Pokerbots 2025

Lecture 8: Counterfactual Regret Minimization (CFR)

Sponsors

HRT
hudson river trading

Jump
TRADING

VIRTU
FINANCIAL

CITADEL | CITADEL Securities

DRW

MIDPOINT
MARKETS

FIVE RINGS

imc

TREX
QUANT

Jane
Street®

{...} codeium

CTC®
CHICAGO TRADING COMPANY®

2σ
TWO SIGMA

SUSQUEHANNA

Announcements

Week 3 Bot Deadline

- Third pokerbot submission due Friday 1/24, 11:59PM EST on scrimmage server
- Mini-tournament 3 will occur shortly after

Next Week's Guest Speakers

- Monday 1/27: GTO Wizard
  - Current state of the art Poker Solver and Study tool
  - Developed by team of top AI researchers and engineers
  - Core team will be presenting their journey and research

- Wednesday 1/29: Dr. Noam Brown
  - Research scientist at OpenAI, previously Meta AI
  - Created Libratus and Pluribus, the world's first superhuman poker AIs
  - Leading role in developing OpenAI's latest o1 and o3 LLMs
  - Poker social after lecture!

Final Dates

- Tuesday 1/28: Final Strategy Reports
    - 3-5 pages, double spaced
    - Submission will be open on Canvas, due 11:59PM
    - Description in syllabus at pkr.bot/syllabus
- Wednesday 1/29: Final Bot Submission
    - Bot that will be used in final tournament
    - Submission to scrimmage server, due 11:59PM
- Friday 1/31: Final Event
    - Sponsor Networking Event
    - Presentation of Awards and Closing Ceremony

Today’s Giveaway!

Twenty Five Game: **pkr.bot/25**

- 25th submission wins!
- Prize: Choose Nintendo Switch OR Airpods Pro

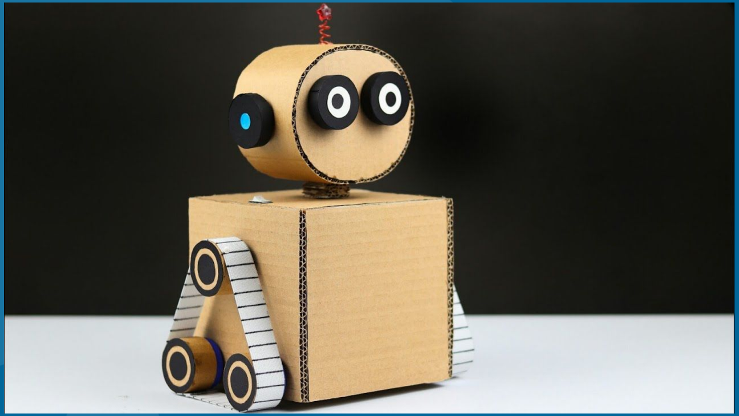
Hint: Last lecture giveaway entries were submitted roughly once every 1.5 seconds

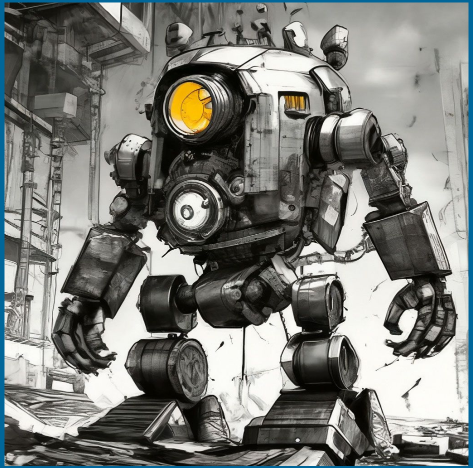
Counterfactual Regret Minimization (CFR)

Paco Gomez-Paz

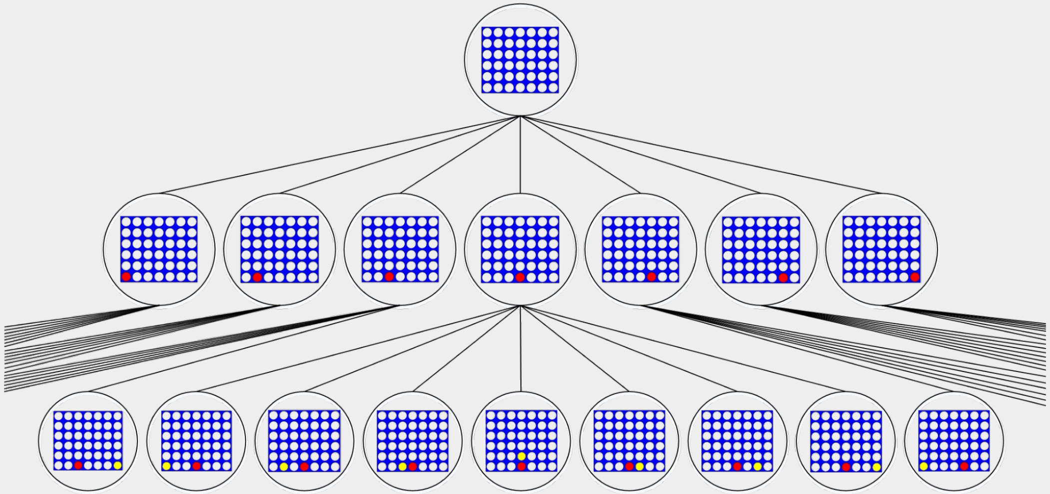
Big Goal

- Formalize some notion of "regret"
- Use this to build algorithms which "learn from past mistakes"





Connect Four Game Tree



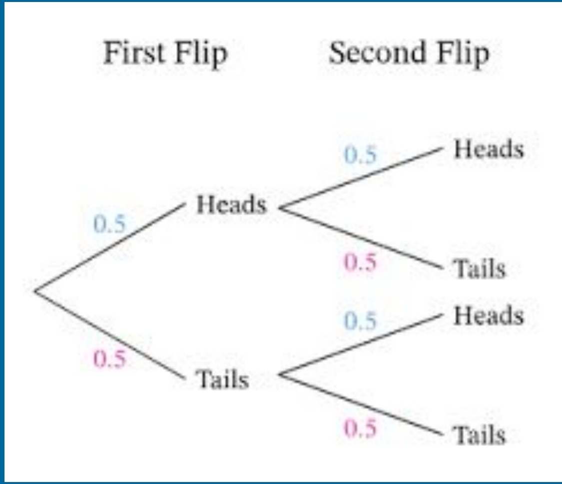
Games and Trees

- Nodes - Represent a unique state in the game

- Decision Node: Player chooses between actions
- Folding/calling/raising in poker

- Chance Node: Randomness dictates what happens
- When the flop is dealt in Poker

- Terminal Node: Game terminates. Contains player rewards
- Showdown

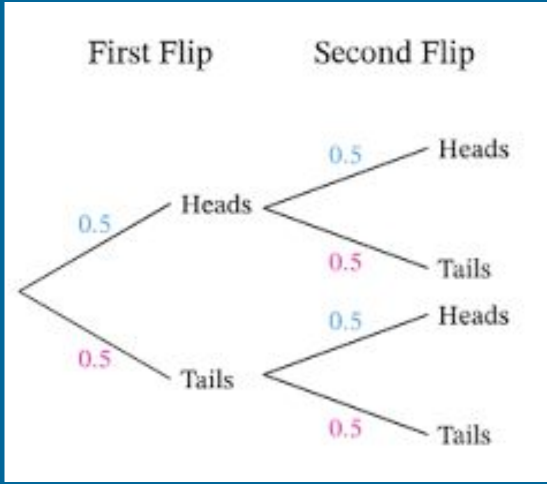


Games and Trees

- Edges - Actions/events that advance the game

- Decision Edge - edges represent player actions
- A strategy assigns a probability to each edge

- Chance Edges - different outcomes for random event
- Edges weighted according to event probabilities



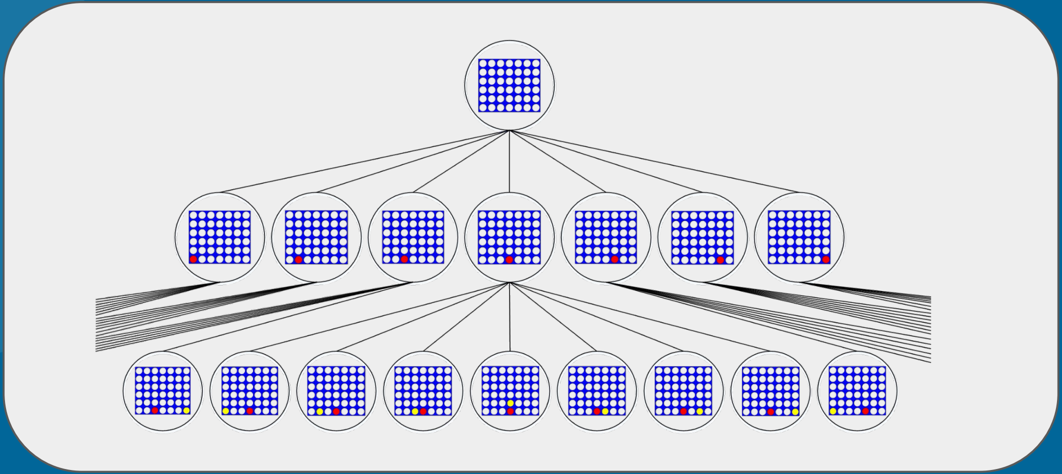
Information Sets

- Imperfect information - Game with hidden information
- Poker: Your opponents hole cards are hidden information

Information Set (I) - Group of game state indistinguishable to a player
Determined by visible information: Hole Cards, Board Cards, Stack Sizes, Betting History ...

Strategy

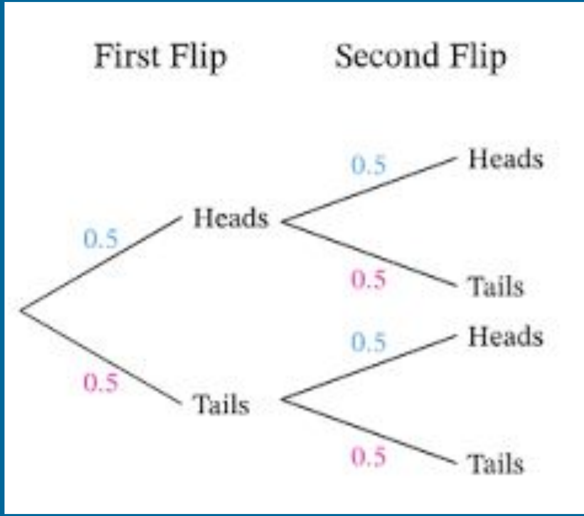
- *Strategy Profile* (S) - Choose actions based off the information available to us
- Represent as function: S(X, I) = Probability of choosing action X at infoset I



State Values

- Nodes has an expected value E(I | S), representing the expected reward starting from that node
- Compute E(I | S) recursively from values of its child nodes + the probabilities associated with transitions to those nodes

- E(Start Node) = ½ E(Heads Node) + ½ E(Tails Node)
- Action Value: E(Action X at I) = E(Resulting Node)
- Notation: E((X, I))



Questions?

Intuition

- “If I hadn’t gone all in I would have saved so many chips”

Counterfactual Regret

- (Counterfactual) Regret: How much would we regret NOT taking action?
- PNL = Chips Won/Lost
- Regret(All in) = E (PNL | Go all in) - E(PNL | Play standard Strategy)

- More generally: Action : X, Infoset : I, Strategy: S
- Regret(X, I) = E((X, I) | S) - E(I | S)

Higher Regret $\Rightarrow$ Better than existing strategy

Negative Regret $\Rightarrow$ Worse than existing strategy

Regret Matching

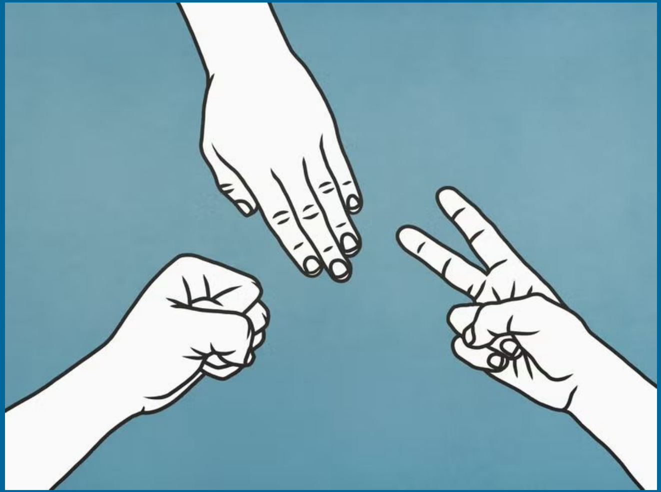
- Goal: Construct a strategy that learns from a set of regrets
- 1. Don't play bad action
- 2. Given multiple good action, play the best the most
- Formalizing Intuitions: $$S(X, I) \propto \max(R(X, I), 0)$$
- Normalizing gives us: $$S(X, I) = \frac{R^{+}(X,I)}{\Sigma_{a \in A} R^{+}(a,I)}$$

CFR

- Have a *strategy profile*
- Use game tree to assign each game state a *value*
- Calculate *(counterfactual) regrets* for each action
- Use the regrets to inform our next strategy profile
- Repeat
- (Optional)- Aggregate all strategy profiles into one final profile

Toy Example

- Rock Paper Scissors: Each player has 3 action (R, P, S)
- Payouts : Loss = 0, Tie = 1, Win = 2



Learning From Cumulative Regrets

1. Compute player strategies by matching *cumulative* regrets
2. Create game tree and calculate new regrets
3. Update cumulative regrets.
4. Repeat T times.
5. Return the *average* strategies across the T iterations.

RPS Example

RPS: We play rock, opponent plays paper

Regrets: 0 for rock, +1 for paper, +2 for scissors $\longrightarrow$ (0, 1, 2)

Mixed strategy: (r=0, p=1/3, s=2/3), opponent plays scissors

Value: 2/3 for our mixed strategy,

New counterfactual regrets: (4/3, -2/3, 1/3),

New mixed strategy: (r=1/3, p=1/12, s=7/12) …

New *average* strategy: (r=1/6, p=5/24, s=5/8)

Questions?

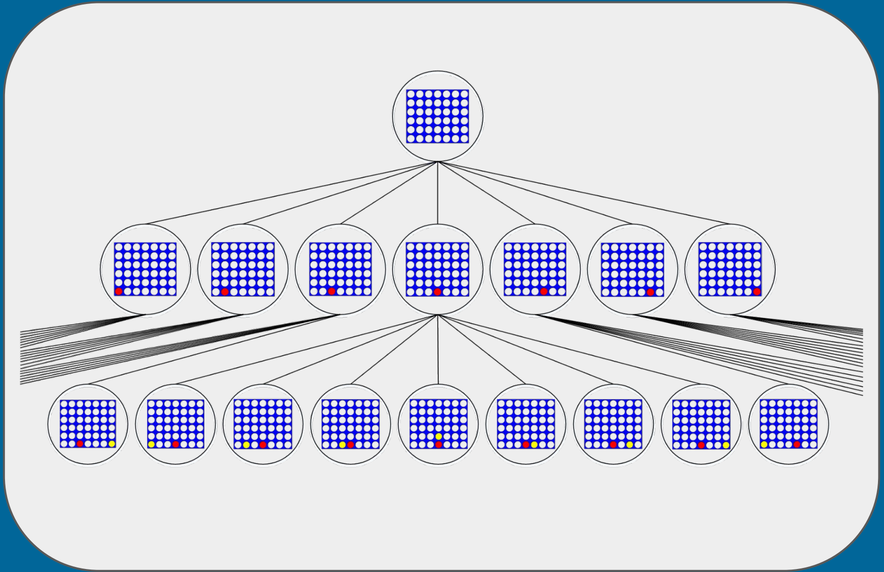
Implementation

Problem: Game tree grows exponentially in number of actions

-makes it infeasible to store

Solution: Discretize Action Space

-Rethink how we compute regrets



Connect Four Game Tree

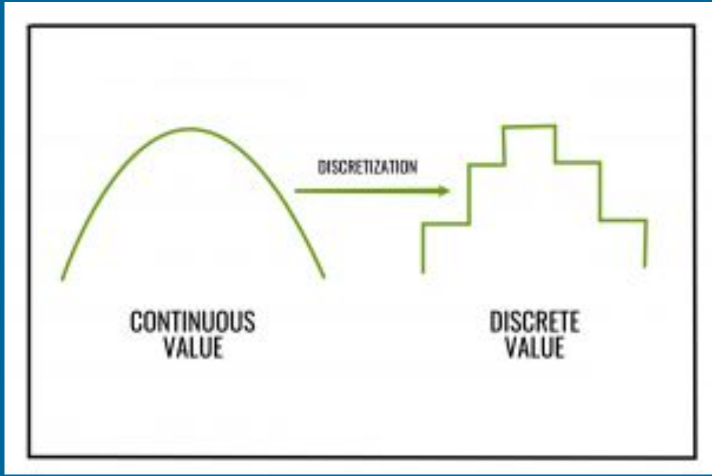
Discretization

Naive Action Space: Fold, Check, Call, Min Raise, Min Raise + 1 …. Max Raise

Discretized Raises: Half Pot, Pot, 3/2 Pot

-Pro: less memory, faster training

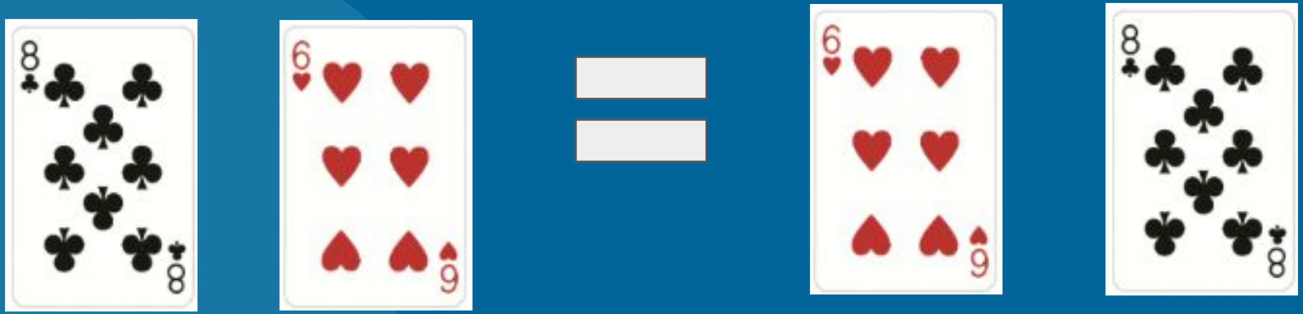
- Con: Worse final strategy



Bucketing

Problem: For large games like poker, we can't store tables of all the infosets

Bucketing: Group similar infosets together to reduce game size (bet sizes, hands ..)



Monte Carlo Simulations

Estimating E(I | S)

- Start at infoset I
- Play according to strategies S, S'
- Store value of terminal State
- Repeat N times and average terminal state values

Estimating E ((X,I) | S)

- Start at infoset I. Take action X.
- Play according to strategies S, S'
- Store value of terminal State
- Repeat N times and average terminal state values

Monte Carlo CFR

- Have a *strategy profile*
- Use monte carlo simulations to estimate game states *values*
- Use monte carlo to estimate *(counterfactual) regrets* for each action
- Use the regrets to inform our next strategy profile
- Repeat
- (Optional)- Aggregate all strategy profiles into one final profile

Questions?

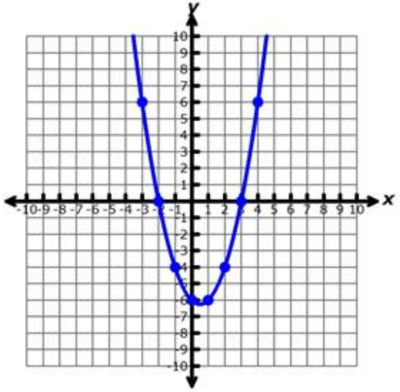
Lunch 😋

Leave any type of feedback at **pkr.bot/feedback**!

Deep CFR

- Replace explicit data with function approximation

x	*y*
−3	6
−2	0
−1	−4
0	−6
1	−6
2	−4
3	0
4	6



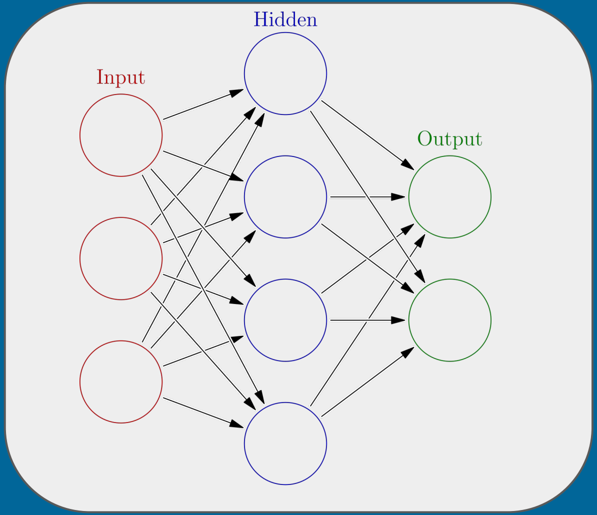
- Regret Table : Maps (X, I) -> R(X,I)

- Deep CFR (2018) : Replace regret tables with neural net approximations

Deep CFR

- Deep CFR (2018) : Store regret models instead of regret tables
- Given regret model F, Strategy determined by regret matching on F(X,I)
- Reservoir Sampling: Uniformly sample stream of data

Data



Cumulative Regrets

Strategy Profile

		Player 2	
		A	B
Player 1	A	2,2	0,1
	B	1,0	1,1

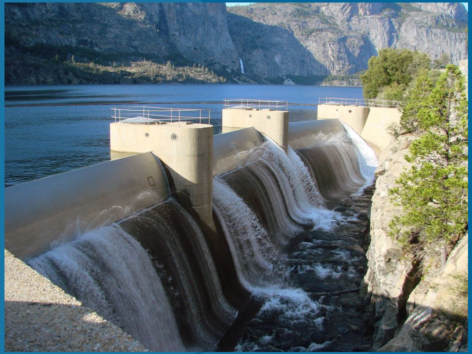
Strategy Profile

		Player 2	
		A	B
Player 1	A	2,2	0,1
	B	1,0	1,1

Explicit Regret calculations

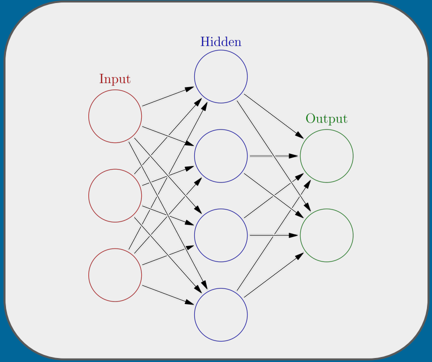
Cumulative Regrets

Regret Reservoir

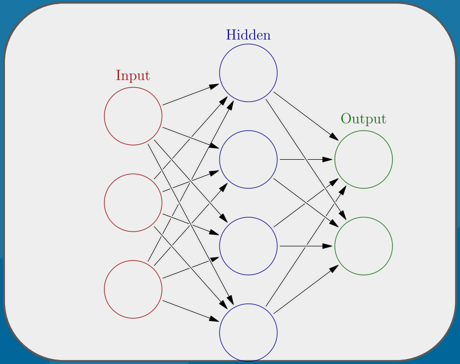


data to train on

Regret Network



Regret Network



Simulated Regret Estimates

Regret Reservoir

Deep MCCFR

- Have a *regret reservoir* [(X,I), R(X,I)]
- Train *regret network* on reservoir data
- Use monte carlo sim to estimate game states *values* for our strategy
- Use monte carlo to estimate *(counterfactual) regrets* for each action
- Add counterfactual regrets to *regret reservoir*
- *Repeat*
- (Optional)- Aggregate all strategy profiles into one final profile

Some History

- Dr. Noam Brown: co-created Deep CFR while PhD student at CMU

  - Completed in Pokerbots while CMU grad student

  -Created Pluribus (Poker) and Cicero (Diplomacy) at FAIR

  - Worked on reasoning in LLMs at OpenAI (o3 and o1)

  - Is giving a lecture + joining for Poker Social (1/29)

Deep Counterfactual Regret Minimization

Noam Brown*
Computer Science Department
Carnegie Mellon University
Pittsburgh, PA 15213
noamb@cs.cmu.edu

Adam Lerer*
Facebook AI Research
New York, NY 10003
alerer@fb.com

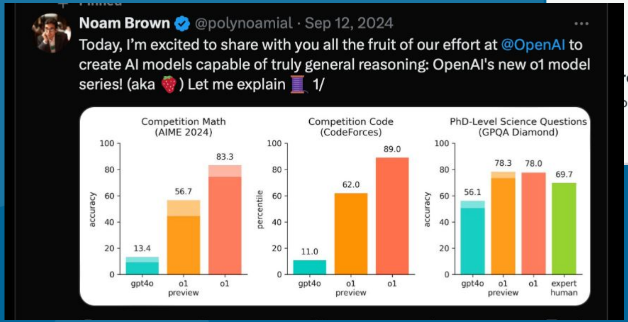
Sam Gross
Research
Y 10003
b.com

Tuomas Sandholm
Computer Science Department
Carnegie Mellon University
Pittsburgh, PA 15213
sandholm@cs.cmu.edu

DIPLOMACY GAMEPLAY

Playing Diplomacy with CICERO

ew Goff — lomacy World Champion
Noam Brown — Meta AI Research Scientist
Adam Lerer — Meta AI Research Engineer
CICERO — AI Agent



Superhuman AI for multiplayer poker

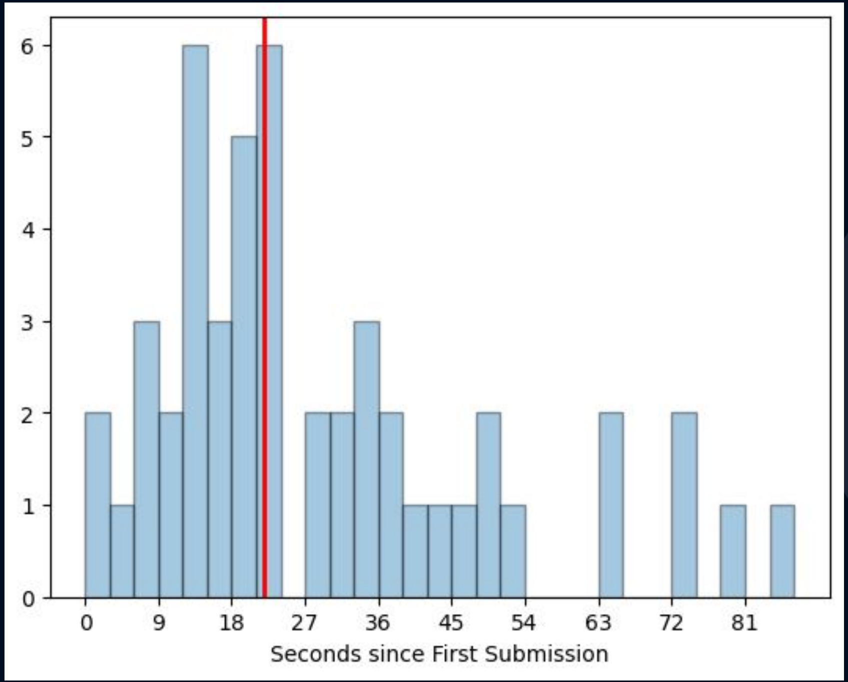
NOAM BROWN AND TUOMAS SANDHOLM Authors Info & Affiliations

Live Coding: reference-8-2025

Giveaway Winners

Twenty Five Game: kerb “jlh28”

at 12:20:05
1st submission at 12:19:43



1/22/2025 12:19:43	justinwz
1/22/2025 12:19:45	eddyc
1/22/2025 12:19:46	naglerj
1/22/2025 12:19:49	sam_wang
1/22/2025 12:19:49	swathy
1/22/2025 12:19:50	juszha
1/22/2025 12:19:53	ejrice
1/22/2025 12:19:53	daniknut
1/22/2025 12:19:55	brianle
1/22/2025 12:19:55	sarun
1/22/2025 12:19:55	azd
1/22/2025 12:19:56	zjperry
1/22/2025 12:19:56	kevinmz
1/22/2025 12:19:57	bocchi
1/22/2025 12:19:58	rgao21
1/22/2025 12:19:58	Merey
1/22/2025 12:19:59	Eposondu
1/22/2025 12:20:01	mattzhou
1/22/2025 12:20:01	gsjau
1/22/2025 12:20:02	rlsalas
1/22/2025 12:20:03	ldhuang
1/22/2025 12:20:04	annieguo
1/22/2025 12:20:03	tiffany8
1/22/2025 12:20:04	tiago13
1/22/2025 12:20:05	jlh28
1/22/2025 12:20:06	Metalor
1/22/2025 12:20:06	megansun
1/22/2025 12:20:06	kjiang77

Thanks for watching!

Slides will be posted on **pkr.bot/resources**

Repo will be pushed to **pkr.bot/github**

Make sure to check **pkr.bot/piazza** for updates

Lecture recordings at **pkr.bot/panopto**

Leave feedback at **pkr.bot/feedback**!