CS398: Computational Education

Want to see something cool?

Conditional Probability (Chris)

pai.stanford.edu/pai1/slides/edit/5oMkefb2g?slide=73de80e5-545c-4b6c-84db-691aaf557846

Open in app

Home

Saved

Present

Conditional Probability (Chris

New Slide | Text | Image | Video | Music | Table | Shape | Demo | Import PPTX | Arrange | Group Write | More

24

25

26

27

28

29

Large Language Models (LLMs) return the probability of the next token, given the prompt.

3 phases

Camera

Prompt

e.g. The capital of France is

Model

Gemini (server)

Temperature: 1.00

Next-Token Probability

Sample

Enter a prompt to see its next-token distribution.

Notes | Script

Click to add notes

New Slide

The Progress Meter Is Not the Learning:
Puzzle Ratings Rise by Hundreds of Points While Solving Ability Barely Moves

Preliminary results · chess.com puzzle log, 2021–22 · `analysis/lantern/` · September 2026

Abstract. We analyse 6.3M puzzle attempts by 12,374 frequent chess.com users (full one-year histories) against a fixed item bank of TrueSkill puzzle difficulties. The number the learner sees—their puzzle rating—rises by a mean of **+449** over the year. Ability measured against the fixed bank rises by roughly **+47** Elo-equivalent, and the two correlate at **r = 0.15** across users. A model-free within-person check confirms it: on puzzles from the same difficulty decile, success in the last third of a learner's history is **−0.9 pp** from the first third, and solve time is unchanged, over a span in which the rating rose +323. The rating rise decomposes into ~400 attempts of slow controller convergence from an underestimate, followed by a persistent drift of +0.15–0.22 per attempt driven partly by a speed bonus and a flat expected-score curve. Secondary results: a split-half design removes the mechanical artifact in the "85% rule" test but the surviving effect is still smaller than its own placebo (null); puzzles are never re-served (repeat designs are unavailable); and difficulty-controlled session residuals show quit-on-a-loss, graded tilt, and no forgetting across 30-day breaks.

1. Data and measurement

Sample. User-hash sample (`user_id mod 251 = 0`) of the 1.556B-row log: 6,287,469 attempts, 12,374 users, median 252 attempts and 232 days per user, 2021-03-12 to 2022-03-11. Users are pre-filtered by the source to 50–5,000 lifetime attempts.

Item bank. TrueSkill puzzle difficulties fit on a separate 10.3M-attempt shard sample (185,859 puzzles; 88% of sample rows covered), converted to logits by $d = (\mu - 25)\cdot 1.702/(\sqrt{2}\cdot\beta)$, $\beta = 25/6$. Difficulties are held **fixed** for the whole year.

Ability. Rasch MLE per user or per window of 50 attempts against the fixed bank. Per-window SE ≈ 0.31 logits. Elo-equivalents use the platform's own item scale (289 Elo/logit from regressing chess.com `difficulty_rating` on d, r=0.755); the textbook 174 would halve every gain reported below.

Residuals. For session analyses, $\hat{p} = \sigma(0.049 + 0.761\cdot(\theta_0 - d))$ with a global calibration; residual = correct − $\hat{p}$. Repeat encounters excluded.

2. Main result

Over the observed year, the platform rating rises +449 (median +386); ability against the fixed bank rises +47 (median +36); across 8,664 users the correlation is r = 0.15. Only 13% of users improve significantly against the bank ($z > 1.96$), 5% get significantly worse, median $|z| = 0.95$.

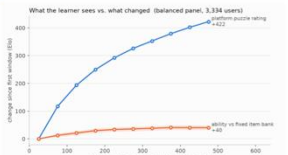


Figure 1. Balanced panel of 3,334 users with ≥10 windows: mean change since the first window in platform rating (blue) and in fixed-bank ability (orange, 95% CI), both in Elo on the platform's item scale.

2.1 A model-free check: same puzzles, same person

Global difficulty deciles; for each user with ≥150 attempts and each decile with ≥5 attempts in both the first and last third of their history, the paired change in success (49k user×decile cells).

decile	p_{first}	p_{last}	Δp	Δ solve-time ratio	Δrating
0 (easiest)	0.797	0.797	−0.000	1.00	+380
1	0.680	0.697	+0.017	1.00	+352
3	0.607	0.611	+0.004	0.97	+329
5	0.560	0.546	−0.014	0.99	+310
7	0.505	0.475	−0.030	1.00	+298
9 (hardest)	0.404	0.355	−0.050	1.00	+279
weighted	**0.585**	**0.576**	**−0.009**	**≈1.00**	**+323**

Table 1. Within-person change on same-difficulty puzzles vs. platform rating change over the same span. Had the +323 been ability, success would have gone 0.585 → 0.891. The negative Δp on hard deciles is consistent with selection (early hard puzzles are served only during hot streaks; later they are routine), so the honest reading is "at most +1–2 pp on easy/medium content."

2.2 Where the +449 comes from

(a) Slow convergence. 32% of users start at the default rating (−400). The platform's per-attempt step is small and constant (median $|\Delta| = 11$ over a user's first 10 attempts, 10.5 over the last 10 — no provisional phase). Learners are served easy puzzles, succeed at 75%, and are ratcheted up for hundreds of attempts.

attempt ordinal	start −400	start 700–1200	start 1200+	success (−400)
0–24	+5.48	+2.49	+1.12	0.753
50–99	+4.15	+2.06	+0.99	0.700
100–199	+2.37	+1.34	+0.72	0.635
200–399	+1.01	+0.66	+0.45	0.583
400–799	+0.39	+0.33	+0.27	0.555
800–1599	+0.19	+0.19	+0.16	0.549
1600+	+0.07	+0.12	+0.12	0.532

Table 2. Mean `rating_change` per attempt by attempt ordinal and starting-rating class, and the observed success rate. The ramp has the same shape for every starting class, including users who begin above 1200.

(b) Post-convergence drift. Restricting to attempts *after* a user's 400th (2,509 users with ≥300 further attempts; ~113 days between thirds) the rating still rises while accuracy does not:

Practice cools the learner: a thermodynamic description of skill acquisition in people and machines

Draft for a general-science venue, 17 September 2026. Data: 1.56 billion chess.com puzzle attempts; 154 pretraining checkpoints of three language models.

How a learner changes with practice is usually described by a learning curve: performance against time. Physics describes a driven system differently, by how it *responds* to a small kick and how it *fluctuates* on its own, as functions of two times—when the kick lands and when its effect is read. These two-time functions carry information a learning curve provably cannot: whether the system remembers its own age, and whether its fluctuations and responses share a temperature. Here we show that a learner on an adaptive platform is, exactly, a thermostatted Fermi system—the standard psychometric model of success against difficulty is the Fermi–Dirac distribution, and the platform's rating tracker is an integral thermostat—and we use that identity to measure both two-time functions for 3.1 million people over 1.4 billion randomized perturbations and for a family of language models over 154 stages of pretraining. Neither learner ages. Both cool: the sharpness with which success depends on difficulty, the learner's inverse temperature, triples over a person's first thousand attempts and rises as a power law of comparable exponent over a transformer's pretraining. And we prove that the thermostat repays every perturbation, so the question the adaptive-learning literature has asked for forty years—how much does a harder problem teach?—has no first-order answer on any platform that holds success at a set point. What such platforms can identify, and how to design them so they can, follows from the same theorem.

Why this matters

Adaptive platforms now mediate a large share of deliberate practice—languages, mathematics, music, chess—and the training of neural networks, where an adaptive sampler plays the tutor. All do the same thing: measure the learner, then choose the next problem to keep success near a target. Two questions follow. What does one problem do to everything the learner does afterwards? And what kind of physical object is a learner—does it relax like a spring, age like a glass, or fluctuate like a stationary noisy system? The first is the practical heart of tutoring. The second has been argued for a century through the *shape* of learning curves, without resolution, because a power-law curve is provably a mixture of exponential learners with different rates (Bernstein's theorem) and the data cannot tell them apart.

Physics answers both with two-time functions, and they are measurable here because of an accident: the platform's published difficulty of each problem is wrong by an amount the platform itself does not know. Every attempt therefore delivers a random kick—107 Elo on average, 1.4×10^9 times—whose effect on all subsequent behaviour can be read as a Green's function. The mathematics of this paper is what such a closed loop can and cannot reveal; the empirical content is the first measurement of a human learner's response and fluctuation functions, and of the same functions for a transformer.

Three exact identities

The learner is a Fermi system. $P(\text{solve}) = (1 + e^{\beta(d-\theta)})^{-1}$ is the Fermi–Dirac occupation of a level d by a reservoir at chemical potential θ and temperature $1/\beta$ (Fig. 1A). Ability is a chemical

Calibrated to Measure, Not to Teach:
Difficulty, Failure, and Disengagement in a Billion Chess Puzzles

Working draft — 1.556 billion chess.com tactical-puzzle attempts, 2021–2022. All figures and statistics reproducible from the analysis pipeline.

Abstract. We analyze 1.556 billion tactical-puzzle attempts from chess.com — to our knowledge the largest record of human skill practice ever examined — to ask whether an adaptive practice system optimizes difficulty for *learning* or for *measurement*. The platform serves problems at the success rate that maximizes psychometric information (≈50%), not the higher rate claimed to maximize learning. Its published difficulty and ability ratings do not compose into calibrated outcome probabilities. We trace disengagement to its microdynamics — learners leave on honest failure, the stop hazard is an asymmetric U in the win/loss streak, and the fine-scale temporal volatility of served difficulty predicts departure beyond mean difficulty or success. Two memory results: tactical skill shows no decay across 1–3 month absences, and returns from long absences improved.

1. Data and difficulty measurement

The corpus is one year of rated puzzle attempts, filtered to learners with 50–5,000 lifetime attempts. Each attempt records learner, puzzle, outcome, solve time, the learner's live rating, and a timestamp. Because shards are ~45-minute global time-slices, sequence analysis uses a by-user sample (12,374 learners, median 252 attempts over 232 days).

We do not trust the platform's published ratings. Four independent estimators — Rasch, Glicko-2, TrueSkill, a Kalman tracker — agree with one another at $r = 0.96$–0.997 but with chess.com's published difficulty at only 0.52–0.77. On held-out prediction every online tracker beats static Rasch, evidence that ability is non-stationary. Critically, the published item and learner ratings *do not compose*: through the textbook Elo formula they predict worse than a constant, and the implied success curve is non-monotone — puzzles rated far above a learner are solved *more* often than puzzles rated slightly above (Fig. 1). Substituting our own estimate on either side removes the anomaly. The platform's *internal* update rule, recovered from observed rating changes, is by contrast well calibrated: the published number is simply not the quantity the server uses.

Fig. 1. Actual success vs. the offset of chess.com's published ratings. The curve is far flatter than textbook Elo predicts and turns around past +150 — harder-rated puzzles solved more often. No rating model produces this; both published scales are noisy.

2. The platform serves at the measurement optimum

Item information is maximized where success probability is 0.5. The learning literature places the learning optimum higher (the '85% rule'). Across 83,573 windows of 50 attempts, chess.com serves at mean success **0.566** (median 0.540), and its own internal model's 50% point sits at essentially the difficulty it most often serves. Empirically the system behaves like an adaptive *test*, not an adaptive *tutor*.

3. Disengagement is failure-driven, with structure

Around every ≥7-day break, accuracy drifts ~3 points below the learner's baseline over 40 attempts, then plunges: the last attempt before a break has accuracy 0.49, while served difficulty *rises* into the break. Three refinements: (i) it is **honest failure, not tilt** — fast-guessing barely rises into the break; (ii) the **stop hazard is an asymmetric U** in the live streak, minimized just after one win, worse for losing than winning runs, surviving session-position controls (Fig. 2); (iii) **fine-scale volatility predicts departure** — in a Haar wavelet decomposition, disengagement risk loads on the finest scales (item-to-item, z = 7.6) and is null at slow drift, holding success and mean difficulty fixed. Total variance cannot see this; the spectrum can.

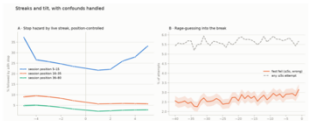


Fig. 2. Stop hazard by live win/loss streak, within session-position bands (left): an asymmetric U, minimized at +1. Rage-guessing into the break (right) barely moves — the accuracy cliff is honest failure.

These connect a chain: serving at the information optimum maximizes failure frequency; failure drives the stop hazard; the served difficulty's fine structure predicts who leaves. The measurement-optimal policy is plausibly persistence-pessimal. (One appealing intervention is *not* supported: how a session ends predicts whether a learner stops, but not how long they stay away.)

4. Two memory results

The platform essentially never re-serves a puzzle (74 repeats in 6.3M), so every attempt is novel and improvement is never recall. Using natural absences as spacing intervals: tactical skill shows **no measurable decay** across 1–3 month gaps — learners return at baseline, instantly, no warm-up — unlike declarative forgetting curves. After 90–400 day absences they return **+4 points above baseline**. The improvement is solid; its mechanism (consolidation, off-platform practice, selective return) is open.

5. A measurement trap

The natural way to ask 'does harder practice cause more learning' — window a history, regress ability gain on experienced difficulty — is mechanically biased, because both are estimated from the same responses and share noise. A within-person fixed-effects estimate looked clean ($z = -12.9$), but a pre-trend placebo showed the relationship running *backward in time*, with a coefficient seven times larger at a lag where causation is impossible. The fix is to measure the outcome on disjoint content — transfer to un-practiced themes.

Previously on CS398

[experience]

Chess.com prediction

Learning goals:

Understand the task

Explore the landscape of solutions

Make real predictions

Three sister problems

Item Response Theory

Infer what a student knows while taking a test.

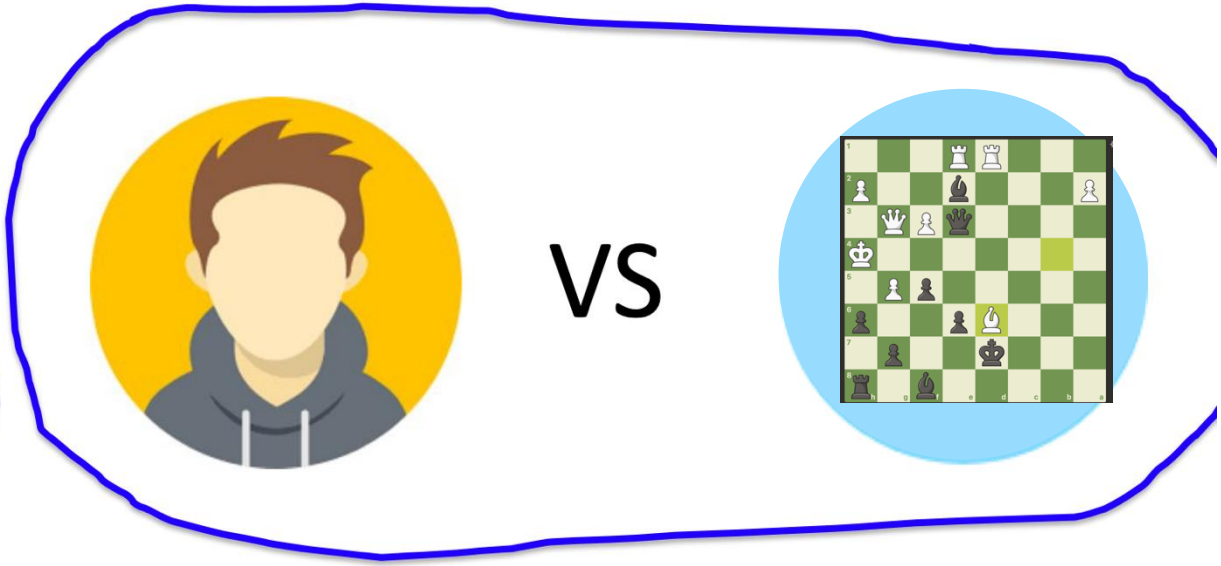
Knowledge Tracing

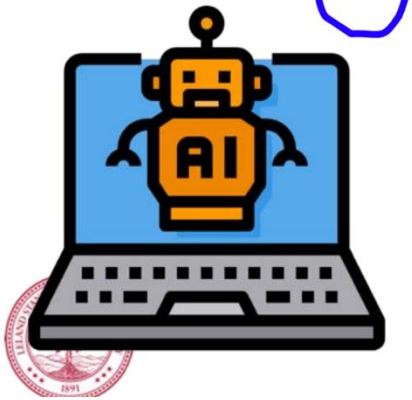
Infer what a students knows while they learn.

Computer Adaptive Testing

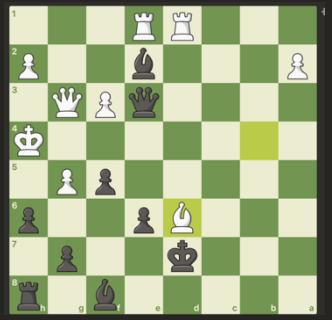
Adaptively chose what to ask students next.

Item Response Theory





A **model** which gives a probability that a particular student will answer a particular question correctly



Solved

Excellent play! You solved it!

Incorrect

That was a good attempt, but once I go here you don't have a strong way to respond.

Show Follow-Up

History of my puzzles

Representation of my ability

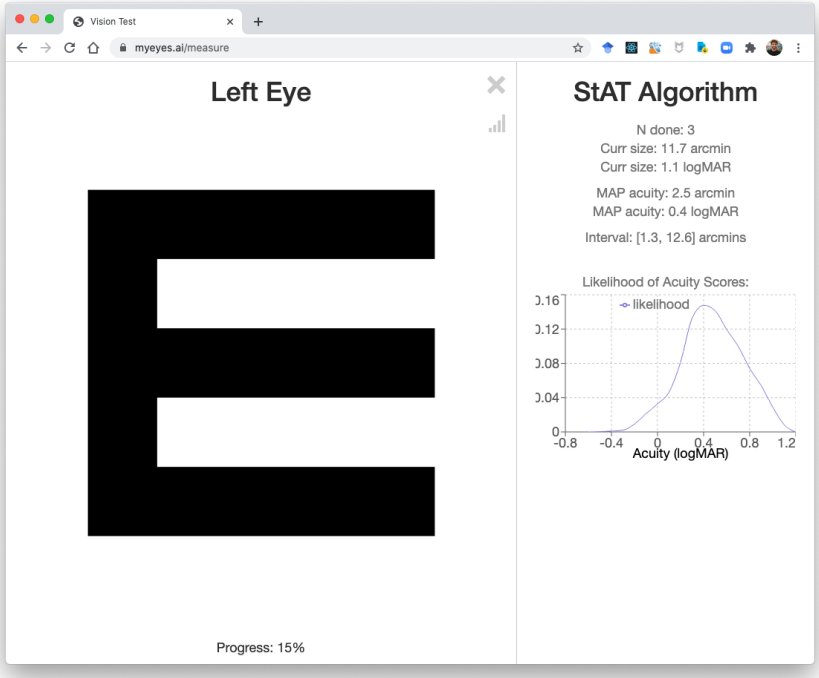
Predict if I will get the next puzzle correct

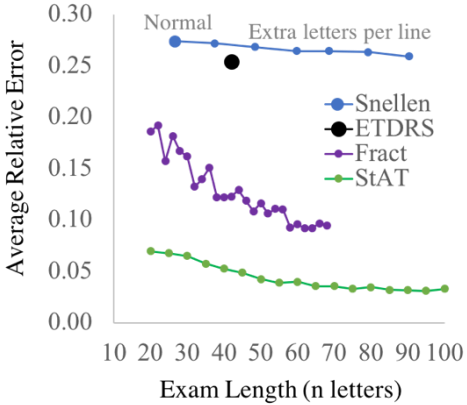
P(correct) = 0.92

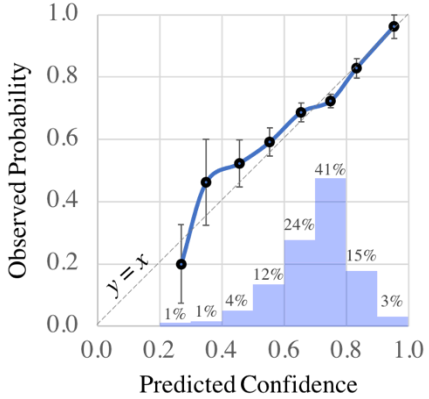

achieve more
SAT
ACT
ETS
GRE
ETS
TOEFL

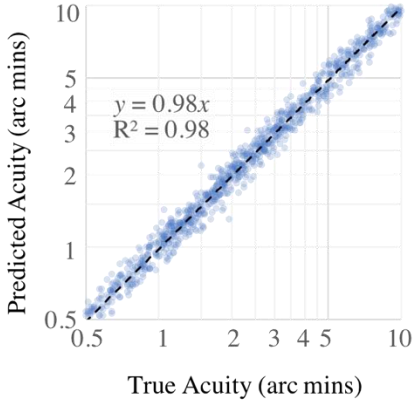
Better Eye Exam

Jan 2020, With a former CS109 student, Ali Malik









Math question: Estimate a continuous valued number.
Get to run noisy experiments of your choosing.


tinder

Leaderboard

Rank	Team	Best model	Best accuracy	Submissions
1	AMF	DKT + IRT head (GRU-64)	65.6%	3
2	Big E	DKT	65.2%	1
3	rube_goldberg	irt	64.5%	1
4	peach	Kalman filter	63.5%	3

Chess.com (both official and replicated) scored 57%

Grade

Participation: 30%

Personal Growth: 30%

Final Project: 40%

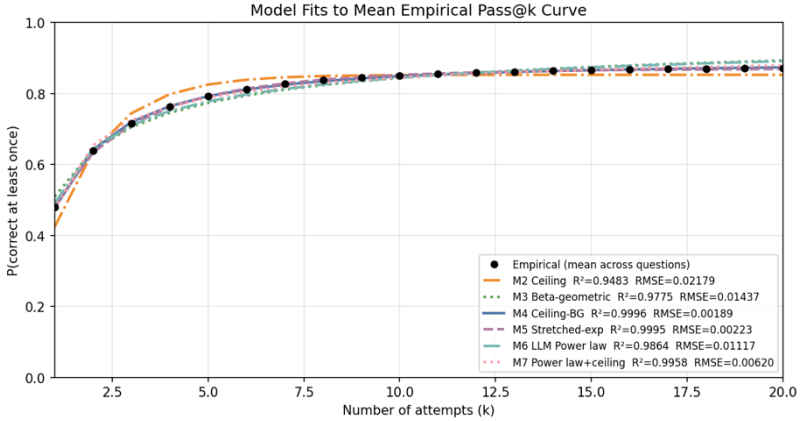
* Time to do assignments during class

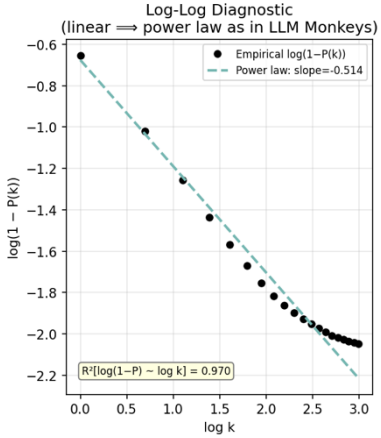
Lets Vibe!

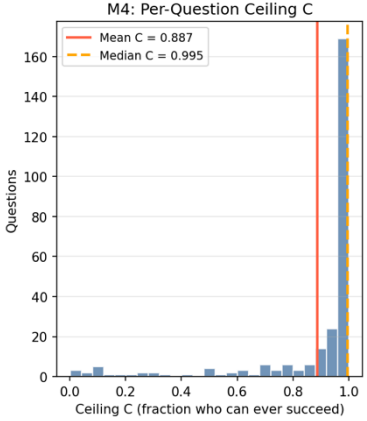

* Welcome to Claude Code
CLAUDE CODE
Press Enter to continue

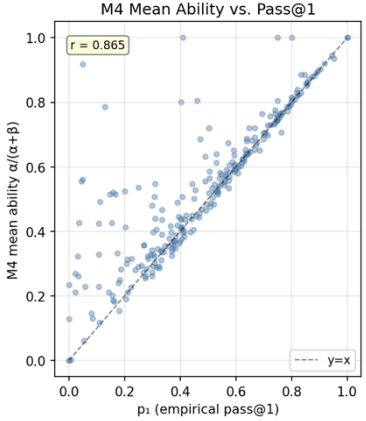
Large Language Monkeys

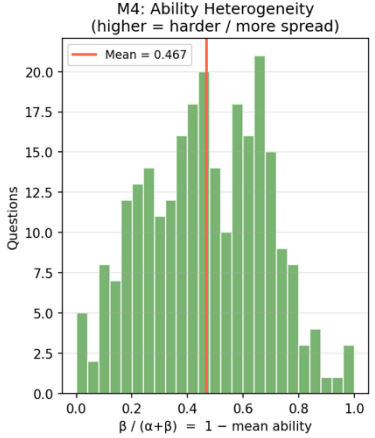
Student Reattempts: Model Fitting





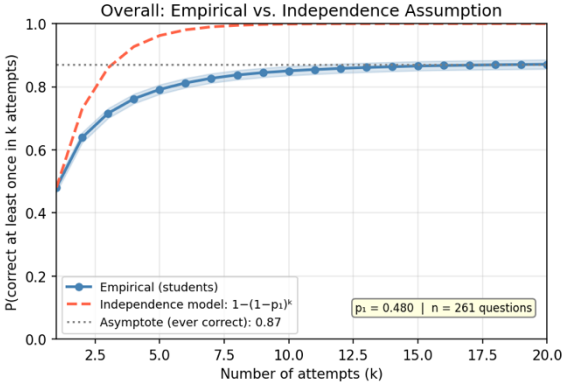


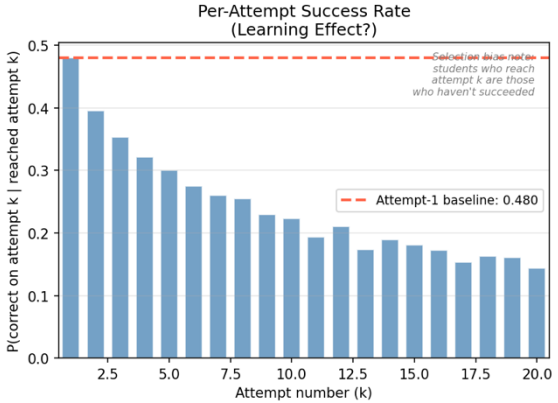


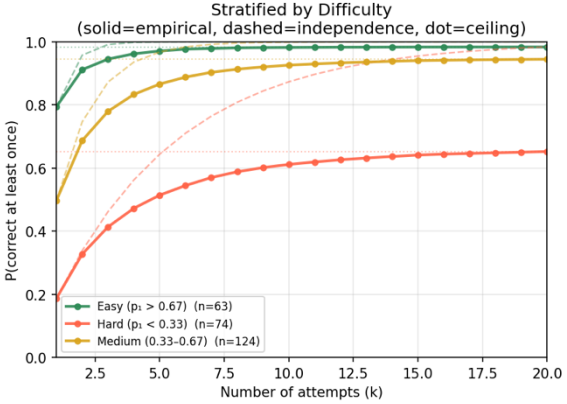


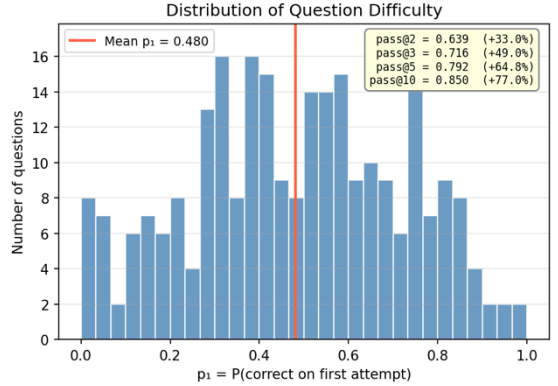
Piech

Student Monkeys: How Reattempts Increase P(Correct)









Seasons of Code.org

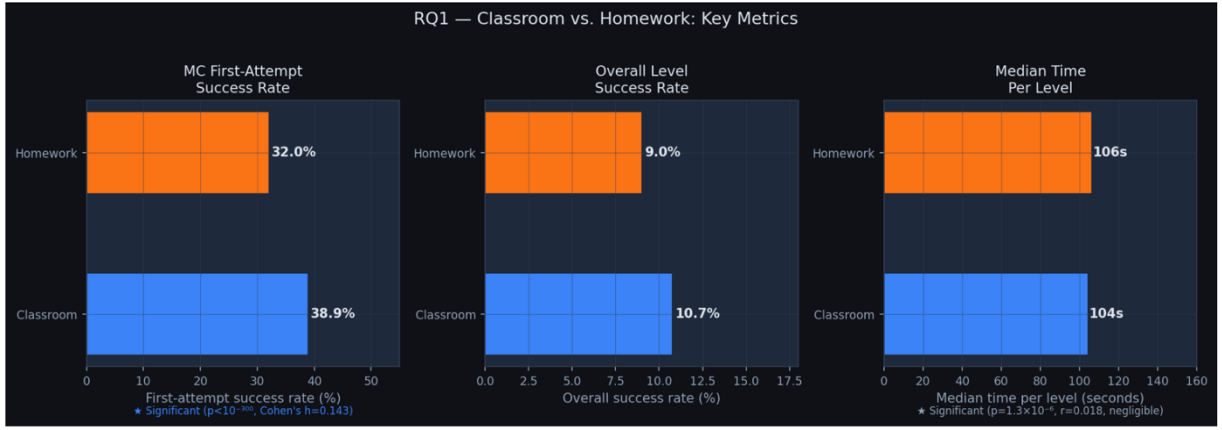
School Context & Student Performance

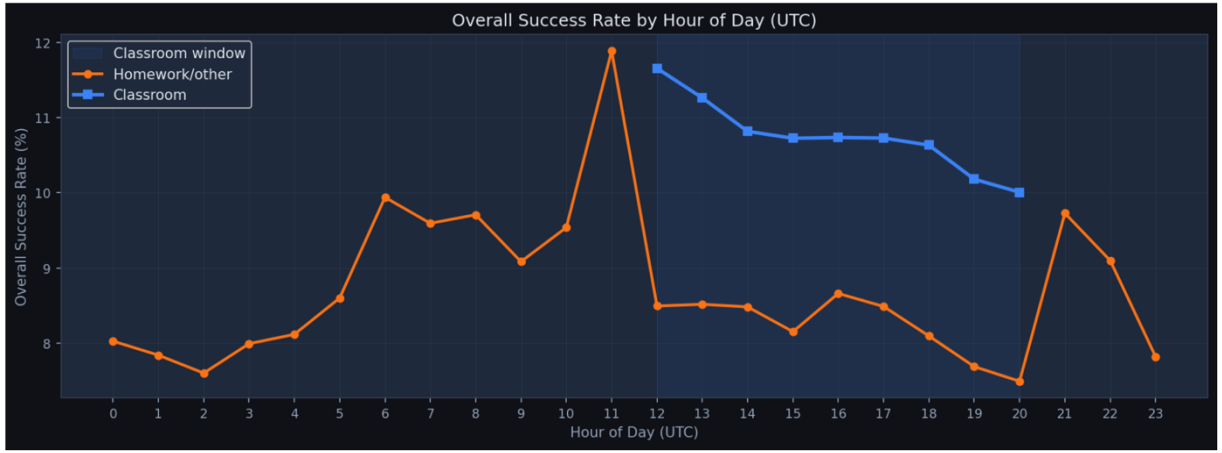
Code.org CSD3 · 2023–2025 Dataset · Temporal Context Analysis

Analysis date: 2026-05-19
Total records: 32,759,915
Dataset: Code.org Computer Science Discoveries Unit 3 (CSD3)

Table of Contents

Piech


RQ1 — Classroom vs. Homework: Key Metrics
MC First-Attempt Success Rate
Homework
32.0%
Classroom
38.9%
0 10 20 30 40 50
First-attempt success rate (%)
★ Significant (p<10⁻³⁰⁰, Cohen's h=0.143)
Overall Level Success Rate
Homework
9.0%
Classroom
10.7%
0.0 2.5 5.0 7.5 10.0 12.5 15.0 17.5
Overall success rate (%)
Median Time Per Level
Homework
106s
Classroom
104s
0 20 40 60 80 100 120 140 160
Median time per level (seconds)
★ Significant (p=1.3×10⁻⁶, r=0.018, negligible)


Overall Success Rate by Hour of Day (UTC)
Classroom window
Homework/other
Classroom
Overall Success Rate (%)
12
11
10
9
8
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
Hour of Day (UTC)

Piech

How did we get here?

Andrej Karpathy
@karpathy

There's a new kind of coding I call "vibe coding", where you fully give in to the vibes, embrace exponentials, and forget that the code even exists. It's possible because the LLMs (e.g. Cursor Composer w Sonnet) are getting too good. Also I just talk to Composer with SuperWhisper so I barely even touch the keyboard. I ask for the dumbest things like "decrease the padding on the sidebar by half" because I'm too lazy to find it. I "Accept All" always, I don't read the diffs anymore. When I get error messages I just copy paste them in with no comment, usually that fixes it. The code grows beyond my usual comprehension, I'd have to really read through it for a while. Sometimes the LLMs can't fix a bug so I just work around it or ask for random changes until it goes away. It's not too bad for throwaway weekend projects, but still quite amusing. I'm building a project or webapp, but it's not really coding - I just see stuff, say stuff, run stuff, and copy paste stuff, and it mostly works.

3:17 PM · Feb 2, 2025 · **7.1M** Views

PhD advised by Fei Fei

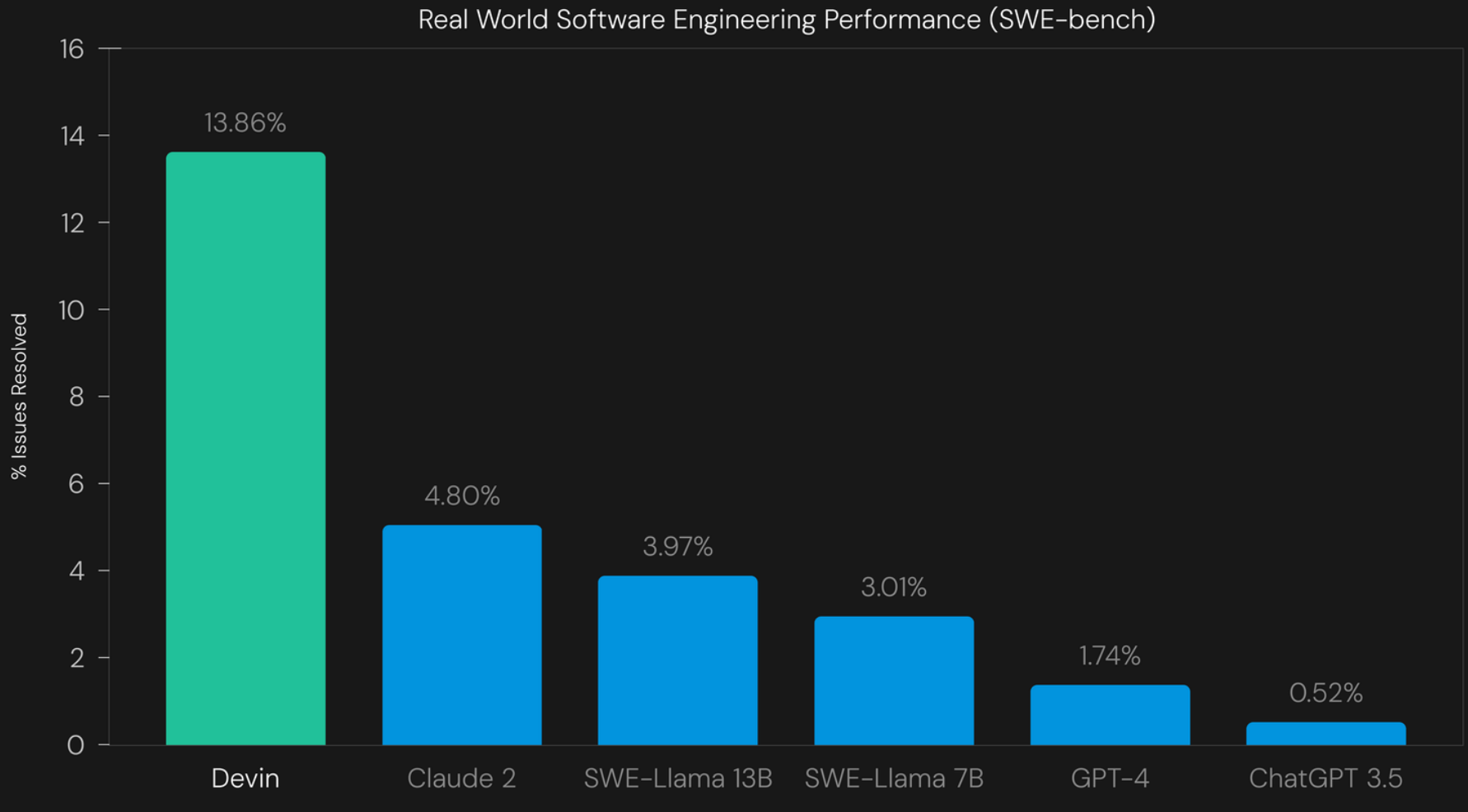
SWE-bench

Leaderboard

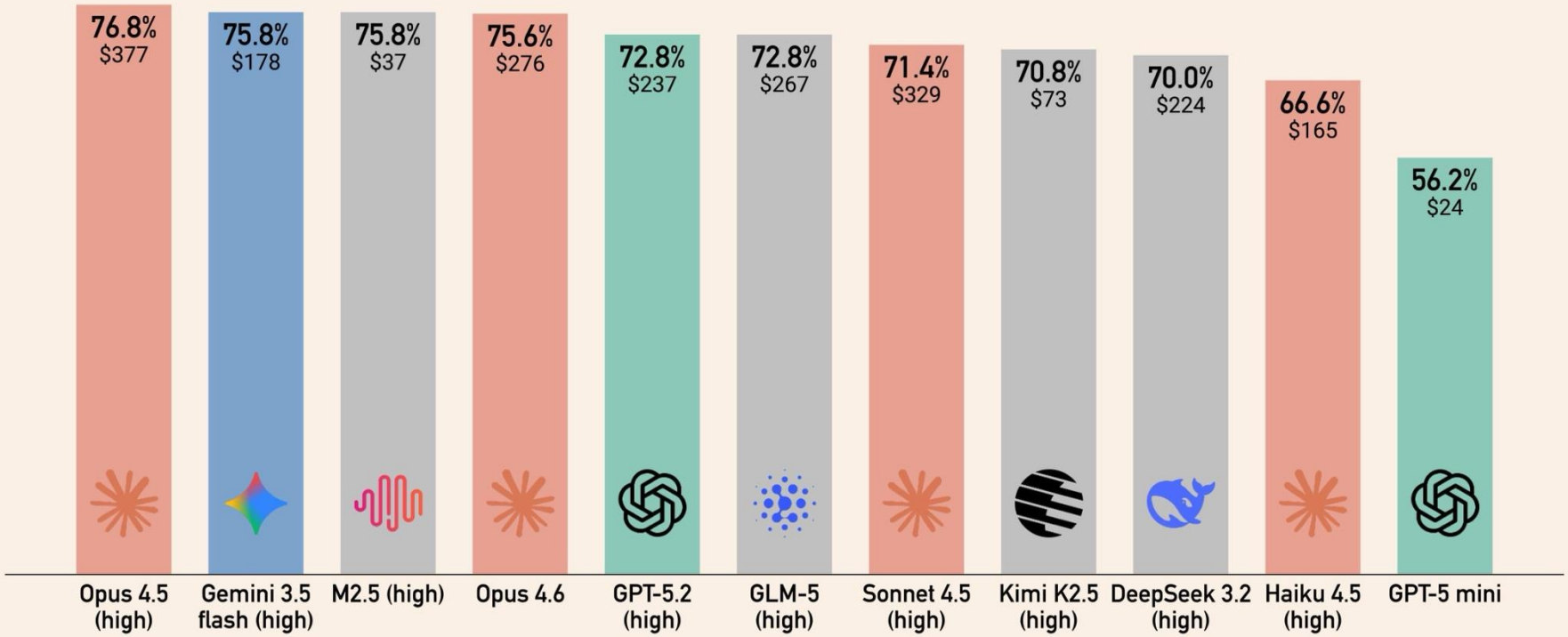
Lite | Verified | Full

Model	% Resolved	Org	Date	Logs	Trajs	Site
Agentless-1.5 + GPT 4o (2024-05-13)	38.80		2024-10-28	✓	✓	
AutoCodeRover (v20240620) + GPT 4o (2024-05-13)	38.40		2024-06-28	✓	-	
SWE-agent + Claude 3.5 Sonnet	33.60		2024-06-20	✓	✓	-
nFactorial (2024-10-07)	31.60		2024-10-07	✓	✓	

2,294 problems sourced from GitHub issues and their corresponding pull requests across 12 popular Python repositories.

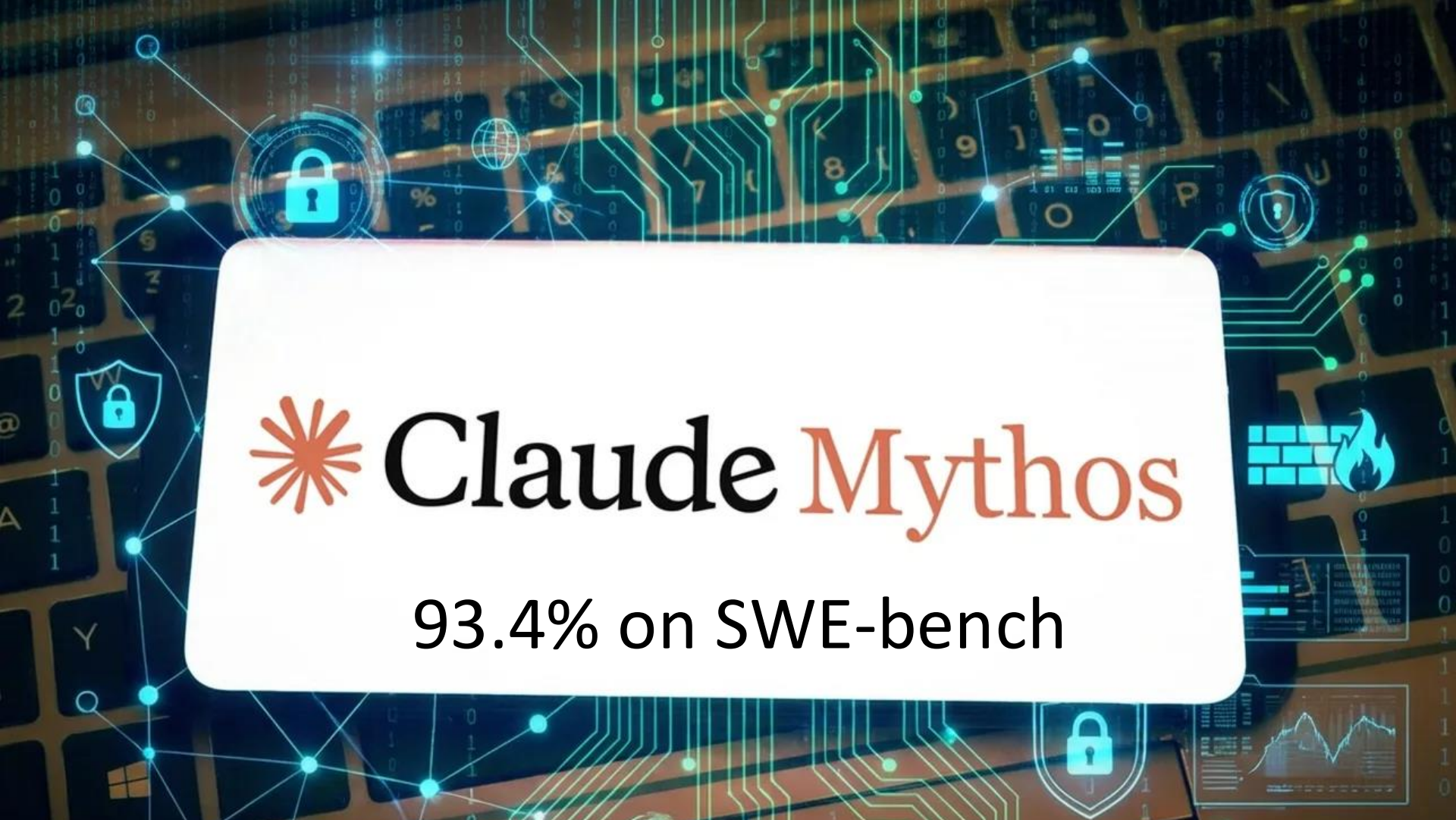

Real World Software Engineering Performance (SWE-bench)
% Issues Resolved
16
14
12
10
8
6
4
2
0
13.86%
4.80%
3.97%
3.01%
1.74%
0.52%
Devin
Claude 2
SWE-Llama 13B
SWE-Llama 7B
GPT-4
ChatGPT 3.5

Official SWE-bench Verified leaderboard



Evaluated with mini-swe-agent v2

mini-swe-agent.com | swebench.com


Claude Mythos
93.4% on SWE-bench


Gemini Advanced
1M
context window
now available
Gemini Advanced
Hello, Elisa
How can I help you today?
Generate a futuristic image
Help me write HTML, CSS, and JS

Good Vibes

Or: How good of a manager are you?

of junior coders

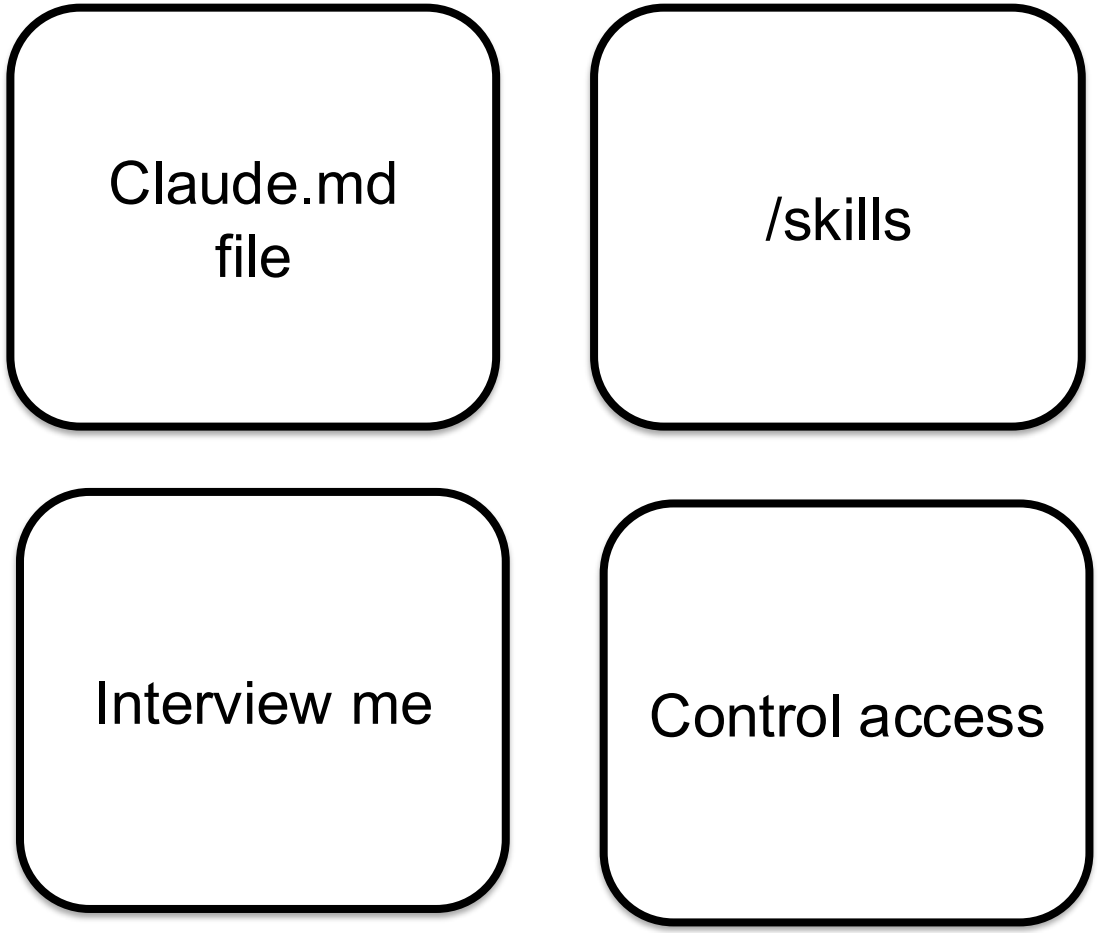
intelivita

Software Development Best Practices

1. Define Requirements Clearly
2. Choose The Right Methodology
3. Utilize Frameworks
4. Document Software Code Properly
5. Take Advantage of Issue Tracker
6. Manage with Version Control
7. Do Thorough Unit Tests
8. Use Sandbox Environment



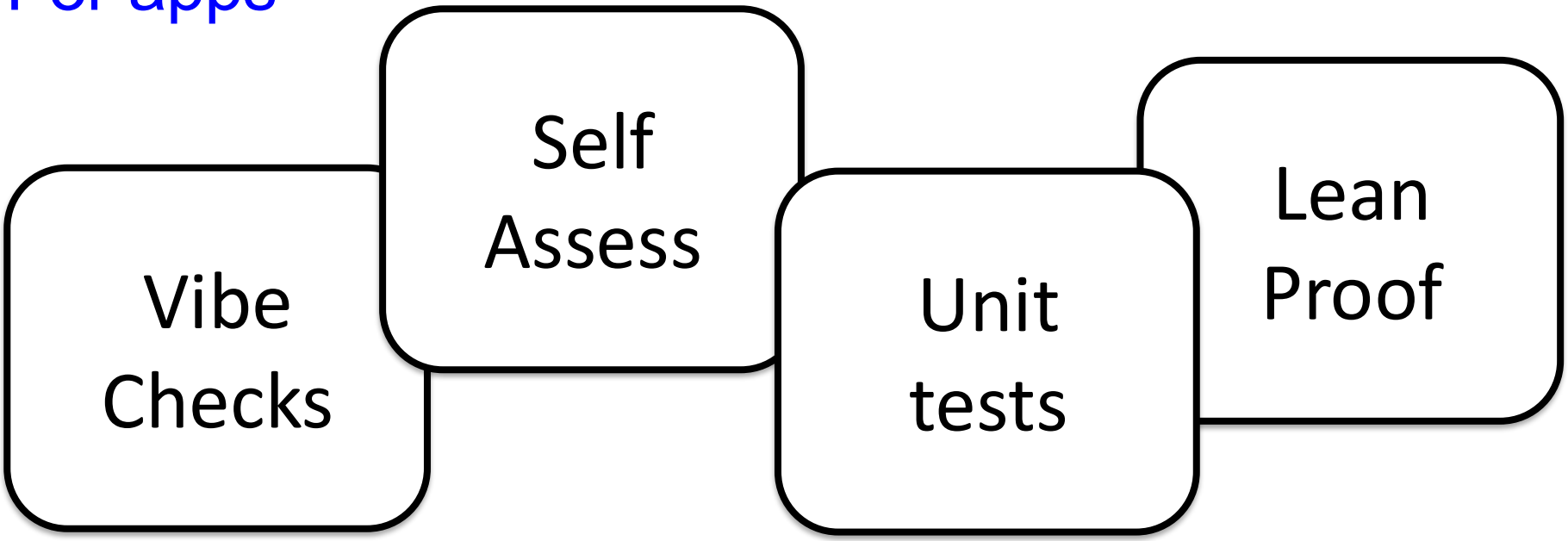

Piech



* These will change in the next 6 months (with high probability)

Evaluate
Evaluate
Evaluate

For apps



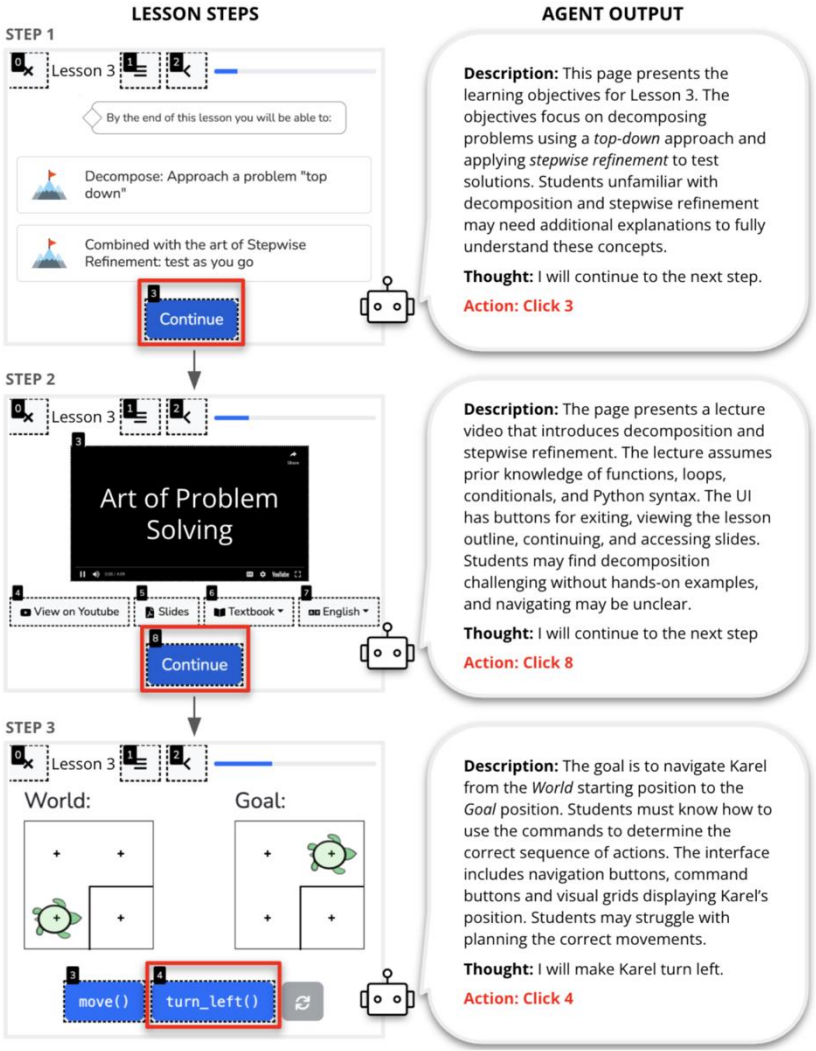
Ask: run the program and make a video

Ask: does this code have any bugs / vulnerabilities?

Ask: write unit tests and end to end tests

Ask: prove that the code works

Piech

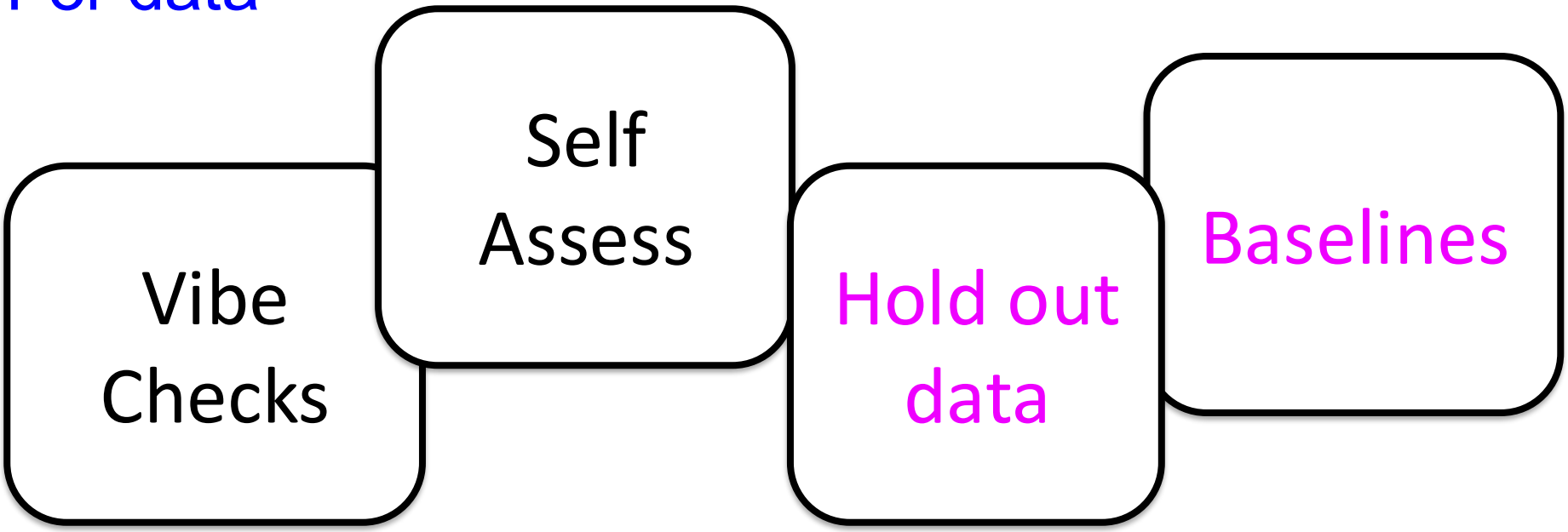


AI Agents Interact with Course and Gain Insights

Dropout Prediction Method	Error (↓)
Baseline Methods	
Sample	0.114
Regression	0.176
LLM-based Methods	
Target Lesson	0.116 ± 0.003
+ Previous Lesson	0.115 ± 0.004
+ Transfer Data	0.100 ± 0.002
+ **Agent**	**0.060 ± 0.003**

Piech

For data



Ask: do the results make sense?

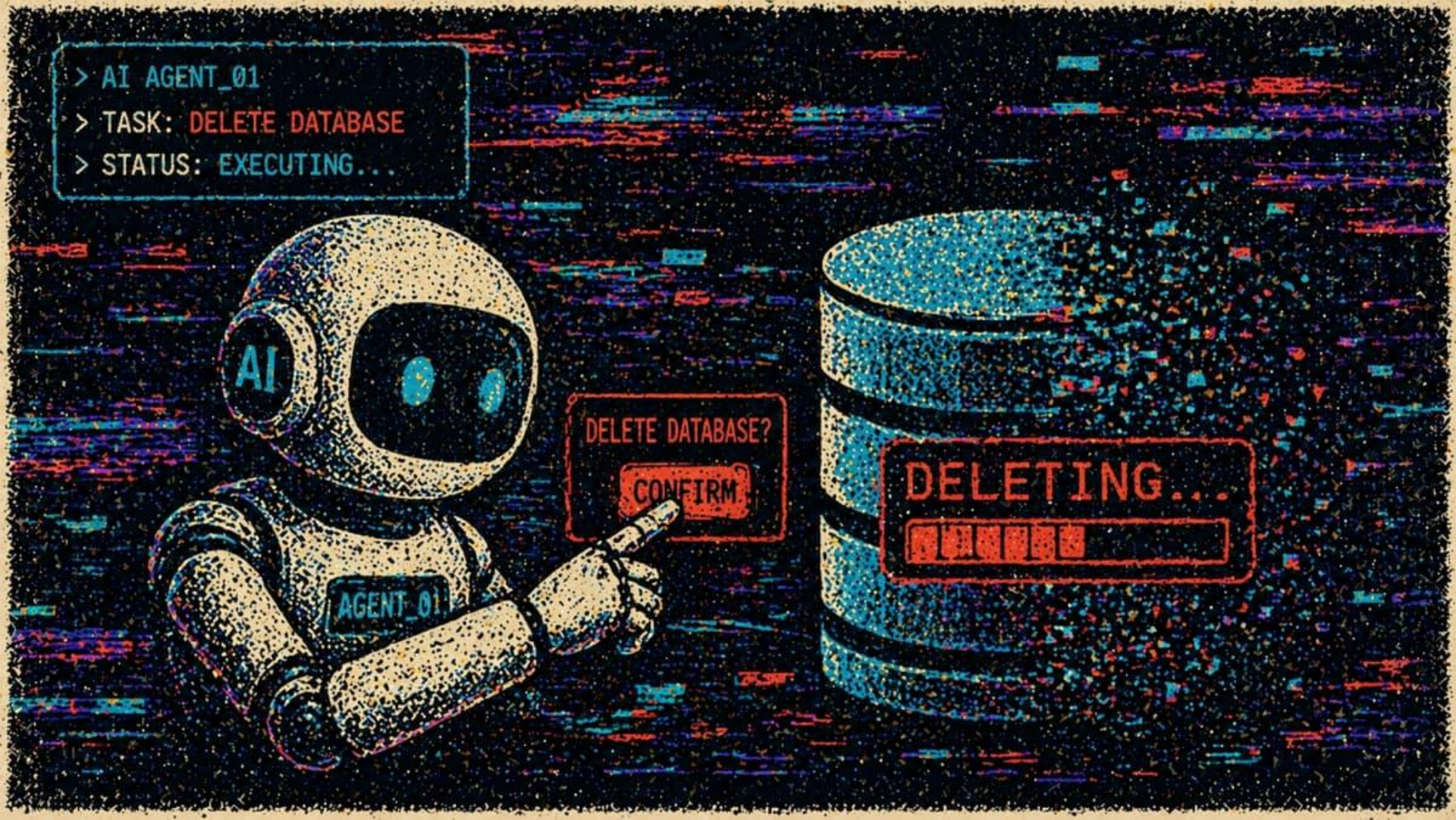
Ask: does this analysis have any **limitations**?

Ask: split data into train/test/eval

Ask: try baselines

Piech

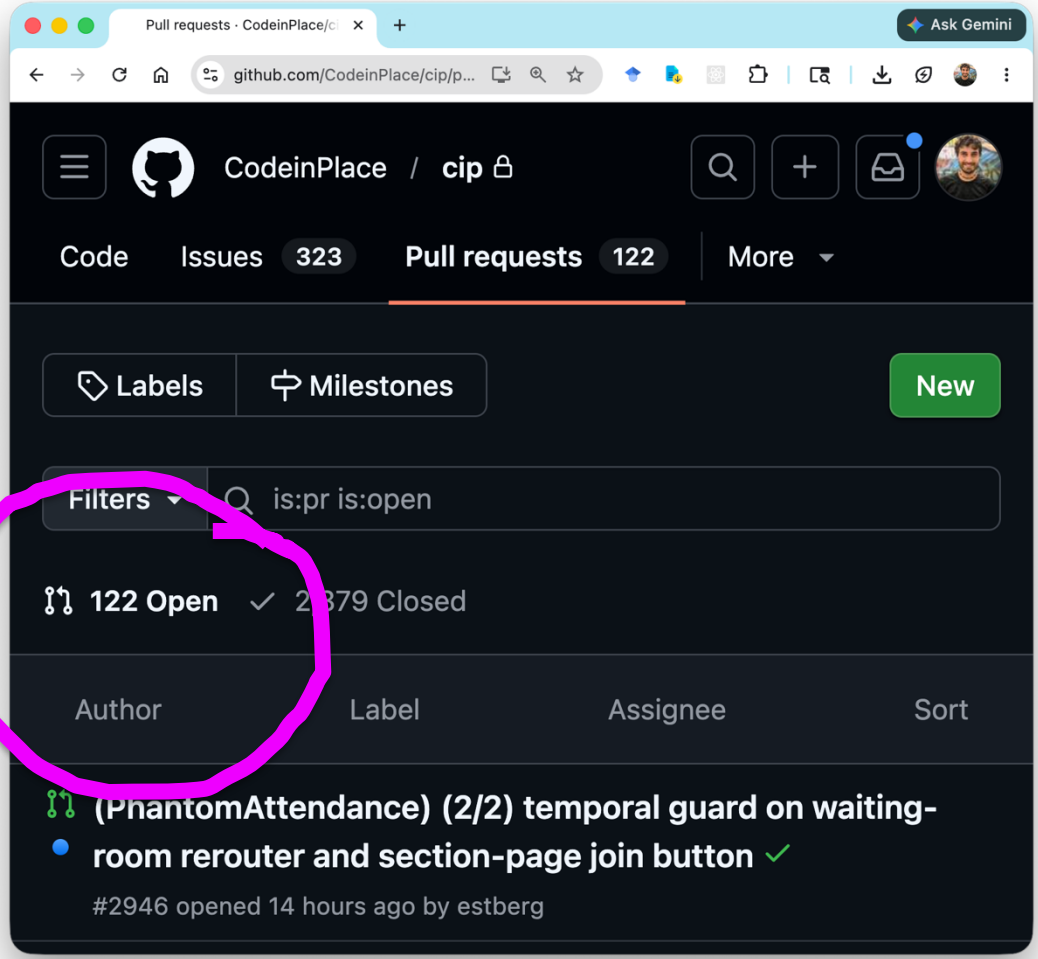
Bad Vibes


> AI AGENT_01
> TASK: DELETE DATABASE
> STATUS: EXECUTING...
AI
AGENT_01
DELETE DATABASE?
CONFIRM
DELETING...

Samsung worker fed company secrets to ChatGPT so it could make a presentation


Awni Hannun @awnihannun · 14h
Adopting Claude speak in my regular life, episode 1:
Partner: Did you do the dishes tonight?
Me: Yes they're done.
Partner: Why are they still dirty?
Me: You're right to push back. I didn't actually do them.
298
2.9K
40K
1.2M

Piech


Pull requests · CodeinPlace/c
Ask Gemini
github.com/CodeinPlace/cip/p...
CodeinPlace / cip
Code
Issues 323
Pull requests 122
More
Labels
Milestones
New
Filters
is:pr is:open
122 Open
Closed
Author
Label
Assignee
Sort
(PhantomAttendance) (2/2) temporal guard on waiting-room rerouter and section-page join button
#2946 opened 14 hours ago by estberg

Piech

PAI: Probability is the language of modern AI

Made for you (and for me too)

Code in Place: Massive Course to Teach Teachers

10x the retention of the equivalent online course

99% teacher completion rate

9.7 average recommender strength

Hundreds of jobs found

20+ experiments in future of education

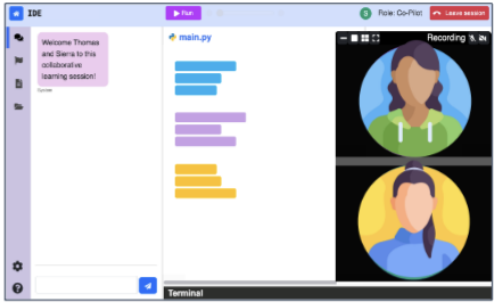
Just in Time Teaching

1

A volunteer teacher has 20 mins of spare time and initiates a 1:1 *ImprompTeach* session

2

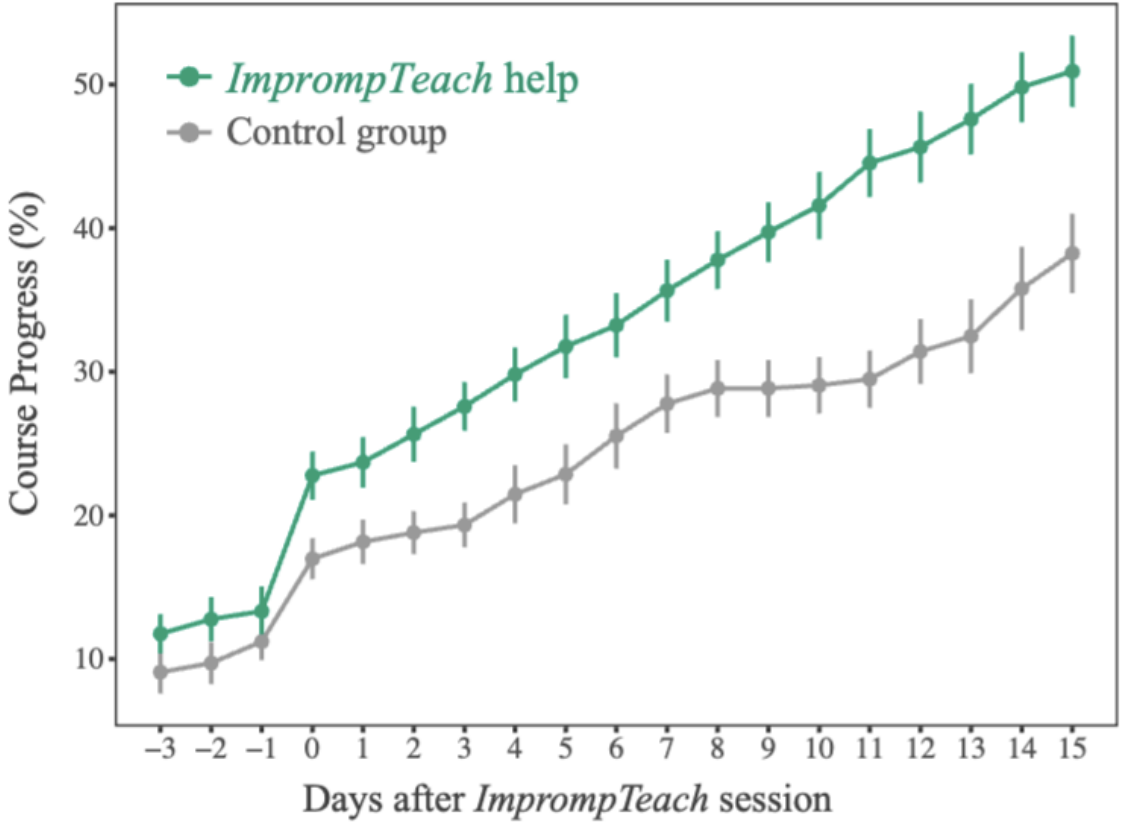
Our system finds students who are currently working on an assignment and offers them 1:1 help one at a time, until someone accepts.

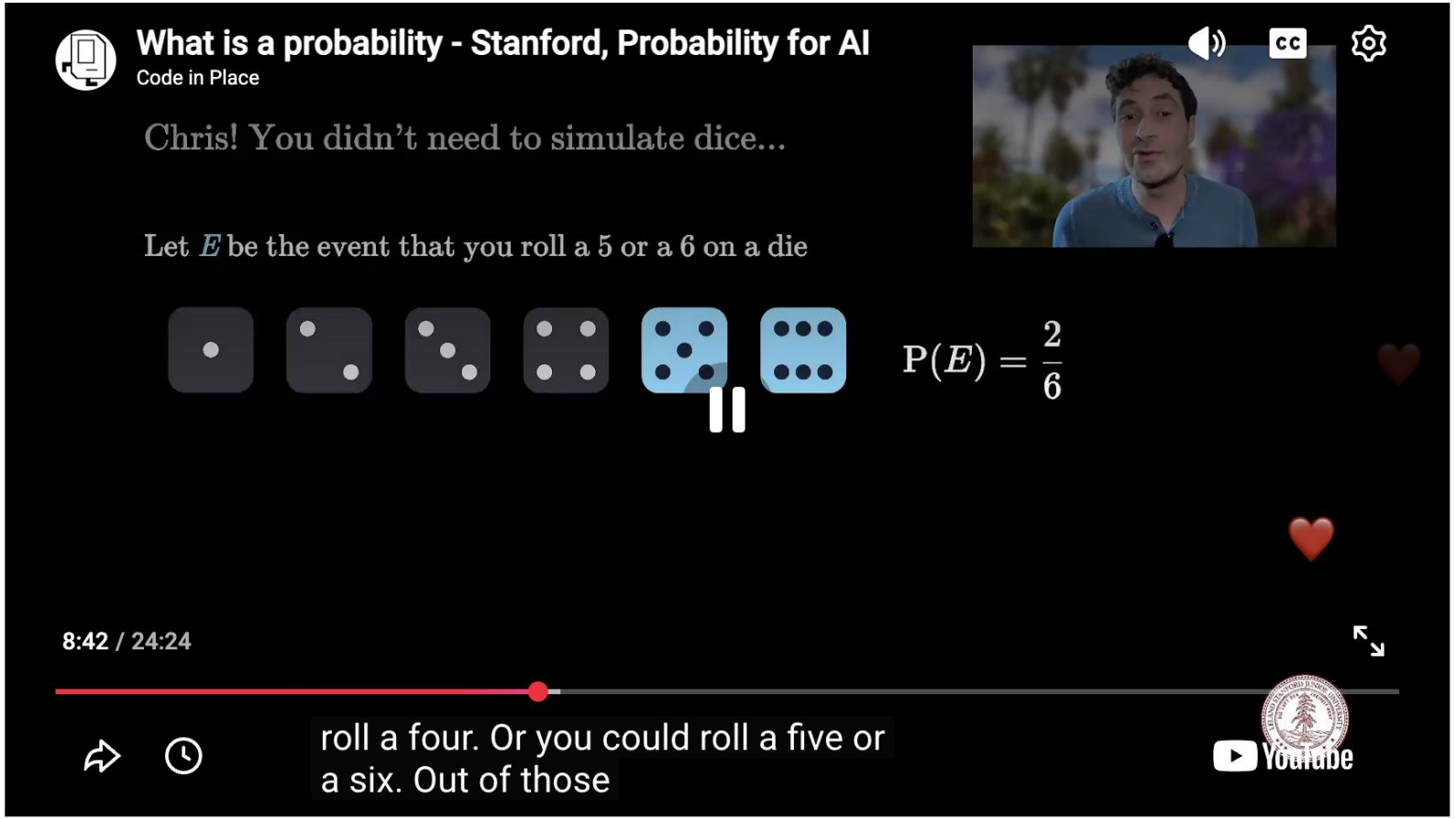


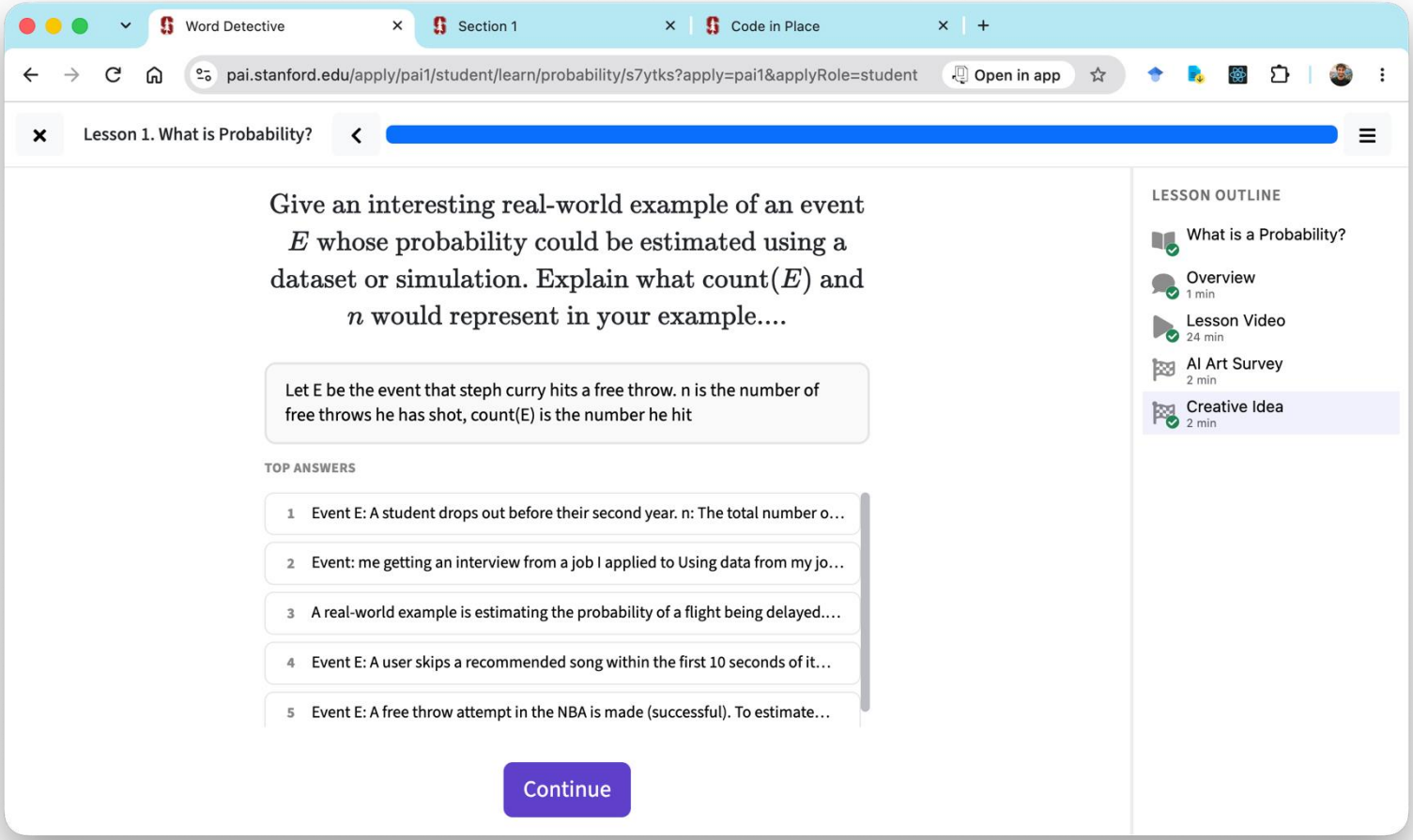
3

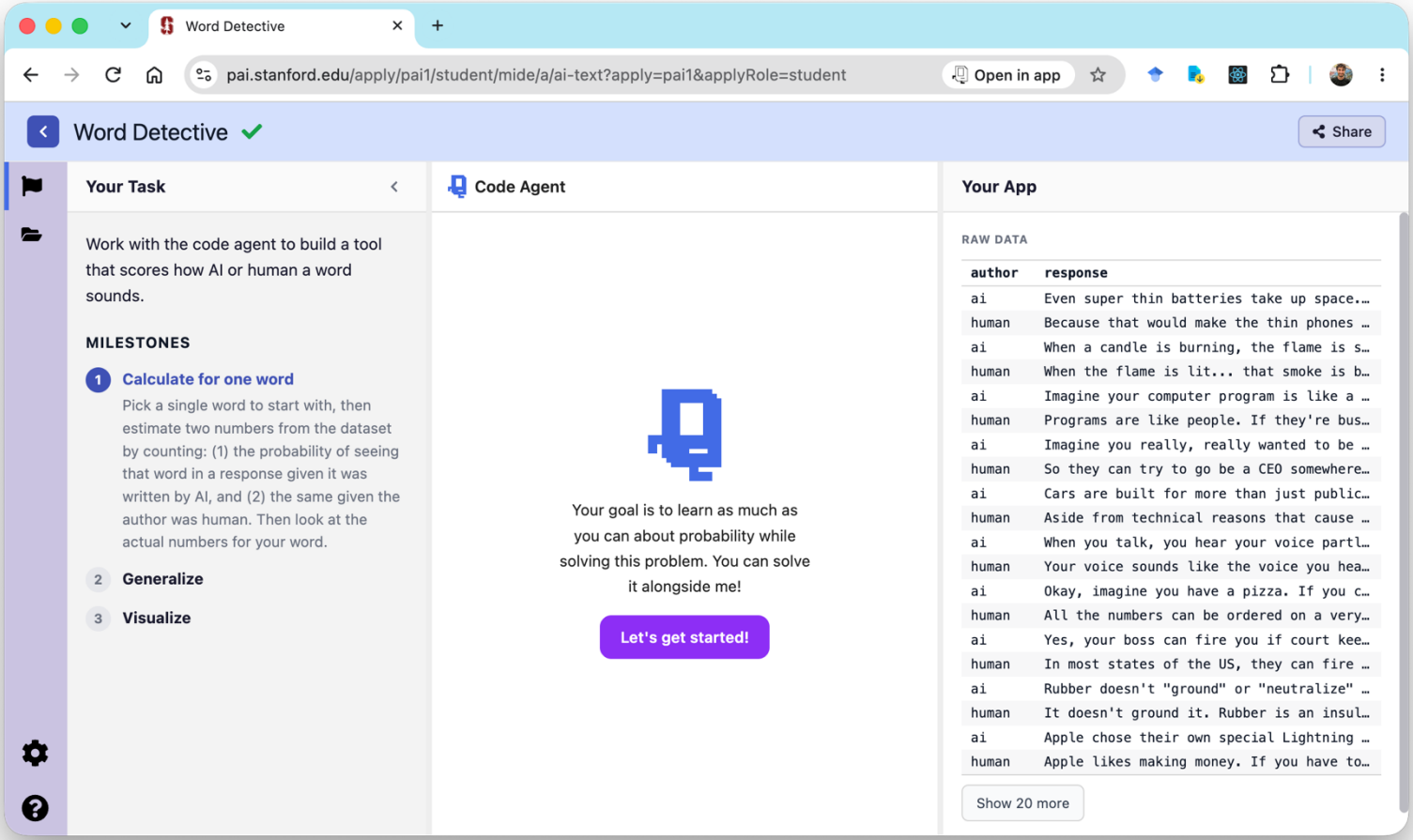
The student and the teacher are placed into a virtual room with shared IDE and video

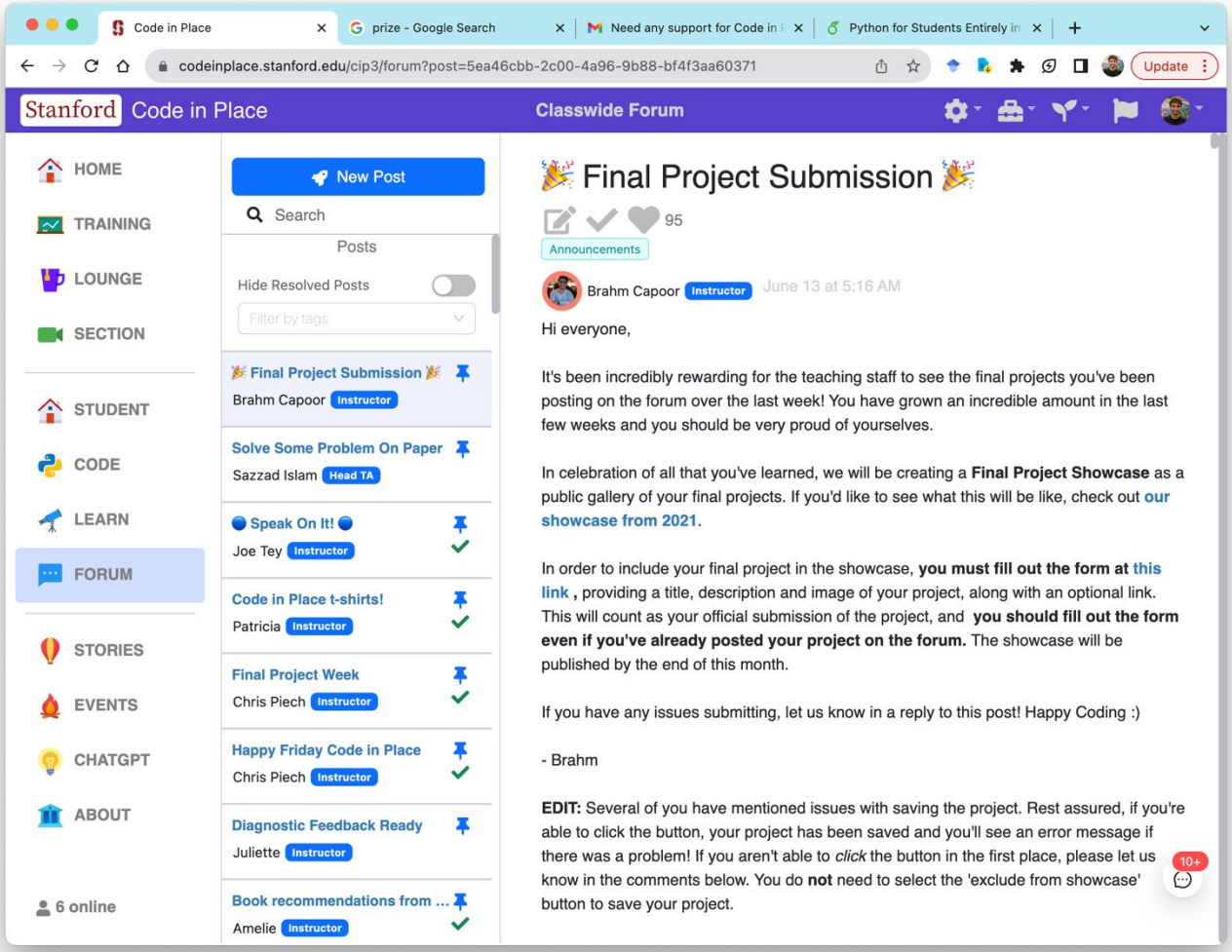
Just in Time Teaching




What is a probability - Stanford, Probability for AI
Code in Place
Chris! You didn't need to simulate dice...
Let E be the event that you roll a 5 or a 6 on a die
$\mathrm{P}(E) = \frac{2}{6}$
8:42 / 24:24
roll a four. Or you could roll a five or
a six. Out of those


Word Detective
Section 1
Code in Place
pai.stanford.edu/apply/pai1/student/learn/probability/s7ytks?apply=pai1&applyRole=student
Open in app
Lesson 1. What is Probability?
Give an interesting real-world example of an event E whose probability could be estimated using a dataset or simulation. Explain what count(E) and n would represent in your example....
Let E be the event that steph curry hits a free throw. n is the number of free throws he has shot, count(E) is the number he hit
TOP ANSWERS
1 Event E: A student drops out before their second year. n: The total number o…
2 Event: me getting an interview from a job I applied to Using data from my jo…
3 A real-world example is estimating the probability of a flight being delayed....
4 Event E: A user skips a recommended song within the first 10 seconds of it…
5 Event E: A free throw attempt in the NBA is made (successful). To estimate…
Continue
LESSON OUTLINE
What is a Probability?
Overview
1 min
Lesson Video
24 min
AI Art Survey
2 min
Creative Idea
2 min


Word Detective
pai.stanford.edu/apply/pai1/student/mide/a/ai-text?apply=pai1&applyRole=student
Open in app
Word Detective
Share
Your Task
Work with the code agent to build a tool that scores how AI or human a word sounds.
MILESTONES
1 Calculate for one word
Pick a single word to start with, then estimate two numbers from the dataset by counting: (1) the probability of seeing that word in a response given it was written by AI, and (2) the same given the author was human. Then look at the actual numbers for your word.
2 Generalize
3 Visualize
Code Agent
Your goal is to learn as much as you can about probability while solving this problem. You can solve it alongside me!
Let's get started!
Your App
RAW DATA
author response
ai Even super thin batteries take up space.…
human Because that would make the thin phones …
ai When a candle is burning, the flame is s…
human When the flame is lit... that smoke is b…
ai Imagine your computer program is like a …
human Programs are like people. If they're bus…
ai Imagine you really, really wanted to be …
human So they can try to go be a CEO somewhere…
ai Cars are built for more than just public…
human Aside from technical reasons that cause …
ai When you talk, you hear your voice partl…
human Your voice sounds like the voice you hea…
ai Okay, imagine you have a pizza. If you c…
human All the numbers can be ordered on a very…
ai Yes, your boss can fire you if court kee…
human In most states of the US, they can fire …
ai Rubber doesn't "ground" or "neutralize" …
human It doesn't ground it. Rubber is an insul…
ai Apple chose their own special Lightning …
human Apple likes making money. If you have to…
Show 20 more

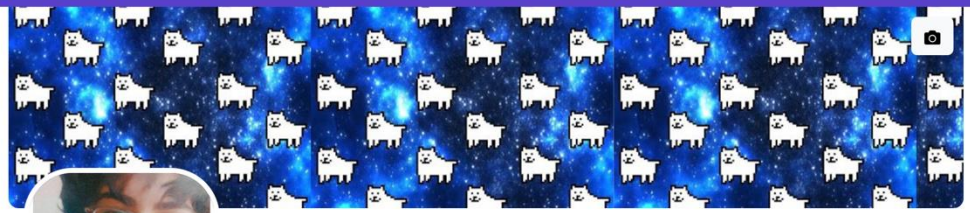

Code in Place
prize - Google Search
Need any support for Code in
Python for Students Entirely in
codeinplace.stanford.edu/cip3/forum?post=5ea46cbb-2c00-4a96-9b88-bf4f3aa60371
Update
Stanford Code in Place
Classwide Forum
HOME
TRAINING
LOUNGE
SECTION
STUDENT
CODE
LEARN
FORUM
STORIES
EVENTS
CHATGPT
ABOUT
6 online
New Post
Search
Posts
Hide Resolved Posts
Filter by tags
Final Project Submission
Brahm Capoor Instructor
Solve Some Problem On Paper
Sazzad Islam Head TA
Speak On It!
Joe Tey Instructor
Code in Place t-shirts!
Patricia Instructor
Final Project Week
Chris Piech Instructor
Happy Friday Code in Place
Chris Piech Instructor
Diagnostic Feedback Ready
Juliette Instructor
Book recommendations from ...
Amelie Instructor
Final Project Submission
95
Announcements
Brahm Capoor Instructor June 13 at 5:16 AM
Hi everyone,
It's been incredibly rewarding for the teaching staff to see the final projects you've been posting on the forum over the last week! You have grown an incredible amount in the last few weeks and you should be very proud of yourselves.
In celebration of all that you've learned, we will be creating a Final Project Showcase as a public gallery of your final projects. If you'd like to see what this will be like, check out our showcase from 2021.
In order to include your final project in the showcase, you must fill out the form at this link , providing a title, description and image of your project, along with an optional link. This will count as your official submission of the project, and you should fill out the form even if you've already posted your project on the forum. The showcase will be published by the end of this month.
If you have any issues submitting, let us know in a reply to this post! Happy Coding :)
- Brahm
EDIT: Several of you have mentioned issues with saving the project. Rest assured, if you're able to click the button, your project has been saved and you'll see an error message if there was a problem! If you aren't able to click the button in the first place, please let us know in the comments below. You do not need to select the 'exclude from showcase' button to save your project.
10+

codeinplace.stanford.edu/cip3/section/Aec03e0wnoqCYDCEUWaC

Stanford Code in Place — Your Section

- HOME
- TRAINING
- LOUNGE
- SECTION
- STUDENT
- CODE
- LEARN
- FORUM
- STORIES
- EVENTS
- CHATGPT
- ABOUT

14 online



The Amazing Programmers Section

Section Leader: Arezoo:)

Thursdays, 11am

Next section: All done!

Section Forum

Email Your Section

Join Zoom

ANNOUNCEMENTS

welcome everyone!😃
Make sure to check the handouts and material for tomorrow's section
Also, check the Section Forum and ask any questions you had while learning.
Happy coding!

WHO IS IN THIS SECTION

U N X E A M S N

K X

N


Cristian
faridaelchuzade
Elena
emily
Metehan Calis
Saidbek Samigonov
Mirelle Kamariza

Question: If prompted twice, what is the probability the model produces two different outputs?

Solution:

Let E be the event that both calls produce ***the same*** output

Let B_i be the event that both calls produce output i.

$$\mathrm{P}(B_i) = p_i^2$$

$$\mathrm{P}(E) = \sum_i p_i^2$$

$$\mathrm{P}(E^C) = 1 - \sum_i p_i^2 \approx 0.72$$