plan of substitutions that we should apply to achieve exactly that. Please do not touch the code itself yet.

Worked for 1m >

Here is a proposed plan for substituting the allocation variables x , x_i , X with y , y_i , Y throughout your lecture notes in [mdbasics.tex](#).

Since variables like x and y are commonly used in mathematics, simply doing a find-and-replace will likely cause unintended side effects (such as renaming axes in TikZ or clashing with other variables). We should apply substitutions in a targeted manner:

1. Sets and Vectors

We will replace the feasible set X and the allocation vector x .

- X to Y : e.g., $X \subseteq \mathbb{R}_{\geq 0}^n$, $X = \{(x_1, \dots, x_n) \mid \dots\}$, $x(b) \in X$, and $\max_{x \in X}$.
- x to y : e.g., $x = (x_1, \dots, x_n)$, allocation x , and vector x .

2. Components / Individual Allocations

We will replace the allocation components x_i and x_j assigned to bidders.

- x_i to y_i : e.g., $x_i \geq 0$, $x_i \in \{0, 1\}$, $\sum x_i$, and $v_i \cdot x_i$.
- x_j to y_j : e.g., $\sum_{j \neq i} x_j b_j$.

... check and ask to execute

Creating figures

(or: Tikz figures are nice, but hard to generate by hand)

Only
lecture
with
.png
figures

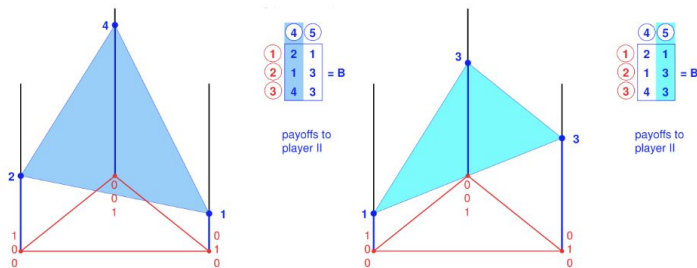
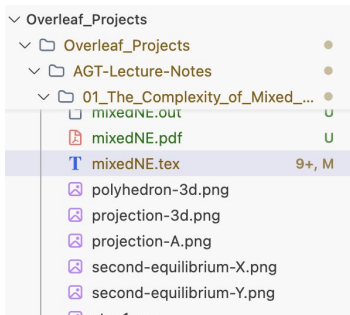


Figure 6: First step in the construction of upper envelope

Ask the model for help

In `T mixedNE.tex` can you help me turn Figure 6 ("First Step in The Construction...)" into a Tikz picture? Please try to match the style of the PNG as closely as possible.

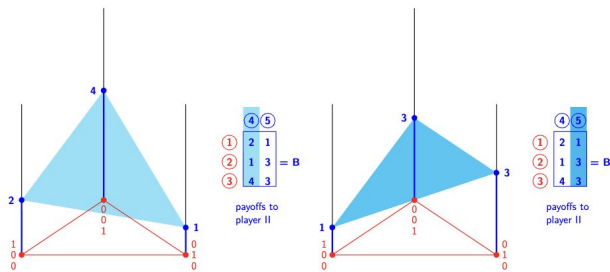
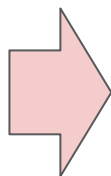


Figure 6: First step in the construction of upper envelope

This looks excellent, except that the poles in the back are a bit too long, I think it would suffice if they would go to height 5 or so instead/



Ask the
model to
refine

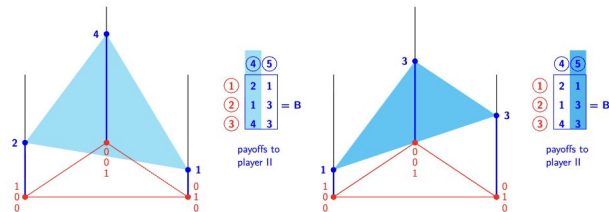
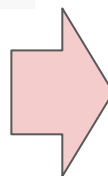
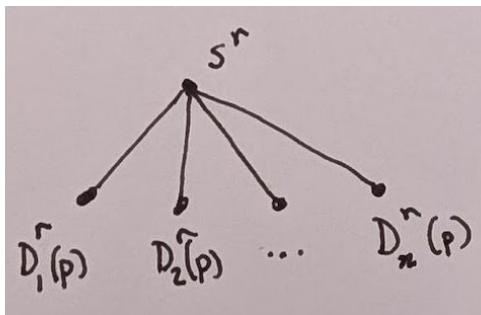


Figure 6: First step in the construction of upper envelope

Creating figures

- You can also hand-draw a figure, then ask the model to turn it into Tikz.



Overleaf_Projects



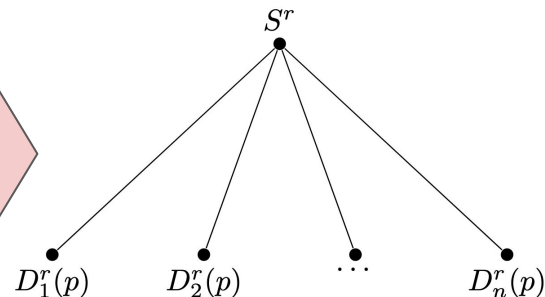
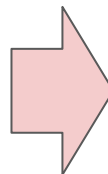
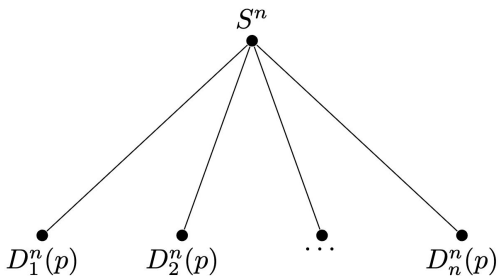
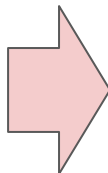
Please create a Tikz version of the attached image as a standalone compilable Latex file, place that file in the folder /Photo_to_Tikz/.



Gemini 3.1 Pro



Add media



Better: use @tree.tex



```
T tree.tex x
Photo_to_Tikz > T tree.tex
1 \documentclass{standalone}
2 \usepackage{tikz}
3 \usepackage{amsmath}
4
5 \begin{document}
6 \begin{tikzpicture}[
7   level distance=2.5cm,
8   sibling distance=1.8cm,
9   dot/.style={circle, fill=black, inner sep=1.5pt},
10  every label/.style={fill=none, inner sep=2pt}
11 ]
12
13 \node[dot, label=above:{$S^n$}] {}
14   child {node[dot, label=below:{$D_1^n(p)$}] {}}
15   child {node[dot, label=below:{$D_2^n(p)$}] {}}
16   child {node[dot, label=below:{$\dots$}] {}}
17   child {node[dot, label=below:{$D_n^n(p)$}] {}};
18
19 \end{tikzpicture}
20 \end{document}
21 %! for Agent
```

In tree.tex in lines 13-17 the superscript ^n should be superscript ^r. Can you fix and recompile?

Worked for 34s >

I have updated lines 13-17 in T tree.tex to use the superscript ^r instead of ^n, and I have successfully recompiled the document. The resulting tree.pdf has been generated. Let me know if there's anything else you need!

1 file changed +4 -4 > Review

Talking to a paper

- You can ask the model to explain certain steps of a proof (e.g. why a certain theorem can be applied, or which concentration bound is used)

Lemma 6.31 (expected revenue = expected virtual welfare). *Consider any DSIC mechanism $M = (x, p)$ for a single-parameter environment, then its expected revenue equals the expected virtual welfare*

$$\mathbb{E}_{v \sim F} \left[\sum_{i=1}^n p_i(v) \right] = \mathbb{E}_{v \sim F} \left[\sum_{i=1}^n \phi_i(v_i) \cdot x_i(v) \right].$$

Proof of Lemma 6.31. For the proof we will consider a DSIC mechanism $M = (x, p)$, and assume—for simplicity—that x is differentiable. All arguments can be made to work with some care to work for arbitrary monotone allocation rules, including piecewise constant rules.

[...]

Taking expectation over v_i , we obtain

$$\mathbb{E}_{v_i \sim F_i} [p_i(v_i, b_{-i})] = \int_{v_i=0}^{v_{\max}} f_i(v_i) \cdot p_i(v_i, b_{-i}) dv_i = \int_{v_i=0}^{v_{\max}} f_i(v_i) \int_{t=0}^{v_i} t \cdot x'(t, b_{-i}) dt dv_i.$$

We can now switch the order of integration (using Fubini's theorem as justification for why this is ok). We thus obtain

Ask questions

In `T optmd.tex` in the proof of Lemma 6.31 can you remind me what Fubini's theorem says and why it can be applied here.

Worked for 25s >

Fubini's Theorem (often used in conjunction with Tonelli's Theorem) provides the conditions under which you can swap the order of integration in a multiple integral.

In its most basic form, it states that if you have a function of two variables $h(x, y)$, and if the function is **absolutely integrable** (meaning the integral of $|h(x, y)|$ over the domain is finite), or if the function is everywhere **non-negative**, then the double integral can be evaluated by iterating the single integrals in any order:

$$\iint h(x, y) \, dx, dy = \int \left(\int h(x, y) \, dx \right) dy = \int \left(\int h(x, y) \, dy \right) dx$$

Why it applies in Lemma 6.31

In the proof of Lemma 6.31 in `T optmd.tex`, you are dealing with the following double integral over the variables v_{-i} and t :

$$\int_{v_{-i}=0}^{v_{-i}=\max} \int_{t=0}^{v_{-i}} f_{-i}(v_{-i}) \cdot t \cdot x'(t, b_{-i}) \, dt \, dv_{-i}$$

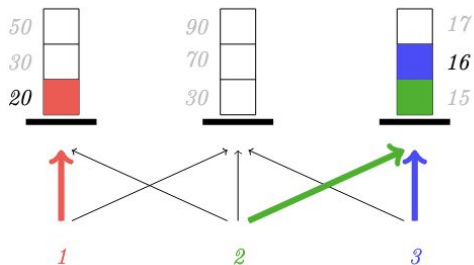
To apply Fubini's theorem (or more precisely, Tonelli's theorem for non-negative functions), we just need to verify that the integrand is **non-negative** over the region of integration. Let's look at the components:

Use `\label{thm:...}`, match notation, style ...

Proving theorems

- You can also ask the model to add proofs (esp. if these are known results)

Example 2.31 (Singleton Congestion Game). Consider a “server farm” with three servers a , b , c (resources) and three players 1,2,3 each of which wants to access a single server.



The colored arrows indicate a pure Nash equilibrium.

Theorem 2.32 ([Ackermann et al., 2008]). In a singleton congestion game with N players and M resources, all improvement sequences have length $O(N^2M)$.

Proof idea:

- Replace original delays by bounded integer values without changing the preferences of the players.
- Show an upper bound on the maximum potential with respect to new delays.
- Due to integer values, decrease of potential in an improvement step is at least 1. Hence, length of every improvement sequence is bounded by maximum potential.

Can you add a proof for Theorem 2.32 (`\label{thm:singleton-congestion-games}`) based on the proof idea sketched provided in the lecture notes.

Worked for 1m >

I have replaced the sketched proof idea with a formal proof block for Theorem 2.32 (singleton congestion games).

The added proof follows the three steps outlined in the lecture notes and uses the exact notation established earlier in the document (such as `\setR` for resources, Rosenthal's potential function notation `\Phi(s)`, and `N` and `M` for the number of players and resources).

You can review the changes in `T bestresponse.tex` around line 1220. Let me know if you would like to make any adjustments!

1 file changed +14 -9 > Review
+14 -9 bestresponse.tex ...ponse.tex

We will return to proving theorems in Part 3!

Typically a good idea to ask the model to double check

Copy-editing

Also useful for lecture notes (e.g., match style of my MA314 lecture notes)

Act as an academic copy editor for ACM EC. Edit the following text to improve grammar, flow, and idiomatic English. Constraints:

- * Idiomatic Phrasing: Fix any non-native phrasing or awkward sentence structures to ensure standard American academic English.
- * Clarity & Flow: Ensure the motivation transitions logically into the technical contribution.
- * Voice Preservation: Retain my original tone and terminology. Do not over-simplify the technical concepts.

+ 🛠️ Tools

Pro ▾

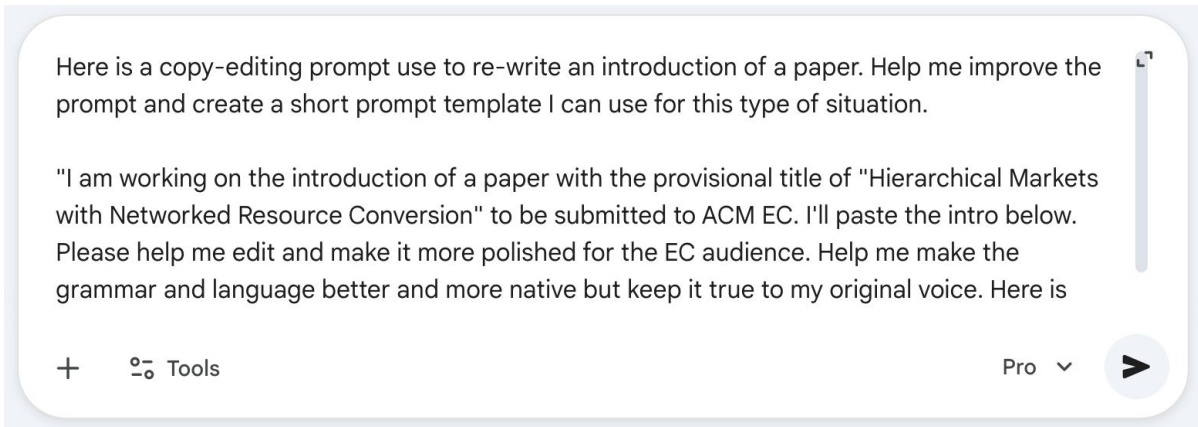


From chatbot mode, but also true for agentic mode

- Copy-editing works best with small section at a time
- Give the full context, conference, topics...
- Directions such as “keep my original voice”, “make it idiomatic”, ...
- “Keep the sentence structure relatively similar to the original unless a change significantly improves clarity”

Again, doing this with an agent has several advantages (e.g., iterate on edits before they are applied)

Prompt for prompting



The version of this in agentic workflows is to ask the agent to write a plan and then you comment on the plan (you can still do prompt for prompting though)



- If unsure what prompt to use, use Gemini (or your favorite chatbot) to suggest the prompt.
- It is useful to keep a list of “prompt templates” that worked for you in the past.

Implementing algorithms

Please review the CROCO algorithm described in this folder.

First, provide a complete Python implementation of the core algorithm. Provide a complete simulation script to plot the iteration-vs-loss curve across a few sample test cases. Run the simulation and produce a report.

+ Gemini 3.1 Pro (High) →

Python Implementation of CROCO

This plan outlines the steps to build, run, and report on the Log-Concave Density Algorithm for Corruption Robust Convex Optimization (CROCO).

User Review Required

IMPORTANT

The simulation will require Python libraries such as `numpy` and `matplotlib`. Please confirm if you have these installed or if I should set up a virtual environment and install them before running the script. I will use Metropolis-Adjusted Langevin Algorithm (MALA) for sampling the log-concave density to find the centroid at each step. The target function will be $f_t(x) = |x - x^*|^2$ with some deterministic corruption ϵ_t on a few rounds.

Proposed Changes

Python Simulation

[NEW] `croco_simulation.py`

A Python script containing:

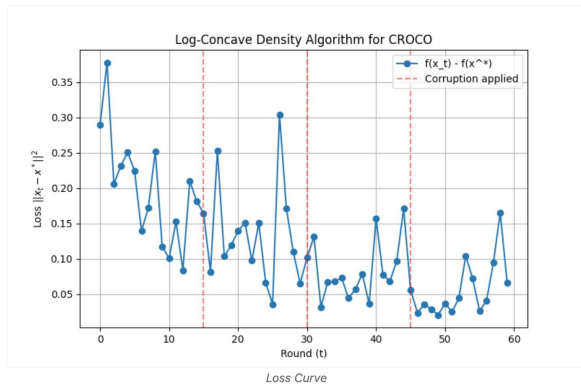
- `LogConcaveDensityAlgorithm`: A class that maintains the history of queries and implements MALA (Metropolis-Adjusted Langevin Algorithm) to sample from μ_t and approximate its centroid.
- `run_simulation()`: A function that runs the algorithm for ST rounds on a specified domain K (e.g., $[-2, 2]$ ball) using a fixed target x^* and corrupted oracle feedback.
- Plotting code to generate the "iteration-vs-loss" curve showing $f(x_t) - f(x^*)$ over time and saving it to `croco_loss_curve.png`.

Implementing algorithms

Simulation Setup

- Domain $\mathbb{S}$:** The unit ball in $\mathbb{R}^2$ ($SD = 2.0$).
- Target Minimizer x^* :** Set to $(0.5, 0.2)$.
- Target Function:** A simple quadratic $f_t(x) = \|x - x^*\|^2$. The gradient max norm within the unit ball yields $L = 4.0$.
- Adversarial Corruption:** At rounds $t \in \{15, 30, 45\}$, a strong artificial corruption $\epsilon_t = (-3.0, 3.0)$ is injected into the gradient to test the robustness of the Log-Concave Density updates. The adversarial vectors are projected back to $\mathbb{S}$ to strictly adhere to the bounded gradient assumption.

Results and Discussion



The iteration-vs-loss curve demonstrates that the algorithm effectively navigates toward the minimum x^* (minimizing $f(x_t) - f(x^*)$) despite the injected corruption.

- Initial Convergence:** The loss drops rapidly from roughly 0.35 down to around 0.15 within the first 15 rounds.
- Robustness to Corruption:** At rounds 15, 30, and 45 (highlighted by vertical dashed red lines), the algorithm receives heavily corrupted gradients. While the loss temporarily spikes slightly in some instances following a corruption due to the adversary pushing the density away from the truth, the soft log-concave density update ensures that the overall mass concentrated around x^* is not irreversibly lost.
- Recovery and Continued Optimization:** After the corruptions, the algorithm smoothly recovers, pushing the loss close to 0.02 - 0.04 by round 60.

Not quite there yet since the algorithm contains parameters that need to be tuned for the data:

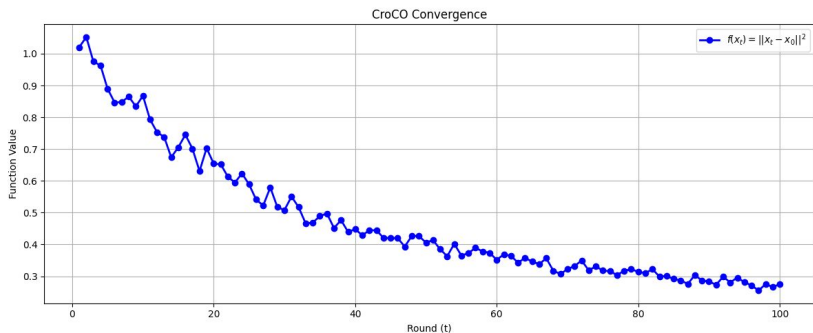
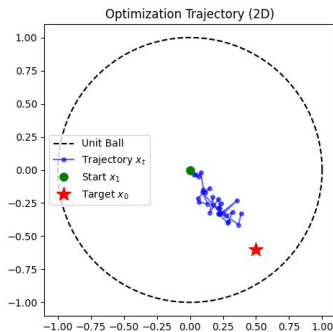
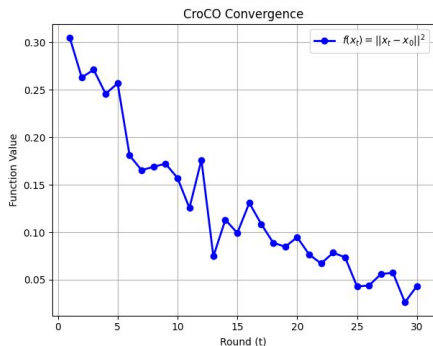
please tune the parameters of the MALA sampler and CROCO algorithm for better practical convergence. describe the tuning process and show the execution for the optimal choice of parameters

+ Gemini 3.1 Pro (High) ^

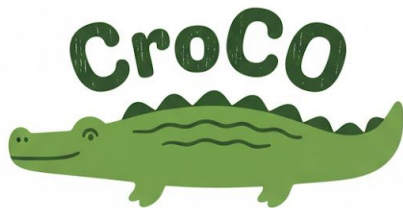


Implementing algorithms

- After a few rounds of interaction (e.g. asking AG to tune the parameters of the algorithm):



- Usually requires verifying the code, fixing mistakes, tweaking parameters, ...
- Some of that is manual and some of that you can ask the model to do using follow-up prompts.
- Bonus:** “Can you design a logo for the algorithm:”



Summarizing and rearranging content

- Agents are useful for summarizing content and reorganizing a paper
- Ask agent: "Make a plan to shorten the paper by 20%"
 - The agent will propose how to shorten different sections. You can provide feedback on where to focus
- Ask agent: "Fit the body in 9 pages"
 - In Latex, we only learn where content lands after compilation. Agents can (i) look at the compilation output to check the number of pages or (ii) open a web browser to view the PDF file with the internal viewer and then perform OCR to detect placement
- Ask agent: "Move all proofs to the appendix, adding references in the main body"

Further tips on using Antigravity

- **Use Mentions (@) to provide context:** The agent uses the current opened file for context, but as projects get large you can use “Mentions” or drag-and-drop to add files to the context.
- **Context pollution degrades performance:** As conversations get long, irrelevant information clutters the AI context window, degrading its reasoning and task focus. Create New Conversations (+) periodically.
- **Prompt for Prompting:** Especially for very long tasks, focus on getting best best possible initial prompt (iterate on the prompt using Antigravity itself). For example: “Help me craft a prompt to do X”
- **Give examples:** Very useful to point Antigravity to something that is already done: “Read the files in this folder / read these notes for inspiration.”
- **Iterate on a design doc:** Ask Antigravity to write a design doc and explicitly ask it to review with you before starting to implement it: “Write a design doc and let’s review it together.” Then write comments in the design doc and iterate from there.

Further tips on using Antigravity

- **Let Antigravity run things for you:** Ask Antigravity to run programs for you. Very useful for experiments: “Run this in script for me and babysit the run. If it crashes, fix the errors and re-run.” The keyword **babysit** is very useful.
- **Run parameters sweeps.** To tune algorithms or find the best set of parameters, you can ask Antigravity “Run a sweep for parameters a and b. Produce a table comparing results.”
- **Don't be afraid to ask Antigravity to figure things out:** If you don't know something ask Antigravity to figure out. Or better: “Figure out some options and let's discuss them.”

Part 3:

Theorem Proving with

Agentic AI

Two types of theorem proving

- **Type 1:** “I don’t know the proof of that statement but the right mathematician probably knows it.”

This is the domain where AI tends to be more useful as of today. However, guiding the AI to a rigorous proof may still require careful human steering.

- **Type 2:** It is truly an open problem for the research community.

AI is beginning to make progress here, but this is significantly harder. Success typically requires combining human intuition with structured AI workflows.

Brainstorm strategies

bilateral-trade ▾

In my paper <https://arxiv.org/abs/2602.12903> I left open the problem of designing a poly(d) regret for contextual bilateral-trade under one-bit feedback and per-period budget-balance. Please read the paper, understand the open problem and present to me a summary of the problem and three new ideas to attack it.

+ Gemini 3.1 Pro (High) ▾



Local ▾

Summary of the Open Problem

The Setting: You study a contextual bilateral trade problem where an online platform mediates between a seller and a buyer. At each time step t , the platform observes a d -dimensional context x_t . The seller and buyer have private valuations $s_t = \langle s, x_t \rangle$ and $b_t = \langle b, x_t \rangle$, parameterized by unknown d -dimensional vectors s and b . The platform posts a price p_t to the seller and q_t to the buyer.

The Constraints:

1. **One-bit feedback:** The platform only observes whether the trade occurs (i.e., both agents accept, meaning $s_t \leq p_t$ and $b_t \geq q_t$) or fails. It does not observe individual rejections.
2. **Per-period budget-balance:** The platform can never run a deficit on any round, meaning it must always post $p_t \leq q_t$.

The Issue: Under two-bit feedback (or one-bit feedback with small budget violations), you achieve $O(d \log d)$ regret by posting "safe prices" to disambiguate whose valuation caused a rejection. However, under strict one-bit feedback and per-period budget-balance, a "no-trade" outcome is inherently ambiguous. To circumvent this, your current algorithm relies only on successful trades to update its confidence regions. In the "strong overlap" case, it posts a single uniform random price $p_t = q_t$ over the union of the uncertainty intervals. Because the algorithm must guess blindly, the probability of hitting the "sweet spot" that guarantees a trade is very small, leading to a Steiner potential reduction of only $(1 - 4^{-d})$ upon a successful trade.

The Open Problem: This random sampling causes the regret to scale exponentially with the dimension $O(d6^d)$ for efficiency maximization, and $O(d6^d \log T)$ for profit maximization. The open problem is to design an algorithm that achieves poly(d) regret in this strict setting.

Three New Ideas to Attack the Problem

To achieve a poly(d) regret, an algorithm must extract information efficiently even from "no-trade" outcomes, or radically change how it probes the uncertainty space. Here are three novel ideas to explore:

1. Joint-Space Volume Cutting via Centroid/Ellipsoid Methods

Currently, your algorithm maintains independent confidence regions S_t and B_t in $\mathbb{R}^d$. A "no-trade" outcome gives the feedback: $\neg(\langle s, x_t \rangle \leq p_t \wedge \langle b, x_t \rangle \geq q_t)$. This translates to removing a convex wedge from the **joint parameter space** $S \times B \subseteq \mathbb{R}^{2d}$.

- **The Idea:** Instead of updating S_t and B_t independently (which requires unambiguous feedback), maintain a joint uncertainty set (or an enclosing volumetric barrier/ellipsoid) in $\mathbb{R}^{2d}$. By using cutting-plane methods (e.g., querying at the volumetric center or analytic center of the joint space), you can guarantee that *regardless* of whether a trade happens or fails, a constant fraction of the joint volume is eliminated.
- **Why it yields poly(d):** Standard geometric arguments (like Grunbaum's Theorem for centroids) guarantee that any cut through the centroid removes at least a $1/e$ fraction of the volume. This would ensure the joint volume shrinks by a constant factor every step, entirely bypassing the exponential waiting time for a successful uniform random probe.

2. Self-Funded Disambiguation (Dynamic Budget Buffers)

You established that violating the budget by an overall $O(d \log d)$ amount is sufficient to recover poly(d) regret via "safe prices" that disambiguate feedback.

- **The Idea:** Maintain strict per-period budget balance ($p_t \leq q_t$) most of the time, but maintain a running ledger of the **profit surplus** accumulated from the "well-separated" and "weak overlap" cases. In the problematic "strong overlap" case, instead of guessing uniformly, dynamically allow a local budget inversion ($p_t > q_t$) to act as a definitive disambiguating probe—but **strictly cap this inversion by the surplus profit generated in previous rounds**.
- **Why it yields poly(d):** As long as the algorithm only spends profit it has already materialized, the *actual* realized profit per round never dips below zero cumulatively, and the platform never requires external subsidies. This provides the algorithmic power of budget-violation (yielding poly(d) regret) while arguably satisfying the economic spirit of a self-sustaining, non-subsidized platform.

3. Optimism in the Face of Ambiguity with Relaxed Steiner Potentials

The core geometric bottleneck is that updating non-convex sets (resulting from the union of half-spaces in a "no-trade" event) breaks the Steiner potential analysis, which relies on

3. Optimism in the Face of Ambiguity with Relaxed Steiner Potentials

The core geometric bottleneck is that updating non-convex sets (resulting from the union of half-spaces in a "no-trade" event) breaks the Steiner potential analysis, which relies on convex bodies.

- **The Idea:** Define a "Smoothed" or "Relaxed" Steiner potential over non-convex regions, for example by measuring the Steiner volume of the **convex hull** of the surviving hypotheses, or by using log-concave sampling (like the hit-and-run algorithm) to maintain a probability measure over the surviving non-convex set.
- **The Execution:** Use an Optimism-in-Face-of-Uncertainty (OFU) approach. Pick the $(\tilde{s}, \tilde{b})$ in the surviving set that maximizes the potential gain from trade, and post the budget-balanced price $p_t = q_t = \frac{\langle \tilde{s}, x_t \rangle + \langle \tilde{b}, x_t \rangle}{2}$. If the trade fails, we immediately invalidate this highly optimistic region. Because the distribution over the version space is tracked probabilistically rather than with strict interval intersections, we can prove that the Gaussian/Steiner measure of the convex hull drops by a polynomial fraction $1/\text{poly}(d)$ every failure, rather than $(1 - 4^{-d})$.

Problem solving idea: Ralph Loop

Strategy #1: **Ralph Loop**: keep trying until you succeed

I think direction 3 "Optimism in the Face of Ambiguity with Relaxed Steiner Potentials" is promising. Please try to prove it. Follow those steps:

- 1) write an algorithm and proof
 - 2) have a different agent verify the proof and write a review and either accept or reject it
 - 3) if the proof is accepted, produce a publication-quality tex file
 - 4) if the proof is rejected, go back to step 1 and using the feedback try to solve it again
- Don't stop until you find a solution. Don't hesitate!

+ Gemini 3.1 Pro (High) ^



er <https://arxiv.org/abs/2602.12903> I left open the problem of designing a gret for contextual bilateral-trade under one-bit feedback and per-period alance. Please read the paper, understand the open problem and present to me y of the problem and three new ideas to attack it.

ection 3 "Optimism in the Face of Ambiguity with Relaxed Steiner Potentials" is . Please try to prove it. Follow those steps:
algorithm and proof
different agent verify the proof and write a review and either accept or reject it

Worked for 6m >

I have written the initial proof for the "Optimism in the Face of Ambiguity with Relaxed Steiner Potentials" idea and launched an expert math subagent to review it rigorously, as requested. The subagent is currently verifying the geometric lemma regarding the drop in the first intrinsic volume and will provide feedback or a mathematical fix shortly. I will wait for its review and iterate until the proof is solid.

Proof Draft

Drafting the proof of poly(d) regret to send to the math expert subagent

1 subagent running

Math Reviewer

Ask anything, @ to mention, / for actions

+ Gemini 3.1 Pro (High) ^

Proof Draft

Proof of poly(d) Regret for Contextual Bilateral Trade

Algorithm: OFU with Relaxed Steiner Potentials

1. Initialize the convex version space $K_1 = B_1^d \times B_1^d \subset \mathbb{R}^{2d}$.
2. At each step t , receive context $x_t \in \mathbb{R}^d$.
3. Compute the optimistic bounds: $\tilde{s}_t = \min_{(s,b) \in K_t} s^\top x_t$, $\tilde{b}_t = \max_{(s,b) \in K_t} b^\top x_t$. Let $W_t = \tilde{b}_t - \tilde{s}_t$. If $W_t \leq 0$, post $p_t = 1$ (no trade, regret 0).
4. Else, post the budget-balanced price $p_t = \frac{\tilde{s}_t + \tilde{b}_t}{2}$.
5. Observe feedback $y_t \in \{0, 1\}$.
 - If $y_t = 1$ (trade occurs): $K_{t+1} = K_t \cap \{s^\top x_t \leq p_t\} \cap \{b^\top x_t \geq p_t\}$.
 - If $y_t = 0$ (no trade occurs): $K_{t+1} = \text{Conv}(K_t \setminus \{s^\top x_t \leq p_t \wedge b^\top x_t \geq p_t\})$.

Regret Analysis: Let r_t be the regret. If $y_t = 1$, the platform gains $b^{*\top} x_t - s^{*\top} x_t$, matching the benchmark, so $r_t = 0$. If $y_t = 0$, the platform gains 0. The benchmark gain is at most W_t . Thus $r_t \leq W_t$. We only suffer regret on no-trade steps.

Potential Function: We use the first intrinsic volume (mean width) $V_1(K_t)$ of the convex body K_t as the potential. $V_1(K_1) \leq O(d)$.

Lemma: If $y_t = 0$, the mean width drops by a polynomial fraction of W_t , i.e., $V_1(K_t) - V_1(K_{t+1}) \geq \Omega(\frac{W_t}{d^{3/2}})$. *Proof Outline:* The corner removed, $C_t = \{s^\top x_t \leq p_t \wedge b^\top x_t \geq p_t\}$, corresponds to a deep cut of depth $W_t/2$ relative to the bounding box of K_t in the projection on x_t . Because K_{t+1} is the convex hull of the remaining points, removing this corner truncates the extreme points in the joint direction, reducing the mean width by at least $\Omega(W_t/d^{3/2})$ by standard properties of intrinsic volumes of convex bodies. Summing over all steps where $y_t = 0$, the total regret is $\sum W_t \leq O(d^{3/2})V_1(K_1) \leq O(d^{5/2})$, which is $\text{poly}(d)$.

Problem solving idea: Ralph Loop

Review

Review of Proof Draft

Flaw in the Original Proof

The proof draft relies on the claim that removing the "no-trade" corner $C_t = \{s^\top x_t \leq p_t \wedge b^\top x_t \geq p_t\}$ and taking the convex hull of the remaining set $R_t = K_t \setminus C_t$ strictly reduces the mean width V_1 by a polynomial factor.

This claim is **mathematically false**. The corner C_t is the intersection of the convex body with a quadrant. The boundary of this quadrant passes through the interior of K_t , and the extreme points of C_t lie exactly on the boundary of the cut. Because these extreme points are in the closure of R_t , taking the convex hull of R_t perfectly reconstructs them. **Counterexample:** Let K_t be the L_1 diamond $\{|u| + |v| \leq 1\}$ and $p_t = 0$. The removed corner $C_t = \{u \leq 0 \wedge v \geq 0\}$ is the top-left triangle with vertices $(0, 0)$, $(0, 1)$, $(-1, 0)$. The remaining set R_t contains points arbitrarily close to $(0, 1)$ and $(-1, 0)$. Thus, taking the convex hull of R_t reconstructs the entire segment between $(-1, 0)$ and $(0, 1)$, restoring the exact triangle C_t . Therefore, $\text{Conv}(K_t \setminus C_t) = K_t$, and the volume (and mean width) drop is exactly zero. The algorithm will get stuck infinitely proposing the same prices.

Conclusion: I reject the original proof.

Rigorous Mathematical Fix: Decoupled Bandit Ellipsoid Method

To achieve $\text{poly}(d)$ regret, we abandon the single convex hull and decouple the learning of the seller and buyer functions using the budget balance slack. We maintain two separate ellipsoids E_s and E_b containing s^* and b^* .

At each round t , we compute the ranges $[s_t, \bar{s}_t]$ and $[\bar{b}_t, b_t]$ of $s^\top x_t$ and $b^\top x_t$ over E_s and E_b . Let $\hat{s}_t$ and $\hat{b}_t$ be the centers of these intervals (which correspond to the projections of the ellipsoid centers). Let $W_s = \bar{s}_t - s_t$ and $W_b = \bar{b}_t - b_t$ be the widths. We proceed with the following strategy:

1. **Central Cut for Seller:** If $\hat{s}_t \leq b_t$, we post $p_t = \hat{s}_t$ and $q_t = b_t$. Budget balance $p_t \leq q_t$ holds.

- Feedback $y_t \in \{0, 1\}$ provides a guaranteed central halfspace cut for s^* (either $s^\top x_t \leq \hat{s}_t$ or $> \hat{s}_t$). We update E_s via the standard Ellipsoid method. Volume of E_s drops by a constant factor, and regret is bounded by 1.

I think direction 3 "Optimism in the Face of Ambiguity with Relaxed Steiner Potentials" is promising. Please try to prove it. Follow those steps:

- 1) write an algorithm and proof
- 2) have a different agent verify the proof and write a review and either accept or reject it

Worked for 6m

I have written the initial proof for the "Optimism in the Face of Ambiguity with Relaxed Steiner Potentials" idea and launched an expert math subagent to review it rigorously, as requested. The subagent is currently verifying the geometric lemma regarding the drop in the first intrinsic volume and will provide feedback or a mathematical fix shortly. I will wait for its review and iterate until the proof is solid.

Message from Math Reviewer (math_expert)

Worked for 2m

I have rewritten the proof incorporating the Decoupled Bandit Ellipsoid Method and corrected a subtle bug in the first review that would have allowed $O(T)$ regret by proving a new geometric lemma (the "empty gap" lemma). The subagent is currently verifying this revised proof. Once it confirms and formally accepts the proof, I will proceed to step 3 and generate the publication-quality LaTeX file.

Proof Draft

Drafting the proof of $\text{poly}(d)$ regret to send to the math expert subagent

Proof Draft V2

Writing the revised proof for the math expert to review

1 subagent running

Math Reviewer

Proof Draft V2

Decoupled Bandit Ellipsoid Method

Algorithm:

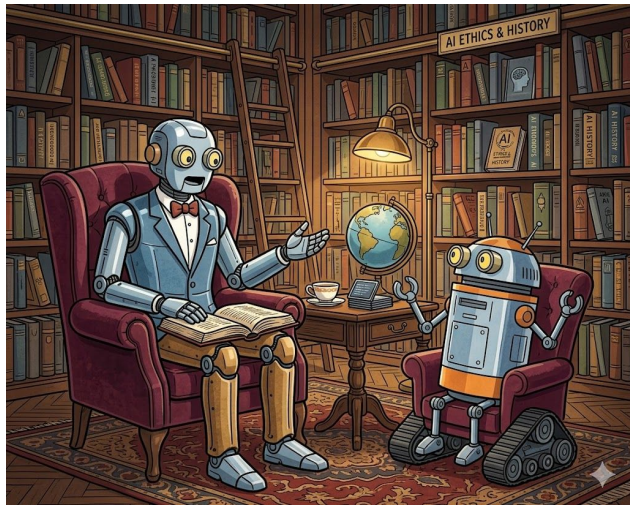
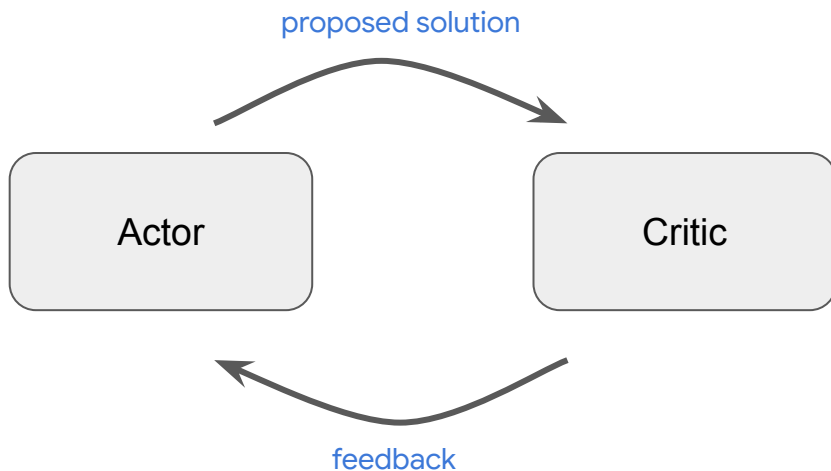
1. Maintain two ellipsoids E_s and E_b in $\mathbb{R}^d$.
2. For each context x_t , compute the intervals $[s_t, \bar{s}_t]$ and $[\bar{b}_t, b_t]$ of $s^\top x_t$ and $b^\top x_t$ over E_s and E_b . Let $\hat{s}_t, \hat{b}_t$ be the centers and W_s, W_b be the widths.
3. **Case 1 (Central Cut for Seller):** If $\hat{s}_t \leq b_t$, post $p_t = \hat{s}_t, q_t = b_t$. Budget balance holds ($p_t \leq q_t$). q_t is safe for the buyer, so $y_t \in \{0, 1\}$ provides a central halfspace cut for E_s .
4. **Case 2 (Central Cut for Buyer):** If $\bar{s}_t \leq \bar{b}_t$, post $p_t = \bar{s}_t, q_t = \bar{b}_t$. p_t is safe for the seller, so $y_t \in \{0, 1\}$ provides a central cut for E_b .
5. **Case 3 (Randomized Price):** If both fail, $\hat{s}_t > b_t$ and $\bar{s}_t > \bar{b}_t$. Post $p_t = q_t \sim \text{Unif}[\bar{s}_t, \bar{b}_t]$. If $y_t = 0$, do not update. If $y_t = 1$, update both E_s (with $s^\top x_t \leq p_t$) and E_b (with $b^\top x_t \geq p_t$).

Proof of $\text{poly}(d)$ Regret: In Cases 1 and 2, one ellipsoid is cut centrally, reducing its volume by a constant factor $1 - \Omega(1/d)$. Regret is at most 2. This can happen $O(d^2 \log(1/\epsilon))$ times. In Case 3, we prove that *any* trade guarantees a deep cut. **Lemma:** Every price p_t is a deep cut for E_s (depth $\geq W_s/4$) or for E_b (depth $\geq W_b/4$). *Proof:* Suppose p_t is shallow for both: $p_t > \hat{s}_t + W_s/4$ and $p_t < \hat{b}_t - W_b/4$. This implies $\hat{b}_t - \hat{s}_t > (W_s + W_b)/4$. However, failure of Cases 1 and 2 means $\hat{b}_t - \hat{s}_t < W_b/2$ and $\bar{b}_t - \bar{s}_t < W_s/2$. Thus $\hat{b}_t - \hat{s}_t < \min(W_s/2, W_b/2) \leq (W_s + W_b)/4$. Contradiction. The gap of shallow cuts is empty.

Since every trade guarantees a deep cut of depth $\geq W/4$, whenever $y_t = 1$, the volume of at least one ellipsoid shrinks by $1 - \Omega(1/d)$. Let $\Phi = \log \text{Vol}(E_s) + \log \text{Vol}(E_b)$. Expected potential drop: $\mathbb{E}[\Delta\Phi] \geq \mathbb{P}(y_t = 1) \cdot \Omega(1/d)$. Trade probability: $\mathbb{P}(y_t = 1) = \frac{\Delta_t}{b_t - s_t} \geq \frac{\Delta_t}{W_s + W_b}$, where $\Delta_t = (b^* - s^*)$. Rearranging: $\Delta_t \leq (W_s + W_b) \cdot O(d) \cdot \mathbb{E}[\Delta\Phi]$. Summing over all Case 3 steps gives total expected regret $O(d \sum \mathbb{E}[\Delta\Phi]) = O(d^2 \log(1/\epsilon))$. Total regret is $\text{poly}(d)$ and independent of T .

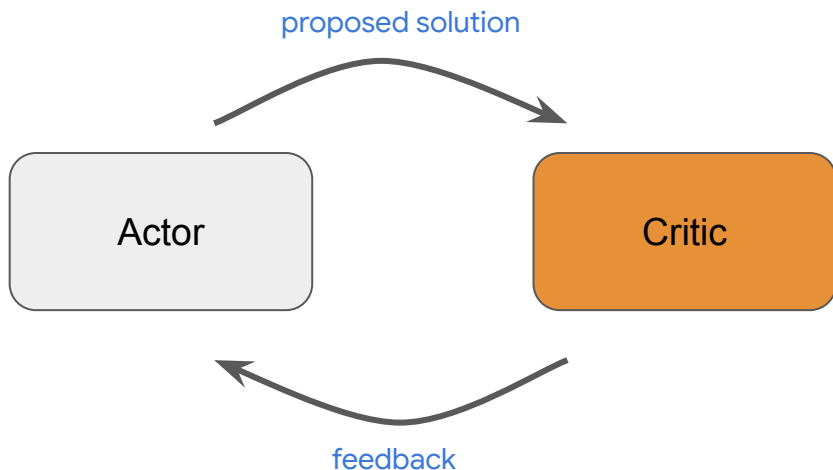
LLM call patterns: Actor-Critic / Self Reflection

- With chatbots: manually call the model to prove a statement, then call it to review the statement, then call it to fix the bugs, then call the verifiers ...
- Automate using: Actor-Critic loops / AI Debate / Self-Reflection



LLM call patterns: Actor-Critic / Self Reflection

- Before: manually call the model to prove a statement, then call it to review the statement, then call it to fix the bugs, then call the verifiers ...
- Automate using: Actor-Critic loops / AI Debate / Self-Reflection



This pattern is very useful when the Critic is something more structured:

- Autoformalizer like Lean
 - See afternoon workshop
- Heuristic Scoring functions
 - E.g. AlphaEvolve
- Other algorithms
 - Equilibrium solver (game theory)
 - Optimization solver
 - Controller
 - ...

Other LLM call patterns

- **Parallelize & Aggregate:** Call the model n times and then ask to choose the best solution or choose majority voting if binary/categorical recommendation
- **Plan & Execute:** Ask the model to propose a plan and then make one call to execute each of the steps of the plan
- **Least-to-Most:** Ask the model to propose a simpler/approachable subproblem, then a call to solve it. Then ask the next approachable subproblem, then a call to solve it...
- **Primal-Dual:** Two calls asking the model to prove and disprove a statement. Then a call to a judge to decide which direction to take.
- **Routing:** Create different prompts to attempt to solve the problem via different techniques (e.g. geometry, algebra, analysis, probability, ...) forcing the model to explore different directions. Then one call to get best possible directions.

Problem solving idea: Primal Dual

For the theorem below, start two separate agents, one trying to prove and one trying to disprove it. Each of them should produce a tex file with the conclusions. Then have a third agent to evaluate both and generate a final report.

Both the Prover and Disprover have finished their analyses!

- The **Prover** submitted a rigorous proof using Polyá's and Handelman's Positivstellensatz, concluding the theorem is true.
- The **Disprover** submitted a counterexample for $n \geq 2$, arguing there is an algebraic contradiction involving the lowest degree homogeneous part of the denominator polynomial, concluding the theorem is false.

I have just launched the **Evaluator** subagent. It is currently reading both `.tex` files. It will evaluate the merits and flaws of both arguments and produce a `final_report.tex` with a definitive verdict.

I'll let you know as soon as the final report is ready!

The Final Report:

The comprehensive final evaluation report has been written and saved as a LaTeX file. You can view the final verdict here: [T final_report.tex](#)

You can also check the individual conclusions from the Prover and Disprover:

- **Prover:** [T proof_conclusions.tex](#)

For the theorem below, start two separate agents, one trying to prove and one trying to disprove it. Each of them should produce a tex file with the conclusions. Then have a third agent to evaluate both and generate a final report.



Let $\mathbf{x} = (x_1, \dots, x_n) \in (0, 1)^n$. A **non-negative Bernstein monomial** is a function of the form:

```
\[
M_{\mathbf{b}, \mathbf{c}}(\mathbf{x}) = a \prod_{i=1}^n x_i^{b_i} (1-x_i)^{c_i}
\]
```

Worked for 47s ▾

Explored ▾

Thought for 19s >

Defined `math_prover` subagent >

Defined `math_disprover` subagent >

Defined `math_evaluator` subagent >

Thought for 10s >

 **Theorem Prover**

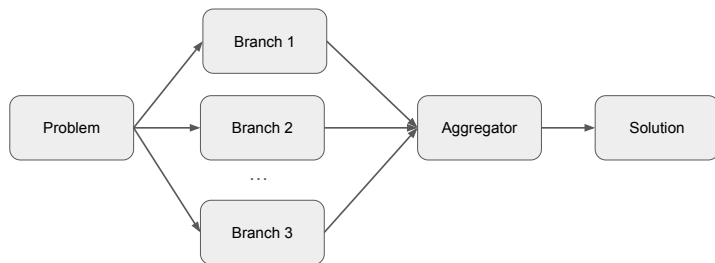
Invoked `math_prover` subagent

 **Theorem Disprover**

Invoked `math_disprover` subagent

Problem solving idea: Parallelize and Aggregate

For the theorem below, start five separate subagents attacking the problem from different directions.. Each of them should produce a tex file with the conclusions. Then have another agent evaluating them and reaching a final conclusion.



The 5 approaches are:

1. **Constructive Prover:** Attempting to construct the Bernstein polynomials directly from the properties of the binary tree.
2. **Counterexample Searcher:** Hunting for corner cases or simple Bernoulli-implementable functions that fail the representation.
3. **Literature Analyst:** Analyzing connections to existing literature on Bernoulli factories and representation of rational probabilities.
4. **Algebraic Methods Expert:** Approaching the theorem using generating functions and formal power series.
5. **Probabilistic Interpreter:** Focusing on the probabilistic meaning of the ratio $P(x)/Q(x)$ and reverse-engineering the tree.

I will wait for them to complete their tasks and report back. Once all five have finished generating their LaTeX files, I will invoke a final evaluator agent to review their work and reach a definitive conclusion.



5 subagents running

- ☐ Probabilistic Interpreter
- ☐ Algebraic Methods Expert
- ☐ Literature Analyst
- ☐ Counterexample Searcher
- ☐ Constructive Prover

Encouraging the model to think longer

- You can encourage the model to think longer:
 - “Don’t stop until you find a proof or a counter-example.”
 - “Provide an exhaustive response. Do not rush to a conclusion.”
 - “Take your time and explore all nuances ...”
 - “Don’t hesitate.”
- With those, you need to be more mindful of token usage.

Create your own skills, workflows and agents

- **Create skills / workflows:** After a non-trivial operation that will be useful in the future, you can ask Antigravity to create a skill / workflow and save it somewhere
 - Modern version of keeping a prompt library
 - Example: [Antigravity Arvix Skill](#)
- **Create your own agents / orchestration patterns:** You can brainstorm with Antigravity how to create the best pattern / orchestration to solve your problem. In general, take the time to discuss with the model:
 - What is the right prompt to use
 - Iterate on a plan
 - Decide what is the right orchestration strategy
- Then start... For difficult tasks, if well specified, Antigravity can run for hours autonomously.

How to use existing workflows....

- For example, consider the submissions to FirstProof Second Batch:
<https://1stproof.org/assets/docs/report.pdf>

1stproof / batch-2 Public

<> Code

Issues

Pull requests

Actions

Projects

Security and quality

Insights

main

1 Branch

0 Tags

Go to file

<> Code

abouzaid-1stproof raw-outputs: batch-2-raw-outputs/IMProofBench 77b4e4b · 3 weeks ago 9 Commits

batch-2-AI-solutions batch-2: solutions, reviews, design, submissions, model-t... 3 weeks ago

batch-2-design batch-2: solutions, reviews, design, submissions, model-t... 3 weeks ago

batch-2-human-solution batch-2: solutions, reviews, design, submissions, model-t... 3 weeks ago

batch-2-modeltesting batch-2: solutions, reviews, design, submissions, model-t... 3 weeks ago

batch-2-raw-outputs raw-outputs: batch-2-raw-outputs/IMProofBench 3 weeks ago

batch-2-reviews batch-2: solutions, reviews, design, submissions, model-t... 3 weeks ago

batch-2-submissions batch-2: solutions, reviews, design, submissions, model-t... 3 weeks ago

.gitignore batch-2: solutions, reviews, design, submissions, model-t... 3 weeks ago

First Proof Second Batch

June 10, 2026

Mohammed Abouzaid¹
Stanford University

Nikhil Srivastava²
University of California, Berkeley

Rachel Ward³
University of Texas at Austin

Lauren Williams⁴
Harvard University

	System A	System B	System C	System D
P1				
P2				
P3				
P4				
P5				
P6	—			
P7				
P8				
P9				
P10				

Table 3: Referee recommendations per submission: essentially flawless, minor revisions, major revisions, reject. Minor-revision pills are split diagonally to show the basis for the revision: the mathematics is correct but the citations are inadequate and/or incorrect, the reviewer flags an actual mathematical error or gap that must be corrected (whether in their summary or in their inline review comments). Each cell is a link to the corresponding editor summary of the referee reports for that problem. The entry for Problem 6, Submission A is blank because of a technical issue (see Section 5.6).

How to use existing workflows....

- Here is the code for the submitted workflows / harnesses (ETH & Aarhus group):
<https://github.com/1stproof/batch-2/tree/main/batch-2-submissions>

Name	Last commit message	Last commit date
 ..		
 improofbench	batch-2: solutions, reviews, design, submissions, model-testing	3 weeks ago
 princeton	batch-2: solutions, reviews, design, submissions, model-testing	3 weeks ago
 ucla	batch-2: solutions, reviews, design, submissions, model-testing	3 weeks ago
 .dockerignore	batch-2: solutions, reviews, design, submissions, model-testing	3 weeks ago
 .gitignore	batch-2: solutions, reviews, design, submissions, model-testing	3 weeks ago

- Let's find out what they do...

How to use existing workflows....

Please read the First Proof Second Batch report <https://1stproof.org/assets/docs/report.pdf> and then read the code for improofbench in <https://github.com/1stproof/batch-2/tree/main/batch-2-submissions/improofbench>. As a first step, generate a document describing the strategy taken by improofbench.

+ Gemini 3.5 Flash (High) ▾



Local ▾

Analysis of the IMProofBench (ProofCouncil) Strategy

This report analyzes the architecture, execution strategy, and benchmark performance of **IMProofBench** (also referred to as **ProofCouncil** or **mathagents** in the code and the report), which was submitted to the **First Proof Second Batch** mathematical research benchmark in June 2026.

1. System Overview

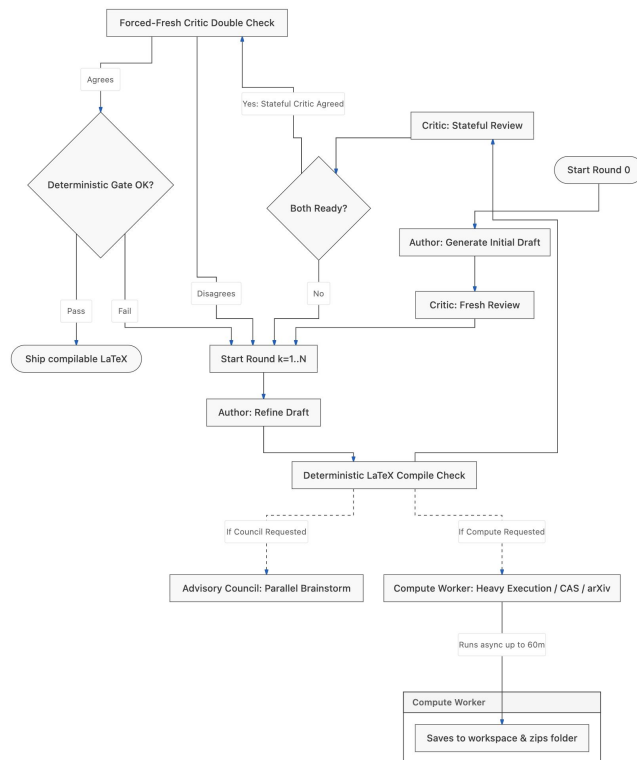
IMProofBench is built on **ProofStack**, an agentic mathematical research workflow framework. The core philosophy of the system is to model mathematical research as an **iterative, multi-agent collaboration** rather than a single-shot completion. It structures the process as an **Author/Critic loop** augmented by specialized computational and collaborative resources. The system is developed by a joint team from **ETH Zürich**, **Aarhus University**, and independent researchers.

Technology Stack

- **Base Models:** Primarily utilizes **GPT-5.5 Pro** for the core cognitive roles (Author and Critic).
- **Advisory Council Models:** GPT-5.5 Pro, Claude 4.7 Opus (Max), and Gemini 3.1 Pro.
- **Compute Worker Model:** GPT-5.5 (reasoning mode set to *xhigh*).
- **Tooling/Sandbox:** Custom sandboxed execution using Docker and Modal, featuring a complete Computer Algebra System (CAS) stack including **SageMath**, **GAP**, **Singular**, and **PARI/GP**.
- **Literature Search:** Semantic Scholar integration + GLM-based OCR for reading paper PDFs.

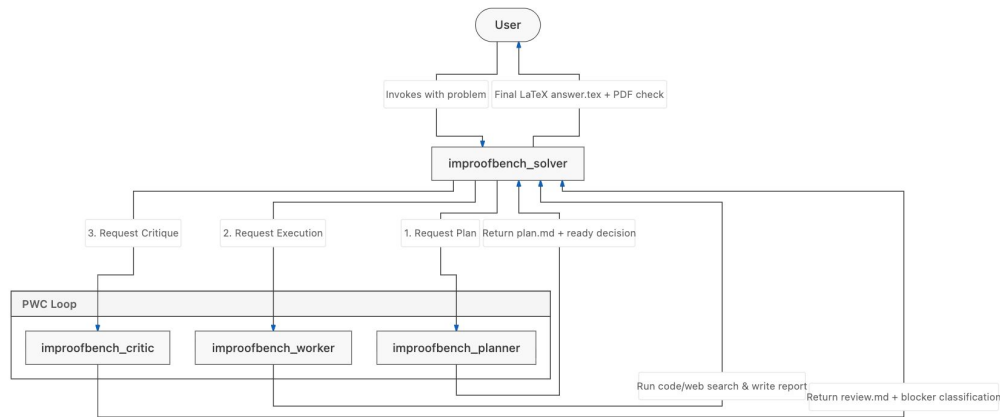
2. Core Architecture: The Author/Critic Loop

At the heart of the system is the **ACWorkflow** (`ac_workflow.py`), which orchestrates the generation, review, and refinement of a mathematical proof across multiple rounds (typically configured for 10 rounds, but capable of adapting up to 200 rounds within time and budget limits).



How to use existing workflows....

Prompt: Please read the FirstProof Second Batch report <https://1stproof.org/assets/docs/report.pdf> and the implementation of improofbench in <https://github.com/1stproof/batch-2/tree/main/batch-2-submissions/improofbench> and create a custom subagents that I can run in this session that have the same behavior of improofbench. Create separate subagents for planner, worker and critic. Use the prompt best practices there. Actually, do some online research to find prompt best practices for solving math problems and incorporate them here. Then show me how I can call it to solve a research problem in this session.



I have implemented and registered the custom subagents based on the approved implementation plan.

Summary of Accomplishments

1. **Planner** (`improofbench_planner`): Defined a subagent to analyze progress and generate concrete instruction steps for the worker.
2. **Worker** (`improofbench_worker`): Defined a subagent equipped with web search and file/command execution capabilities to verify lemmas via Python, query mathematical databases, and edit document files.
3. **Critic** (`improofbench_critic`): Defined a strict mathematical peer-review agent to identify logical gaps and classify them as blockers or concerns.
4. **Solver Orchestrator** (`improofbench_solver`): Defined the coordinator agent that manages the workspace, sequences the other subagents iteratively (up to 5 rounds), and verifies LaTeX compilation.
5. **Walkthrough Document**: Created a comprehensive guide here: [walkthrough.md](#).

How to Use in this Session

You can invoke the system by asking me to call `improofbench_solver` to solve a mathematical problem. For example:

"Please run the `improofbench_solver` subagent to solve: 'Prove that the square root of any prime number is irrational. Run 3 rounds and output the final answer.tex.'"

Walkthrough

Walkthrough summarizing the defined IMProofBench ProofCouncil subagents and illustrating how the user can invoke the solver to tackle mathematical research problems in this session.

How to use existing workflows....

```
efx.md x
efx.md > ## Existence of EFX Allocations for Additive Valuations > ### Research Goal
1  ## Existence of EFX Allocations for Additive Valuations
2
3  ### Problem Description
4
5  In the fair division of indivisible resources, there is a set  $M=\{1,\dots,m\}$ 
6  of goods and a set  $N=\{1,\dots,n\}$  of agents. Each agent  $i \in N$  has a
7  valuation function  $v_i: 2^M \rightarrow \mathbb{R}_{\geq 0}$ , typically assumed to be
8  monotone. A valuation is additive if  $v_i(S) = \sum_{j \in S} v_i(\{j\})$ 
9  for all  $S \subseteq M$ .
10
11  An allocation  $S=(A_1,A_2,\dots,A_n)$  is a partition of the items among the
12  agents. Because exact envy-freeness is generally unattainable with indivisible
13  items (e.g., allocating a single diamond to two agents), the literature focuses on
14  relaxations. The most promising candidate solution concept is Envy-Free up to Any
15  Good (EFX).
16
17  An allocation is EFX if no agent  $i$  envies any other agent  $j$  after the
18  hypothetical removal of any single good  $g$  from agent  $j$ 's bundle:
19
20  
$$v_i(A_i) \geq v_i(A_j \setminus \{g\}) \quad \text{for all } i, j \in N \quad \text{and} \quad \text{for all } g \in A_j.$$

21
22  ### Known Results
23
24  Positive Results:
25  - Exact EFX allocations are guaranteed to exist for any number of agents with
26  identical valuations, and for 3 agents with additive valuations (Chaudhury, Garg,
27  and Mehlhorn, 2024).
28  - For additive valuations, an exact EFX allocation exists that leaves at most  $n-2$ 
29  items unallocated to charity (Berger et al., 2022). The existence of  $(1-\epsilon)$ -EFX
30  allocations with  $O(n/\epsilon^2)$  charity for additive valuations has also been established
31  (Akrami et al., 2024).
32  - Simulated annealing finds exact EFX allocations for additive valuations with
33  random instances in experiments.
34
35  Impossibility Results:
36  - Akrami, Mayorov, Mehlhorn, Srinivas, and Weidenbach (2026) used SAT-solving to
37  construct a counterexample showing that exact EFX allocations do not always exist
38  for  $n \geq 3$  agents and  $m \geq n$  items with submodular (and general monotone)
39  valuations. These counterexamples explicitly rely on submodular/subadditive
40  complementarities and do not apply to additive valuations.
41  - Mackenzie and Suzuki (2026) provided a compact, human-verifiable counterexample
42  for 3 agents and 8 items refuting EFX existence for weighted coverage (submodular)
43  valuations, again not applicable to the additive case.
44
45  ### Research Goal
46
47  Do exact EFX allocations always exist for the canonical case of additive
48  valuations for  $n \geq 4$  agents? The case of 3 agents is resolved (Chaudhury,
49  Garg, and Mehlhorn, 2024), and for 3 types of agents is resolved as well. All known
50  counterexamples to EFX existence are for more general types of valuations
51  (submodular/subadditive), leaving the additive case as the central open question in
52  fair division.
```

Please read the research question in efx.md run the `improofbench_solver` to solve that problem for $n=4$. 10:53 AM

Worked for 22s >

I have read the research question from `efx.md` and successfully launched the `improofbench_solver` subagent to tackle the problem for $n = 4$ agents.

The orchestrator is now running in the background and coordinating the ProofCouncil loop (Planner, Worker, and Critic) for up to 5 rounds. It will construct the workspace, formulate plans, execute them, write the TeX files, review the proofs, and finally verify LaTeX compilation.

I am waiting for it to finish its work and send the results back to me. Once the solver completes, I will present the final proof (`answer.tex`), the critic's review, and the compilation check.

10:53 AM

1 subagent running

ProofCouncil Coordinator

Worked for 36s >

I have invoked the `improofbench_planner` subagent. Now I will wait for it to return the plan.

Message from Proof Planner (improofbench_planner)

Worked for 13s >

I have saved the plan and invoked the `improofbench_worker` subagent to execute it. I am now waiting for its report.

Message from Proof Worker (improofbench_worker)

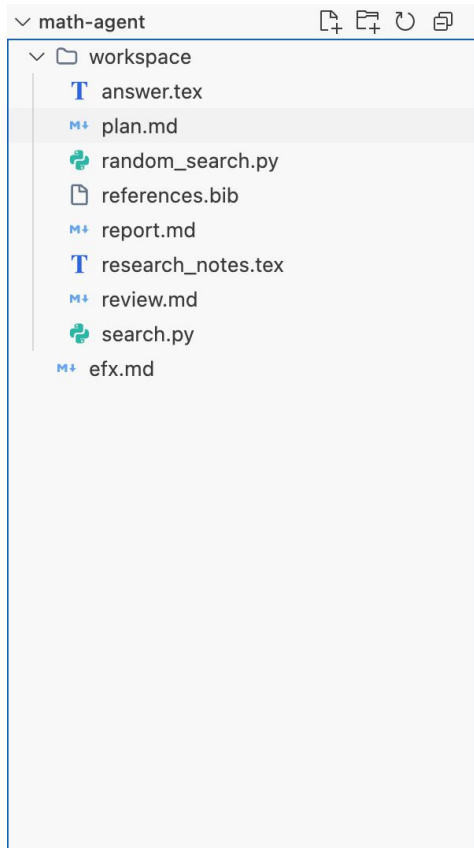
Exploring 2 files >

Thought for 1s >

Analyzed `answer.tex` #L1-41

Analyzed `research_notes.tex` #L1-31

How to use existing workflows....



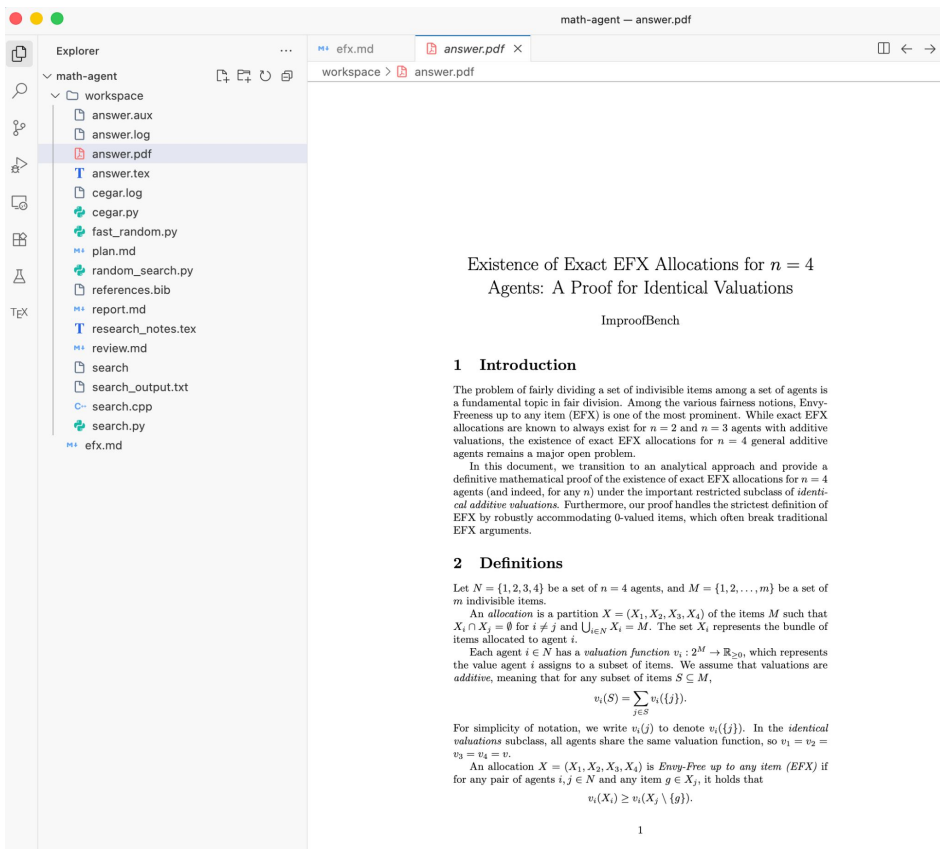
workspace > plan.md

- 1 1. Implement a Python script using a computational solver (e.g., PySAT, Z3, or SciPy/MIP) to systematically search for a counterexample to the existence of exact EFX allocations for $n=4$ agents with additive valuations.
- 2 2. The script should iterate over small item counts (e.g., $m=4$ to $m=10$) and explore the space of additive valuation profiles (e.g., restricting to small non-negative integer values to make the search space tractable).
- 3 3. For each generated valuation profile, encode the EFX conditions into the solver to check if at least one exact EFX allocation exists. If the solver determines that no EFX allocation is possible for a specific profile, we have found a counterexample.
- 4 4. If a counterexample is found, extract the exact valuation matrix, manually verify the non-existence of an EFX allocation, and rewrite ``answer.tex`` to present this definitive counterexample.
- 5 5. If the computational search does not yield a counterexample within reasonable limits, transition to the analytical approach. Attempt to mathematically prove the existence of exact EFX for $n=4$, perhaps starting with restricted subclasses of additive valuations, and rigorously document the proof in ``answer.tex``. The final document must present either a proven existence or a concrete counterexample.

6

A small thumbnail image of a document page. It contains several lines of text and a diagram with nodes and arrows, likely representing a flowchart or a network structure.

How to use existing workflows....



```
Ran pdflatex -interaction=nonstopmode answer.tex >
```

Explored 1 file >

I have successfully coordinated the `improofbench_planner`, `improofbench_worker`, and `improofbench_critic` subagents to produce a rigorous mathematical proof.

Summary of Process: The subagents iterated over the problem over 3 rounds. Initially, a dual strategy of computational search for a counterexample and analytical proof was formulated. After the computational search over small item numbers yielded no counterexamples for $n = 4$ agents with additive valuations, the focus shifted to the analytical approach. The worker provided a complete, formal, constructive proof using Leximin allocation for the important restricted subclass of **identical additive valuations**, while correctly establishing the open nature of the general problem. The proof robustly handles edge cases, such

Working..

Use Case: Generating Open Problems

Prompt:

This is my DBLP page: <https://dblp.org/pid/53/1607.html> please fetch the tex file from the last 5 papers on arXiv, read them in detail and extract two open problems for each. For each problem create an entry as an md file that describes the open problem with the background and mentions the paper it came from.

Paper 1: Contextual Online Bilateral Trade (2602.12903)

#	Problem	File
1	Budget vs. Regret Tradeoff — Characterize minimax regret as a function of allowed budget violation in one-bit feedback	problem_01_bilateral_trade_budget_regret_tradeoff.md
2	Noisy/Imprecise Feedback — What regret rates are achievable when valuations are only approximately linear?	problem_02_bilateral_trade_noisy_feedback_k.md

Paper 2: Autobidding Equilibria in Sponsored Shopping (2602.21966)

#	Problem	File
3	Computational Complexity — Is computing an autobidding equilibrium PPAD/FIXP-complete?	problem_03_autobidding_complexity.md
4	Learning Dynamics Convergence — Do natural autobidding algorithms converge in multi-item	problem_04_autobidding_convergence.md

```

open-problems
├── papers
│   ├── problem_01_bilateral_trade_budget_re...
│   ├── problem_02_bilateral_trade_noisy_fee...
│   ├── problem_03_autobidding_complexity....
│   ├── problem_04_autobidding_learning_dy...
│   ├── problem_05_full_swap_regret_convex...
│   ├── problem_06_discretized_calibration_hi...
│   ├── problem_07_variance_allocation_graph...
│   ├── problem_08_variance_allocation_lower...
│   ├── problem_09_bafo_cost_variation.md
│   └── problem_10_bafo_incomplete_informat...

```

```

** problem_10_bafo_incomplete_information.md > # Open Problem: BAFO Auctions with Incomplete Information
1  # Open Problem: BAFO Auctions with Incomplete Information Among Sellers
2
3  ## Source Paper
4  **Procurement Auctions with Best and Final Offers**
5  Vasilis Gkatzelis, Randolph Preston McAfee, Renato Paes Leme
6  arXiv: [2503.10910] (https://arxiv.org/abs/2503.10910)
7
8  ## Background
9
10 The paper's efficiency results for BAFO procurement auctions hold under two key
    assumptions:
11
12 1. **Full information among sellers:** Each seller knows the costs of all other
    sellers. This enables backward induction and the computation of subgame perfect
    equilibria.
13 2. **The buyer has no information** about seller costs and uses the auction for
    price discovery.
14
15 The full-information assumption is natural in some settings (e.g., well-established
    industries where competitors' cost structures are roughly known) but unrealistic in
    others (e.g., innovative products, new market entrants).
16
17 The paper's main result – that every SPE of a BAFO auction is efficient – relies
    fundamentally on sellers' ability to anticipate each other's behavior. In the
    Name-Your-BAFO auction, a seller decides their bid knowing what subsequent sellers
    will do (by backward induction on the game tree). In the descending auction, a
    seller's decision to freeze depends on predicting how other sellers will respond.
18
19 ## Open Problem
20
21 **What efficiency guarantees can BAFO auctions provide when sellers have incomplete
    information about each other's costs?**
22
23 Specific formulations:
24
25 1. **Bayesian setting:** Suppose sellers' costs are drawn from commonly known
    distributions. What is the Price of Anarchy of Bayes-Nash equilibria in
    Name-Your-BAFO and Descending-with-BAFO auctions? Is efficiency still guaranteed, or
    does it break down?

```


Use case: Educational Algorithm Implementation

Prompt (refined using Gemini):

Act as an expert in Algorithmic Game Theory and C++ programming.

Task: Write an educational C++ implementation of the Lemke-Howson algorithm for finding a Nash equilibrium in a 2-player bimatrix game.

Context: The code and its comments will serve as standalone lecture notes for students learning computational game theory. A reader must be able to read the source file from top to bottom and understand both the algorithmic mechanics and the theoretical justification.

Specific Requirements:

- Theoretical Overview:** Begin the file with a comprehensive block comment explaining the mathematical intuition behind the Lemke-Howson algorithm. This should cover best-response polytopes, slack variables, labels, and the parity argument for why the algorithm always terminates at an equilibrium.
- Step-by-Step Annotations:** Inside the code, clearly comment on the tableau setup, the logic for choosing the entering and dropping variables, and the pivoting steps.
- Code Quality:** Use modern C++ (C++17 or later) and standard library containers (e.g., `std::vector`). Prioritize algorithmic clarity and readability over raw performance.
- Assumptions:** Explicitly state any assumptions made (e.g., assuming the game is non-degenerate) and briefly explain how degeneracy would normally be handled (e.g., lexicographic pivoting).
- Example:** Include a `main()` function that initializes a classic bimatrix game, runs the algorithm, and prints the step-by-step pivoting process alongside the

```

/*
=====
* LEMKE-HOWSON ALGORITHM — EDUCATIONAL IMPLEMENTATION
* Finding a Nash Equilibrium in a 2-Player Bimatrix Game
* =====
*
* OVERVIEW
* -----
* Given a 2-player game specified by payoff matrices A (m×n) and B (m×n),
* where Player 1 chooses a row (m strategies) and Player 2 chooses a column
* (n strategies), the Lemke-Howson algorithm finds one Nash equilibrium in
* mixed strategies by following edges of a certain polytope.
*
*
* 1. BEST-RESPONSE POLYTOPES
* -----
* Assume all payoff entries are strictly positive (we can always add a
* constant to achieve this without changing equilibria).
*
* Define the "unscaled" best-response polytopes:
*
* P = { x in R^m : x >= 0, B^T x <= 1 } (constrains Player 1)
* Q = { y in R^n : y >= 0, A y <= 1 } (constrains Player 2)
*
* The origin is a vertex of both P and Q. Any non-zero vertex x of P
* can be normalized to a mixed strategy x / (sum x_i), and similarly for Q.
*
*
* 2. LABELS AND COMPLEMENTARY SLACKNESS
* -----
* We introduce slack variables:
*
* Tableau T1 (n rows): B^T x + s = 1, x >= 0, s >= 0
* Tableau T2 (m rows): A y + r = 1, y >= 0, r >= 0
*
* Each variable carries a "label" from {1, ..., m+n}:
*

```


Use case: Educational Algorithm Implementation

```
// -----
//  lemke_howson
//  -----
//  Runs the Lemke-Howson algorithm on the bimatrix game (A, B).
//
//  Parameters:
//  ^
//  drop_label  m×n payoff matrix for Player 1
//              m×n payoff matrix for Player 2
//  drop_label - the initial label to drop (1 to m+n)
//  verbose     - if true, print every pivot step
//
//  Returns: (x, y) where x is Player 1's equilibrium mixed strategy
//           and y is Player 2's equilibrium mixed strategy.
//  -----
std::pair<std::vector<double>, std::vector<double>>
lemke_howson(const std::vector<std::vector<double>>& A,
             const std::vector<std::vector<double>>& B,
             int drop_label,
             bool verbose = true)
{
    const int m = static_cast<int>(A.size()); // Player 1 strategies
    const int n = static_cast<int>(A[0].size()); // Player 2 strategies

    // -----
    //  Shift payoffs so that every entry is strictly positive.
    //  This is required so that the polytopes P and Q are bounded and
    //  the minimum-ratio test is always well-defined.
    //  Adding a constant to all entries does not change the set of NE.
    //  -----
    double min_A = 0.0, min_B = 0.0;
    for (int i = 0; i < m; ++i)
        for (int j = 0; j < n; ++j) {
            min_A = std::min(min_A, A[i][j]);
            min_B = std::min(min_B, B[i][j]);
        }
}
```

Techniques that are usually helpful

The AI as a smart junior collaborator: What advice / hints would you give a student working on this problem?

- Provide as much intuition and references as possible. If you have old research notes lying around for unsolved problems, try to enter them as context .
- Give as much detail as possible (what special cases are known, impossibilities, potentially some useful lemmas).
- Add related references to the context (“this may be related to Lemma ... in the PDF attached”).

Part 4: Data Science with Agentic AI

Data science example: NYC Citibike Data

Let's start exploring the data

citibike ▾

Citibike publishes its data on <https://citibikenyc.com/system-data>, can you please check the format of the data, give me examples of some records and give me some basics stats about it.

+ Gemini 3.5 Flash (High) ▾

Local ▾

🔒 Allow reading this URL?

citibikenyc.com

- 1 Yes, allow this time
- 2 Yes, and always allow in this project
- 3 Yes, and always allow
- 4 No (tell the agent what to do instead)

Skip Submit ↗

AG asking for permission to read this webpage.

A. Trip History Data (CSV)

The historical trip dataset has evolved over time. In early 2021, Citi Bike transitioned to a standardized schema designed by Lyft:

Current Format (2021–Present)

- `ride_id`: Unique string identifier for each ride.
- `rideable_type`: Type of bike (e.g., `classic_bike`, `electric_bike`).
- `started_at`: Start timestamp (YYYY-MM-DD HH:MM:SS).
- `ended_at`: End timestamp (YYYY-MM-DD HH:MM:SS).
- `start_station_name`: Name of the starting station.
- `start_station_id`: ID code of the starting station.
- `end_station_name`: Name of the destination station.
- `end_station_id`: ID code of the destination station.
- `start_lat` / `start_lng`: Latitude and longitude coordinates of the start station.
- `end_lat` / `end_lng`: Latitude and longitude coordinates of the end station.
- `member_casual`: Type of user subscription (`member` for annual subscribers, `casual` for single/day pass riders).

Historical Format (Prior to 2021)

Older datasets included rider demographics and individual bike tracking IDs:

- `tripduration`: Duration of the trip in seconds.
- `starttime` / `stoptime`: Timestamps for start and stop.
- `start_station_id` / `name` / `latitude` / `longitude`
- `end_station_id` / `name` / `latitude` / `longitude`
- `bikeid`: Integer tracking number of the physical bike.
- `usertype`: `Subscriber` (annual member) or `Customer` (24-hour/3-day pass).
- `birth_year`: Rider's year of birth (if registered).
- `gender`: Rider's gender (0 = unknown, 1 = male, 2 = female).

Data science example

Current Format (CSV)

CSV



```
ride_id,rideable_type,started_at,ended_at,start_station_name,start_station_id,
end_station_name,end_station_id,start_lat,start_lng,end_lat,end_lng,member_cas
ual
55034D2134300000,classic_bike,2024-05-01 08:00:15,2024-05-01 08:12:45,E 17 St
& Broadway,5896.01,W 21 St & 6
Ave,6140.05,40.737397,-73.990693,40.741740,-73.994156,member
```

Historical Format (CSV)

CSV



```
tripduration,starttime,stoptime,start station id,start station name,start
station latitude,start station longitude,end station id,end station name,end
station latitude,end station longitude,bikeid,usertype,birth year,gender
750,2019-06-01 12:05:00,2019-06-01 12:17:30,164,E 47 St & 2
Ave,40.753231,-73.970325,504,1 Ave & E 16
St,40.732219,-73.981656,28540,Subscriber,1990,1
```

• Annual Totals:

- **2013:** ~5.8 million trips (partial year)
 - **2016:** ~14.1 million trips
 - **2019:** ~21.0 million trips
 - **2021:** ~28.0 million trips
 - **2024: Over 45 million trips** (the busiest year in system history)
- **Peak Monthly Volume:** October 2024 holds the record for highest monthly ridership with **over 5.13 million rides**.

Trip Characteristics

- **Average Trip Duration:** Typically between **11 and 15 minutes** (median is slightly lower at **9 to 12 minutes**).
 - *Members* average around 12 minutes.
 - *Casual riders* average around 20–25 minutes.
- **Average Trip Distance:** Approximately **1.1 to 1.2 miles** (straight-line distance).
- **Data Filtering:** Before publishing data, Citi Bike removes trips shorter than 60 seconds (deemed "false starts" or docking errors) and trips used by technicians servicing the fleet.

Fleet Behavior

- **E-Bike Share:** As of 2024, e-bikes represent only about 40% of the active fleet but generate nearly **66% of all trips**, indicating higher daily usage per bike than traditional pedal models.
- **Transit Connectivity:** Roughly **85% of users** incorporate Citi Bike as a "first/last-mile" link to other public transit systems (like subways or commuter rail).

Data science example

Please download the data from Jan 2025 and then from Jun 2025 and for each month plot a histogram of trip length. Plot different histograms for classic and electric bikes

+ Gemini 3.5 Flash (High) ^



AG will install necessary packages:

```
~/repos/citibike $ python3 -m pip install pandas matplotlib request
s
```

Citi Bike Trip Length Analysis (January & June 2025)

This report presents a comparative analysis of Citi Bike trip lengths (in minutes) for January 2025 (representing winter ridership) and June 2025 (representing summer ridership). The analysis breaks down trips by bicycle type: **Classic Bikes** vs. **Electric Bikes**.

Key Findings

- Seasonal Volume Shift:** June ridership is more than **double** that of January. The warmer weather in summer significantly increases demand across both bike types.
- Electric Bike Dominance:** In both months, electric bikes account for the vast majority of trips—representing approximately **70.3% of trips in January** and **71.0% in June**.
- Trip Durations Increase in Summer:** The average and median trip lengths increase by **25% to 38%** in June compared to January, as riders take advantage of pleasant weather to ride further.
- Bike Type Performance:**
 - Winter (January):** Electric bike trips are slightly shorter on average than classic bike trips (9.20 mins vs 9.71 mins). The pedal assist allows riders to reach their destinations

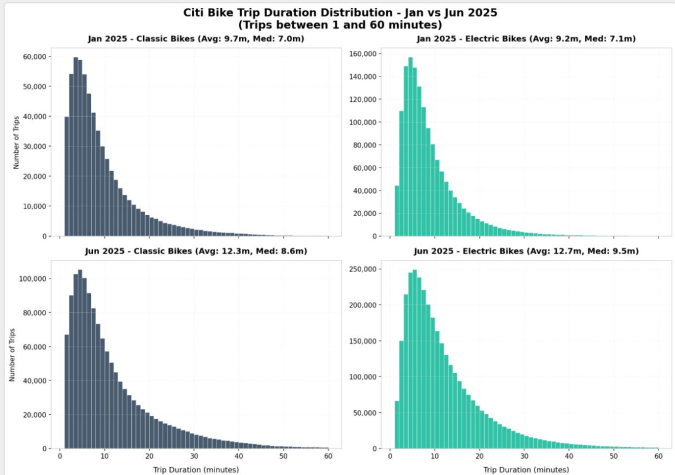
Trip Statistics Table

Month	Bike Type	Total Trips (1-120 min)	Average Duration	Median Duration	Std Dev
January 2025	Classic	630,020	9.71 min	7.01 min	8.84 min
January 2025	Electric	1,492,971	9.20 min	7.05 min	7.73 min
June 2025	Classic	1,375,260	12.27 min	8.62 min	11.48 min
June 2025	Electric	3,371,856	12.71 min	9.53 min	11.13 min

Note: Trips under 1 minute and over 120 minutes were excluded as outliers (representing false starts or unreturned bikes).

Trip Length Distributions (Histograms)

Below are the histograms showing the distributions of trip durations for classic and electric bikes across both months. Each plot represents trips between 1 and 60 minutes, bucketed into 1-minute intervals.



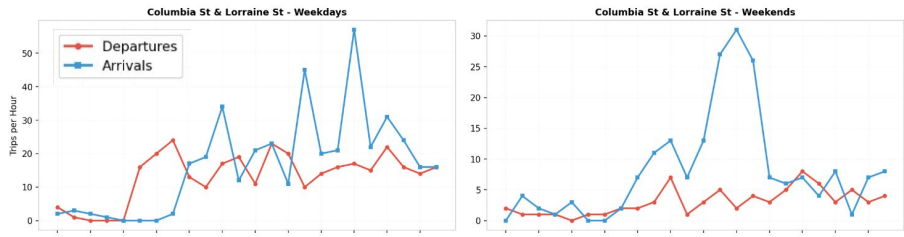
Citi Bike Trip Length Histograms

Data science example

Let's use the Jun 2025 dataset to compute for each station the number of trips originating from that station and the number of trips ending at that station. Then rank stations by the difference. Tell me the top and bottom 5 in the ranking and then try to explain why is that.

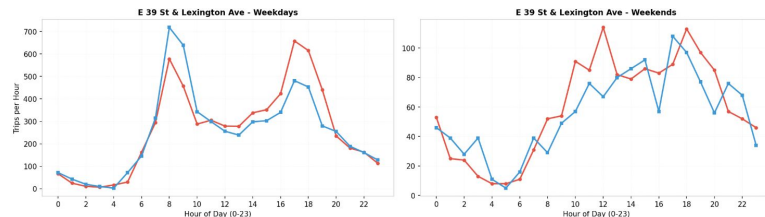
4. Columbia St & Lorraine St (Red Hook Transit Desert)

- **Difference: -203** (391 Departures | 594 Arrivals)
- **Why:** Red Hook has no subway access. In June, people ride Citibikes *into* Red Hook to visit waterfront restaurants, bars, and parks (especially on weekend afternoons). After eating and drinking, many opt for rideshares, buses, or ferries to leave, resulting in a net weekend accumulation of bikes.



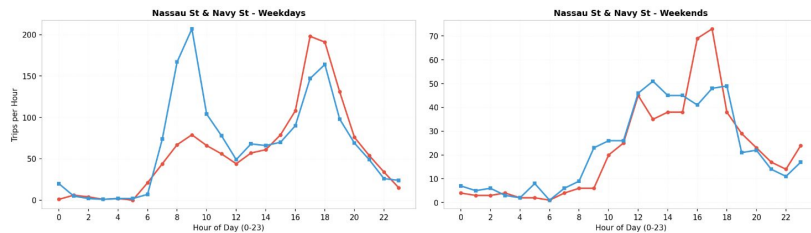
5. E 39 St & Lexington Ave (Grand Central Commuter Corridor)

- **Difference: +361** (7,755 Departures | 7,394 Arrivals)
- **Why:** Located just south of Grand Central Terminal. It receives arrivals in the morning, but experiences a massive surge of departures in the PM rush (+422 net departures between 4-7 PM) as workers head back to major transit hubs or residential neighborhoods.



5. Nassau St & Navy St (Brooklyn Navy Yard Gate)

- **Difference: -203** (1,918 Departures | 2,121 Arrivals)
- **Why:** Located at the entrance to the Brooklyn Navy Yard (an office/industrial hub) and Wegmans. It experiences a heavy inflow of morning commuters riding to work, with fewer returning by bike in the evening.



Data science example

Let's use the Jun 2025 dataset to compute for each station the number of trips originating from that station and the number of trips ending at that station. Then rank stations by the difference. Tell me the top and bottom 5 in the ranking and then try to explain why is that.

I will write another Python script

`/Users/renatopl/repos/citibike/analyze_station_temporal.py` to examine the hourly and weekday vs. weekend patterns of departures and arrivals for the top 5 and bottom 5 stations. This will give us concrete, data-backed evidence for why these stations experience such extreme net imbalances.

Edited  `analyze_station_temporal.py` +180 -0

I will run the temporal analysis script to compute the hourly profiles of trips and arrivals for these key stations. This will run as a background task.

Ran `python3 analyze_station_temporal.py` >

Thought for 1s >

I will write a quick script `/Users/renatopl/repos/citibike/summarize_results.py` to parse the temporal statistics and print a summary for each of the 10 station weekday vs. weekend ratios, peak hours, and peak flow directions). This will explain the underlying causes with concrete data.

Edited  `summarize_results.py` +44 -0

I will execute the summarizer script to inspect the detailed temporal patterns

```
import os
import sys
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

def main():
    data_dir = 'data/202506_tripdata'
    if not os.path.exists(data_dir):
        print(f"Data directory {data_dir} does not exist.")
        sys.exit(1)

    csv_files = []
    for root, dirs, files in os.walk(data_dir):
        for file in files:
            if file.endswith('.csv') and not file.startswith('_'):
                csv_files.append(os.path.join(root, file))

    if not csv_files:
        print("No CSV files found in {data_dir}")
        sys.exit(1)

    # The 10 stations to analyze
    target_stations = {
        # Top 5 Sources (net departures)
        'Broadway & W 61 St': 'source',
        'W 49 St & 8 Ave': 'source',
        'W 54 St & 9 Ave': 'source',
        'Columbus Ave & W 59 St': 'source',
        'E 39 St & Lexington Ave': 'source',
        # Bottom 5 Sinks (net arrivals)
        'Greenwich St & W Houston St': 'sink',
        'E 161 St & River Ave': 'sink',
        'Front St & Washington St': 'sink',
        'Columbia St & Lorraine St': 'sink',
        'Nassau St & Navy St': 'sink'
    }

    # Structure to hold temporal counts:
    # temporal_stats[station][event_type][day_type][hour] = count
```

Bottom 5 Net Sinks (Highest Negative Difference: Arrivals > Departures)

These stations accumulate excess bikes and require manual pick-up/rebalancing by Citibike trucks.

1. Greenwich St & W Houston St (Hudson Square Tech District)

- Difference:** -483 (8,387 Departures | 8,870 Arrivals)
- Why:** A classic office commuter sink. Located in Hudson Square, just two blocks from Google's massive new **St. John's Terminal** office and Disney's new NYC headquarters. Weekday mornings see a massive influx of arrivals (-981 net arrivals between 7-10 AM) that far outweighs the evening outbound flow.

2. E 161 St & River Ave (Yankee Stadium)

- Difference:** -208 (2,690 Departures | 2,898 Arrivals)
- Why:** Event-driven. In June (peak baseball season), fans ride Citibikes to evening games (peaking at 6 PM). After the game, fans exit the stadium (peaking at 9 PM) but disproportionately choose to take the subway or rideshares home (due to fatigue, alcohol consumption, or lack of available bikes), leaving a net surplus of bikes docked at the stadium.

3. Front St & Washington St (DUMBO photo spot)

- Difference:** -206 (4,666 Departures | 4,872 Arrivals)
- Why:** DUMBO is a major tech office sink on weekdays and a tourist/recreational destination on weekends. Many riders travel over the Brooklyn or Manhattan bridges into DUMBO, dock their bikes, and choose to return via subway (F train) or ferry, leaving their bikes behind.

4. Columbia St & Lorraine St (Red Hook Transit Desert)

- Difference:** -203 (391 Departures | 594 Arrivals)
- Why:** Red Hook has no subway access. In June, people ride Citibikes *into* Red Hook to visit waterfront restaurants, bars, and parks (especially on weekend afternoons). After eating and drinking, many opt for rideshares, buses, or ferries to leave, resulting in a net weekend accumulation of bikes.

5. Nassau St & Navy St (Brooklyn Navy Yard Gate)

- Difference:** -203 (1,918 Departures | 2,121 Arrivals)
- Why:** Located at the entrance to the Brooklyn Navy Yard (an office/industrial hub) and Wegmans. It experiences a heavy inflow of morning commuters riding to work, with fewer returning by bike in the evening.

Data science example

New York also publishes the list of TLC rides (<https://www.nyc.gov/site/tlc/about/tlc-trip-record-data.page>). Can you grab the data for the same period? I want a side by side comparison of the Citibike and the TLC dataset. How can we do that?

+ Gemini 3.5 Flash (High) ^



Now I have a thorough understanding of the TLC data. Let me first quickly verify the download URLs work and check if June 2025 data is available, then present a plan to the user.

```
Ran curl -sI "https://d37ci6vzurychx.cloudfront.net/trip-data/yellow_t >
```

Yellow taxi June 2025 is available (~73MB). Let me check the other TLC data types too.

```
Ran echo "=== Green Taxi ===" && curl -sI "https://d37ci6vzurychx.clou >
```

All datasets are available for June 2025. Here's what we're working with:

Dataset	Size	Description
Yellow Taxi	~74 MB	Traditional yellow cabs
Green Taxi	~1.2 MB	Boro taxis (outer boroughs)
FHV	~24 MB	For-hire vehicles (black cars, limos)
HVFFHV	~491 MB	High-volume FHV (Uber, Lyft)
Zone Lookup	~12 KB	Maps LocationID → Borough/Zone

Here we ask AG to download a different dataset (TLC data) in a different format and compare them. Our instructions are vague here on what comparison we want.

figures out how to get the data

preliminary analysis of the dataset

Data science example

Plan to reconcile the data:

The Data Alignment Problem

The two datasets use **completely different spatial systems**:

	Citibike	TLC (Taxi/Rideshare)
Location format	Station name + lat/long coordinates	TLC Taxi Zone IDs (265 zones across NYC)
Granularity	Point-level (station dock)	Zone-level (~neighborhood)
Coverage	Manhattan, parts of Brooklyn/Queens/Bronx	All 5 boroughs + airports

To do a meaningful side-by-side comparison, we need to **map Citibike stations into TLC Taxi Zones** so both datasets share the same geographic unit.

Asks the user what comparison we want:

Which comparison analyses interest you the most? I'll download the TLC data and build these.

Multi-select

- 1 ****Zone-level flow imbalance comparison**** — Compute net trip flow (pickups - dropoffs) per TLC zone for both datasets. Do zones that are Citibike "sources" also have more taxi pickups than dropoffs? Or do taxis compensate for the bike imbalance?
- 2 ****Temporal patterns by zone**** — Compare hourly demand curves for bikes vs taxis in the same neighborhoods. Do they compete (substitute) or complement each other (e.g., bikes for short trips, taxis for longer ones)?
- 3 ****Trip volume heatmap**** — Total trip volume per zone for both modes. Where is each mode dominant? Are there zones that are bike-heavy but taxi-light (or vice versa)?
- 4 ****All of the above**** — Do a comprehensive comparison covering flow imbalance, temporal patterns, and volume heatmaps.
- 5 Other (write your answer)

Skip Submit

Data science example

Citibike vs TLC Trip Data — Comprehensive Comparison (June 2025)

Goal

Download NYC TLC trip data for June 2025, align it spatially with our existing Citibike data, and produce three comparison analyses: flow imbalance, temporal patterns, and volume heatmaps.

Data Sources

Dataset	URL	Size	Format
Yellow Taxi	https://d37ci6vzurychx.cloudfront.net/trip-data/yellow_tripdata_2025-06.parquet	~74 MB	Parquet
Green Taxi	https://d37ci6vzurychx.cloudfront.net/trip-data/green_tripdata_2025-06.parquet	~1.2 MB	Parquet
FHV	https://d37ci6vzurychx.cloudfront.net/trip-data/fhv_tripdata_2025-06.parquet	~24 MB	Parquet
HVHV (Uber/Lyft)	https://d37ci6vzurychx.cloudfront.net/trip-data/hvhv_tripdata_2025-06.parquet	~491 MB	Parquet
Zone Lookup (CSV)	https://d37ci6vzurychx.cloudfront.net/misc/taxi_zone_lookup.csv	~12 KB	CSV
Zone Shapefile	https://d37ci6vzurychx.cloudfront.net/misc/taxi_zones.zip	~0.5 MB	Shapefile

NOTE

The HVFHV file is ~491 MB (Uber/Lyft combined). This is the largest download but represents the dominant rideshare mode in NYC.

Spatial Alignment Strategy

Citibike uses station lat/lng coordinates, while TLC uses Taxi Zone IDs (265 zones). To compare them:

- Download the official TLC **taxi zone shapefile** (polygon boundaries)
- For each Citibike station (lat/lng), perform a **point-in-polygon** lookup to find which

Analysis 2 — Temporal demand patterns

- For selected high-activity zones, plot hourly demand curves (pickups) for Citibike vs TLC
- Split by weekday/weekend
- Show whether bikes and taxis substitute or complement each other

[NEW] [compare_volume.py](#)

Analysis 3 — Trip volume comparison

- Per-zone total trips for Citibike vs TLC
- Compute the Citibike "mode share" = Citibike trips / (Citibike + TLC trips)
- Create a bar chart of top zones by mode share
- Identify zones where bikes dominate vs where taxis/rideshare dominate

Phase 4: Report

[NEW] [citibike_vs_tlc_analysis.md](#)

- Comprehensive artifact with all findings, embedded plots, and interpretation

Verification Plan

Automated Tests

- Verify all TLC Parquet files download successfully and are readable
- Verify station-to-zone mapping produces matches for >95% of Citibike stations
- Verify all analysis scripts produce output plots without errors

Manual Verification

- Sanity-check that zone-level aggregations produce reasonable numbers
- Verify scatter plots and temporal charts are visually coherent

Open Questions

IMPORTANT

HVFHV file size: The Uber/Lyft dataset is ~491 MB. Should I include it, or stick with just Yellow+Green taxis to keep things faster? Including HVFHV gives a much more complete picture of NYC mobility (Uber/Lyft now dominate over yellow taxis), but will take longer to download and process.

NOTE

I plan to combine all TLC vehicle types (Yellow aggregate for the comparison. If you'd prefer Taxi vs Uber vs Citibike), let me know.

Let's stick with Yellow+Green taxis and

ignore HVFHV for now



Cancel

Add Comment

Since it is a complex task, AG will generate a plan that you can comment on before it executes.

Model poses open questions to the user.

Data science example

This is very interesting! Let's see: In zones where Citibike operates (e.g. Manhattan and parts of Brooklyn):

- **East Village** is the #1 Citibike volume zone (366,493 trips), which is larger than its Taxi volume (205,575 trips).
- **Williamsburg (North Side)** in Brooklyn has a massive Citibike volume (254,540 trips) and very low Taxi volume (24,972 trips) — a whopping **91% Citibike share!**
- In contrast, the top taxi zones are **Upper East Side South** (344,751 taxi trips vs. 53,992 Citibike), **Midtown Center** (330,428 taxi trips vs. 102,661 Citibike), and **Upper East Side North** (322,119 taxi trips vs. 68,860 Citibike). These are taxi-dominant zones.

Let's do some more analysis: what about the correlation of flows? Let's check which zones have the biggest flow imbalances in both, or how they relate. Let's run a quick script to print out:

Assets Generated in Your Workspace:

1. Aggregated comparison dataset:

citibike_vs_tlc_zone_totals.csv

2. Analysis scripts:

- download_tlc_data.py (Downloads Yellow/Green Parquet, Zone CSV, and shapefiles)
- map_stations_to_zones.py (Performs spatial point-in-polygon mapping)
- compare_trips.py (Aggregates trips and generates plots)

3. Visualizations:

• Busiest Zones Volume Comparison:

citibike_vs_tlc_volume_comparison.png (Grouped bar chart showing total monthly trips and bike mode share)

• Net Flow Imbalance Scatter Plot:

citibike_vs_tlc_flow_imbalance.png (Plots Citibike net flow vs. Taxi net flow, detailing commuter and leisure outliers)

• Hourly Demand Profiles:

citibike_vs_tlc_temporal_profiles.png (Normalized weekday vs. weekend hourly curves showcasing business vs. nightlife signatures)

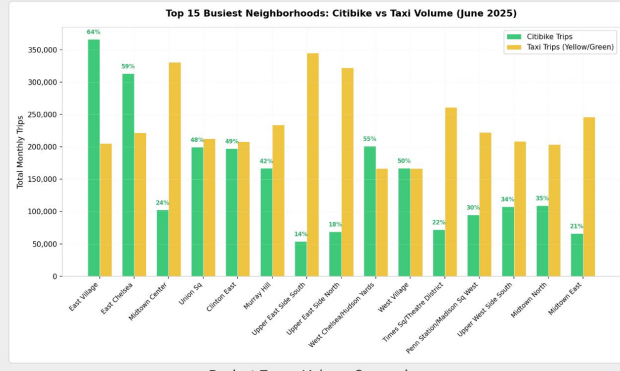
Data science example

Neighborhood Dominance

- Citibike-Dominant Zones:** Primarily located in Brooklyn (where taxis are rare) and residential/social areas of Manhattan.
 - Prospect Park, Brooklyn:** 98% Citibike Share (70,338 bike vs. 1,419 taxi trips)
 - Brooklyn Navy Yard:** 95% Citibike Share (34,923 bike vs. 1,802 taxi trips)
 - Williamsburg (North Side), Brooklyn:** 91% Citibike Share (254,540 bike vs. 24,972 taxi trips)
 - East Village, Manhattan:** 64% Citibike Share (366,493 bike vs. 205,575 taxi trips)
- Taxi-Dominant Zones:** Business districts, transit hubs, airports, and the Upper East Side.
 - JFK Airport / LaGuardia Airport:** 100% Taxi Share (Citibike does not serve the airports)
 - Upper East Side South:** 86% Taxi Share (344,751 taxi vs. 53,992 bike trips)
 - Midtown Center:** 76% Taxi Share (330,428 taxi vs. 102,661 bike trips)
 - Times Sq/Theatre District:** 78% Taxi Share (260,973 taxi vs. 72,330 bike trips)

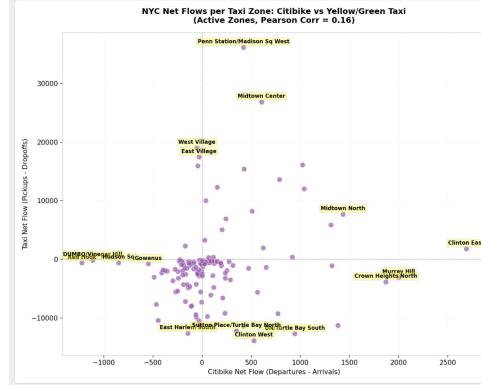
Top 15 Busiest Neighborhoods Volume Comparison

The chart below illustrates the total monthly trips for both modes in the top 15 busiest zones:



Scatter Plot of Net Flows

The scatter plot below charts Citibike net flow against Taxi net flow, showcasing key outliers:



Regression with NYC Weather Data

Our next task is to produce a regression model on how much the weather (temperature, rain, cloud) affects the citibike demand at a given station. Let's fix one particular station (figure out one that is most interesting), and then for June 2025 find a way to get weather data and build the regression.

2:38 PM  

Asks to install stats tools:

 Allow running this command?

```
/Library/Developer/CommandLineTools/usr/bin/python3 -c "import statsmodels, sklearn; print('Both installed')"
```

- 1 Yes, allow this time
- 2 No (tell the agent what to do instead)

Skip

Submit 

Figures out how to get NYC weather data:

I will test fetching the weather data for NYC in June 2025 using curl to verify the Open-Meteo response.

Run `curl -s "https://archive-api.open-meteo.com/v1/archive?latitude=40.7829&longitude=-73.9654&start_date=2025-06-01&end_date=2025-06-30&hourly=temperature_2m,precipitation,cloud_cover,wind_speed_10m&temperature_unit=fahrenheit" | head -n 30`

```
~/repos/citibike $ curl -s "https://archive-api.open-meteo.com/v1/archive?latitude=40.7829&longitude=-73.9654&start_date=2025-06-01&end_date=2025-06-30&hourly=temperature_2m,precipitation,cloud_cover,wind_speed_10m&temperature_unit=fahrenheit" | head -n 30
```

Waiting for user input..

 Allow running this command?

```
curl -s "https://archive-api.open-meteo.com/v1/archive?latitude=40.7829&longitude=-73.9654&start_date=2025-06-01&end_date=2025-06-30&hourly=temperature_2m,precipitation,cloud_cover,wind_speed_10m&temperature_unit=fahrenheit"
```

- 1 Yes, allow this time
- 2 No (tell the agent what to do instead)

Skip

Submit 

Regression

Plan for a complex workflow:

Implementation Plan: Weather Regression Model for Citibike Demand (June 2025)

This plan outlines the design and execution of a regression analysis showing how weather conditions (temperature, rain, cloud cover, and wind speed) affect Citibike demand in June 2025. To make this analysis particularly interesting and suitable for a lecture/demo, we will analyze and compare two different types of stations:

1. **Recreational/Leisure Station:** `7 Ave & Central Park South` (high tourist and leisure use, highly weather-sensitive).
2. **Commuter/Utility Station:** `8 Ave & W 31 St` (near Penn Station, high commuter utility, potentially more weather-resilient).

Proposed Changes

Dependencies and Libraries

We will install `statsmodels` and `scikit-learn` to support advanced regression modeling (like Poisson Regression) and diagnostic metrics.

- Installation command: `./Library/Developer/CommandLineTools/usr/bin/python3 -m pip install --index-url https://pypi.org/simple statsmodels scikit-learn`

Component 1: Sourcing Weather Data

We will write a script to download hourly historical weather data for June 2025 from the Open-Meteo Archive API (which uses ERA5/ERA5-Land reanalysis data).

- **Location:** Central Park coordinates (`latitude=40.7829`, `longitude=-73.9654`).
- **Variables:** Temperature (2m, in °F), Precipitation (in mm), Cloud Cover (%), and Wind Speed (10m, in mph).
- **Output:** Cached local JSON file `data/weather_june_2025.json` to prevent repeated API requests.

Component 2: Citibike Trip Extraction & Aggregation

We will write a python script to process the June 2025 Citibike trip data in chunks for memory efficiency.

- Filter trips departing from our two selected stations:
 - Leisure: `7 Ave & Central Park South` (Station ID / Name lookup)
 - Commuter: `8 Ave & W 31 St`
- Extract departure hour and aggregate to get hourly counts of departures (720 hours in June 2025).
- Create a complete time series template for June 2025 to ensure hours with zero departures are properly represented.

Component 3: Regression Modeling

For each station, we will build and compare two types of regression models:

1. **OLS Linear Regression (Baseline):** Models hourly departures directly as a linear combination of features. Good for a simple baseline.
2. **Poisson Regression (Generalized Linear Model with Log Link):** The statistically rigorous choice for count data. We will explain how Poisson modeling prevents negative count predictions and handles heteroscedasticity.

chooses two stations

discusses where to get weather data and what features to extract

proposes two regression methodologies

Starts a task list:

Weather Regression Model Task List

- ✓ Install Python dependencies (`statsmodels`, `scikit-learn`)
- Retrieve and cache hourly weather data from Open-Meteo for June 2025 (`data/weather_june_2025.json`)
- Aggregate hourly Citibike demand at selected stations for June 2025 (720 hours)
- Merge weather and demand data
- Build OLS and Poisson regression models, comparing leisure vs. commuter stations
- Generate visualizations (actual vs. predicted demand, marginal weather effects, coefficient IRRs)
- Create walkthrough and final analysis report (`weather_regression_analysis.md`)

Regression

Statistical Performance and Diagnostics

Metric	Leisure: Central Park S	Commuter: 8 Ave & W 31 St
Total Month Trips	13,051	11,779
OLS R^2	0.7192	0.6637
Poisson McFadden Pseudo R^2	0.7928	0.7356
Poisson AIC	5399.9	5205.6
Poisson Residual Deviance	2567.7	2168.4

NOTE

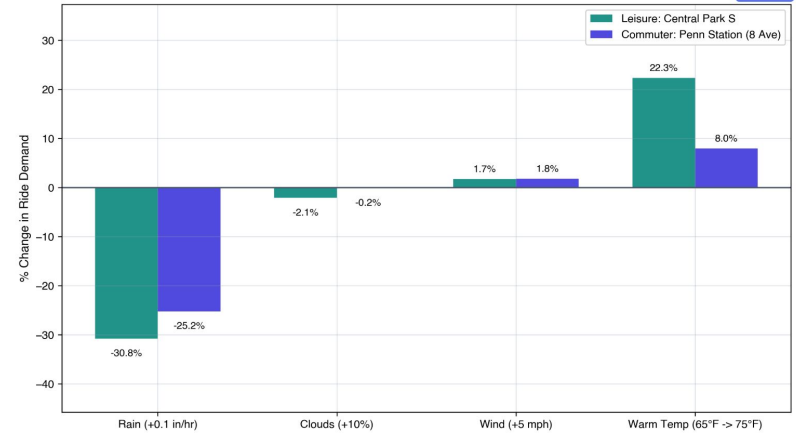
McFadden's Pseudo R^2 values between 0.2 and 0.4 represent an excellent model fit for GLM models. The model explains a very large proportion of the variance once hourly cycles and day-of-week baselines are controlled.

Regression Model Features & Coefficients Comparison

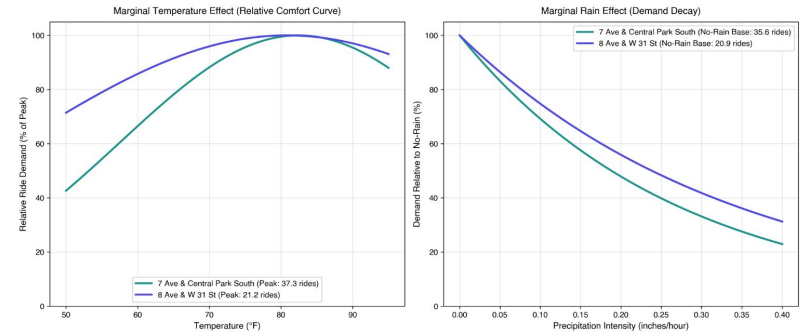
The table below displays the weather features, their coefficients under both OLS and Poisson models, and the model fit (R^2 or McFadden's Pseudo- R^2) for both locations.

Feature / Metric	Leisure: OLS Coeff	Leisure: Poisson Coeff	Commuter: OLS Coeff	Commuter: Poisson Coeff
Temperature (F°)	1.32856	0.13354	0.78197	0.05726
Temperature Squared (F^2)	-0.00811	-0.00081	-0.00490	-0.00035
Precipitation (inches)	-56.25187	-3.68160	-49.98171	-2.90707
Cloud Cover (%)	-0.03755	-0.00214	0.00227	-0.00016
Wind Speed (mph)	-0.01502	0.00344	0.13563	0.00354
Model Fit (R^2 / Pseudo- R^2)	0.7192 (R^2)	0.7928 (Pseudo- R^2)	0.6637 (R^2)	0.7356 (Pseudo- R^2)

Weather Sensitivity: Demand Impact Comparison



Weather Sensitivity Comparison



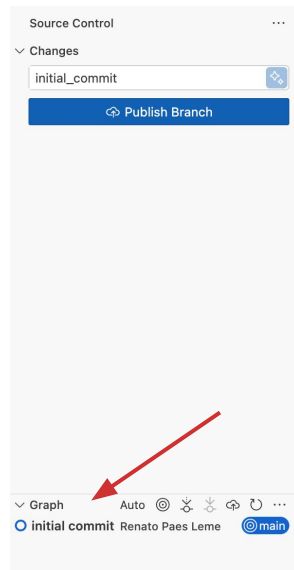
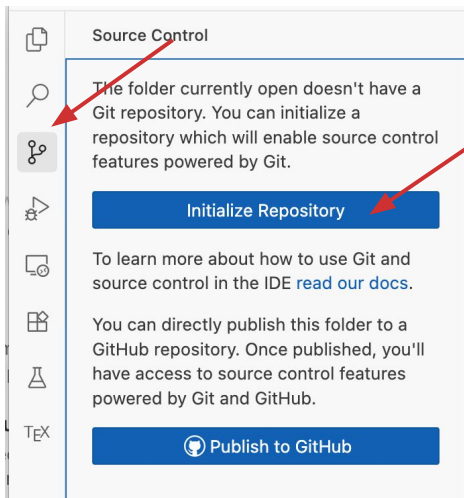
Is it correct?

- Remember... no magic bullet. You still need to check the work, similar to when working a research assistant.
- One trick is to ask another student to review the work. You can ask:
“Instantiate an independent evaluation agent with no prior context of the current development thread. Direct the agent to execute a comprehensive audit and generate a detailed report of all identified errors and inaccuracies.”
 - Alternative: open a new conversation
 - Try asking different models to do the verification.
 - Try asking k separate agents to do independent verifications.

Is it correct?

- If it all passes, it is time for a human to check it.
- Ask Antigravity to explain things:
 - “Can you explain what function X is doing?”
 - “Please add comments in file Y to explain what each line is doing.”
- Or you can challenge it with corner cases:
 - “Write a test that verifies if this corner case is correct.”

Using version control (git) as a safety net



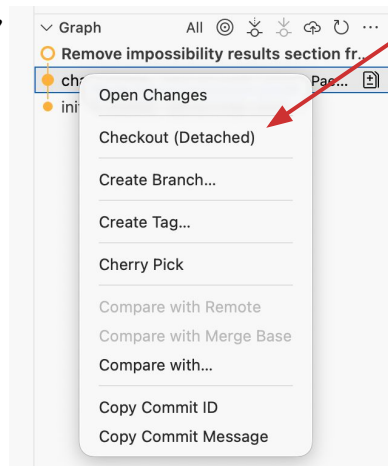
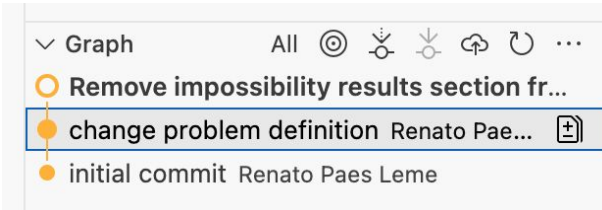
When pair-programming with AI agents, version control is your safety net. Checking out previous states or branching your codebase allows you to experiment freely, track the agent's changes, and easily roll back if the implementation heads in an unexpected direction.

If you don't want to learn git, ask the agent to do execute commits, pull, push, checkout on your behalf....

math-agent

create an initial git commit

+ Gemini 3.5 Flash (High) ▾



Visualization

First we ask to help us refine the prompt:

Let's visualize imbalance in departures and arrivals in citibike. Please write for me a prompt to build an application (javascript) that visualizes the imbalance along 2025 where I can choose the month in the UI.

1. Project Context & Objective

Build a modern, interactive single-page web application (HTML, CSS, JavaScript) that visualizes the monthly spatial imbalance between departures and arrivals across all NYC Citibike stations throughout the year **2025**.

The purpose of this visualization is to highlight spatial imbalances (i.e., stations acting as "sources" that accumulate net departures, and "sinks" that accumulate net arrivals) to help students/lecturers analyze rebalancing demands.

2. Architecture & Tech Stack

Data Pipeline (Python Preprocessing)

Citibike CSV files are massive (often >200MB and millions of rows per month). Loading these directly in the browser is not feasible. Write a preprocessing Python script (`preprocess_imbalance.py`) to prepare the data:

- **Source Data:** Downloads the monthly `.zip` files for 2025 from `https://s3.amazonaws.com/tripdata/2025MM-citibike-tripdata.zip` (for months `01` through `12`).
- **Processing:** For each month, read the trip data in chunks and compute:
 - For each station: total departures, total arrivals, net difference (`departures - arrivals`), and average coordinates (`lat`, `lng`).

Paste the prompt in a new conversation:

citibike ▾

Build a Citibike Flow Imbalance Interactive Visualizer (2025)

1. Project Context & Objective

Build a modern, interactive single-page web application (HTML, CSS, JavaScript) that visualizes the monthly spatial imbalance between departures and arrivals across all NYC Citibike stations throughout the year 2025.

The purpose of this visualization is to highlight spatial imbalances (i.e., stations acting as "sources" that accumulate net departures, and "sinks" that accumulate net arrivals) to help students/lecturers analyze rebalancing demands.

2. Architecture & Tech Stack

Data Pipeline (Python Preprocessing)

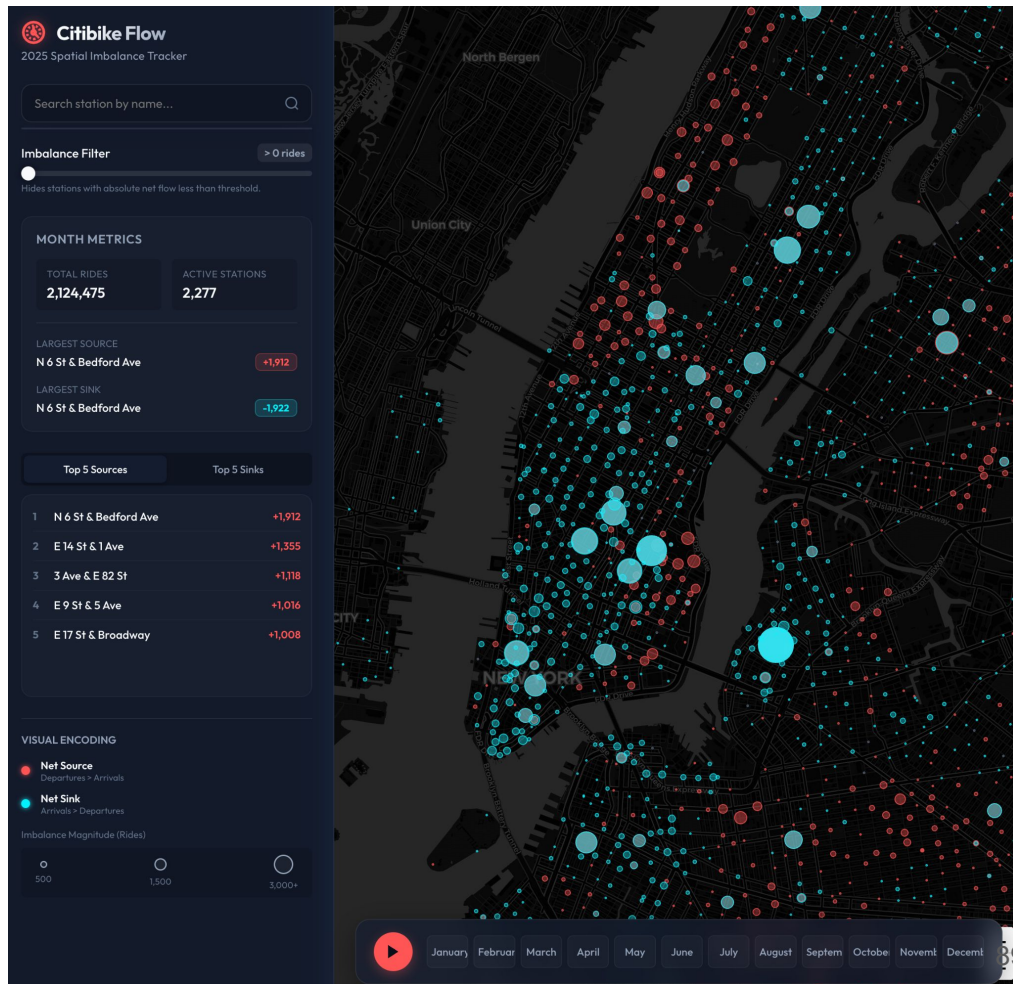
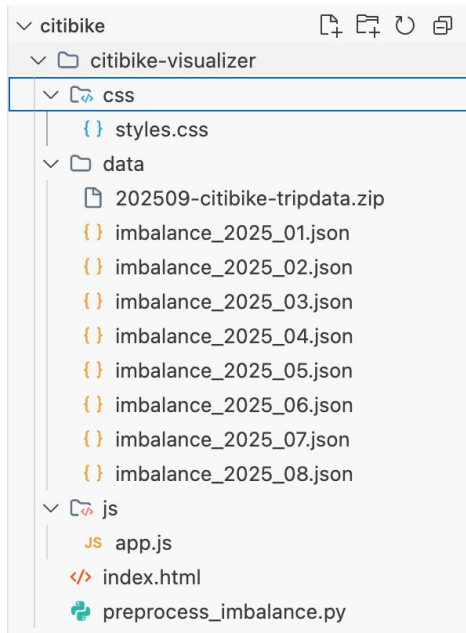
+ Gemini 3.5 Flash (High) ▾



Local ▾

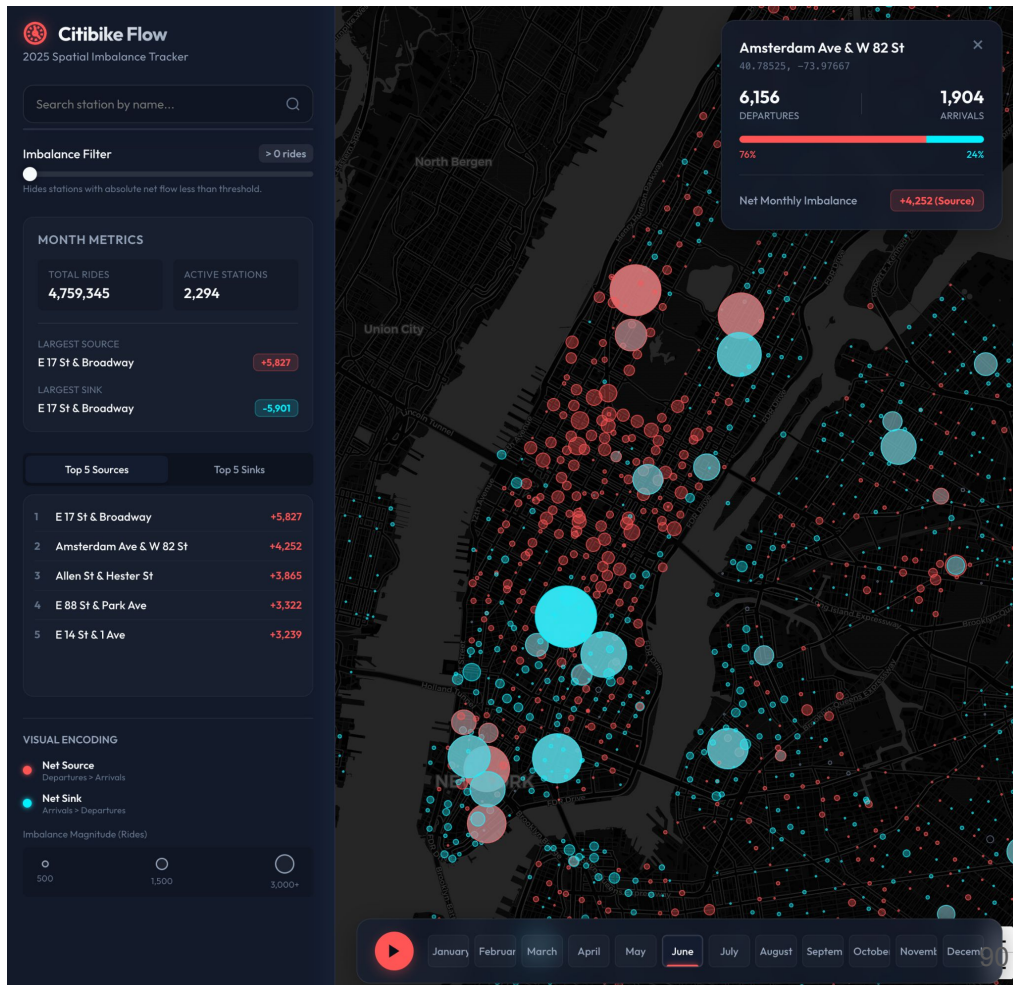
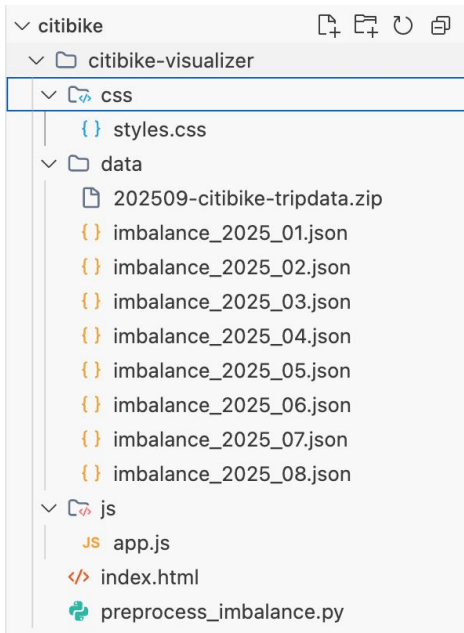
Visualization

Different agents are dispatched to do different parts of the project: data processing, create HTML/Javascript, ...



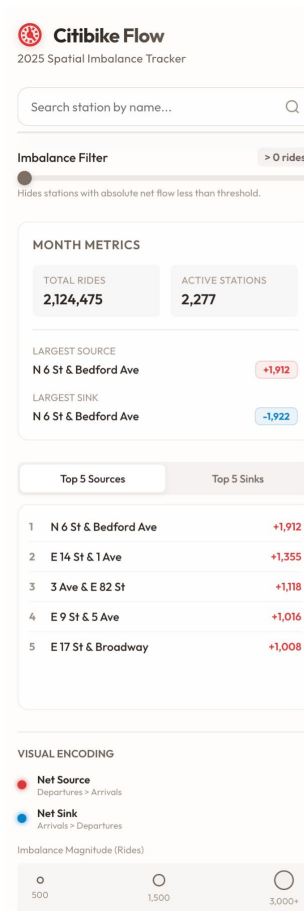
Visualization

Different agents are dispatched to do different parts of the project: data processing, create HTML/Javascript, ...



Visualization

Change to a light color scheme with a hand-drawn map



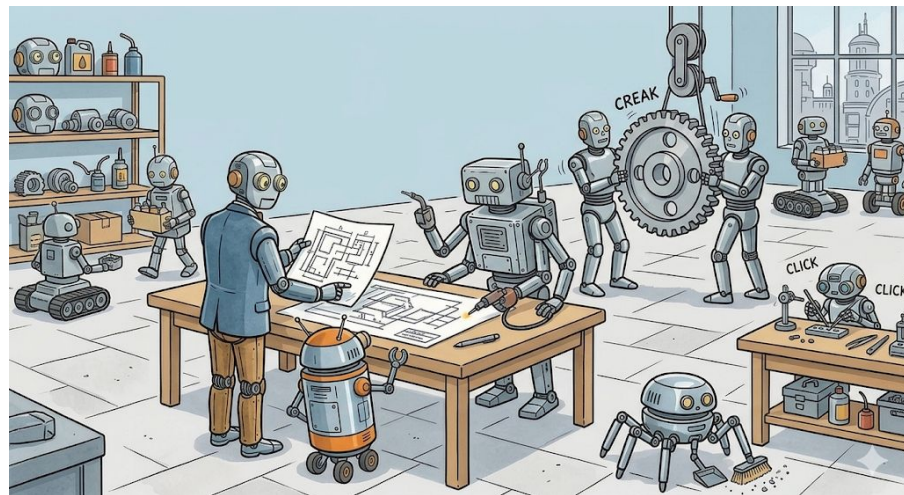
Final thoughts

- Different tasks require levels of “**human in the loop**”
 - **One end:** read all the reports and interact with AI as your collaborator
 - **In between:** advisor-student relationship, where AI runs autonomously for some time but you inspect and steer it from time to time
 - **Other end:** specify the problem and let the AI run for hours / days / ...
- For more complex tasks, you may want to try an agentic IDE like Antigravity instead.
- Human ingenuity is still there, but we may be using different workflows
- In the past, a lot of engineer's time was spent doing calculations and computers freed up time for us to spend in more complex tasks. We hope AI will help focus on creativity on tackling more difficult problems.

The image shows two open pages of a logarithmic table, likely from a technical manual or reference book. The pages are filled with columns of numbers and their corresponding logarithmic values. The title at the top of each page is "LOGARITHMS OF COTANGENTS". The table is organized into columns for numbers 1 through 10, and the rows represent the logarithmic values for those numbers. The text is small and dense, typical of a reference table.

Final thoughts

- Don't assume the systems can't do something. Models are getting better very fast.
- But getting what you want may require many iterations (back-and-forth, corrections, restarts, novel agentic loops) → new kinds of difficulty
- Always check the work!
- How you ask matters a lot. Try different prompt variations and ask Gemini to refine prompt ideas. Try different systems.
- How the research community should evolve in the presence of AI is a big open question.



Thanks!

For more advanced techniques, come to our workshop in the afternoon!

Feedback to {duetting, renatoppl}@google.com
Slides are available at renatoppl.com/ai/tutorial.html