Pokerbots 2025

Lecture 10: GTO Wizard

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

Subject Evaluations Open!

pkr.bot/eval

Please leave an honest review of your experience in this class!

Week 3 Mini Tournament Results

WINNER [$1000]:

Pineapple

BIGGEST UPSET [$500]:

zen_fold

Tournament ELO 1097, won majority matches against “Hawk Tuah” (ELO 2122)

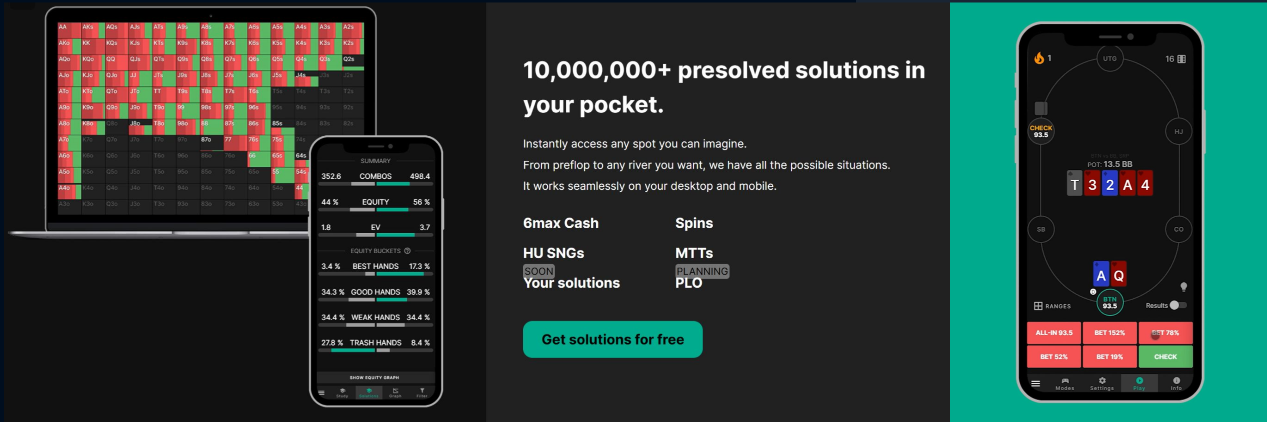
MOST IMPROVED [$750]:

Peak Productivity

Tournament ELO 1006 → 1865

Today 1/27

Guest Lecture: GTO Wizard



Tomorrow 1/28

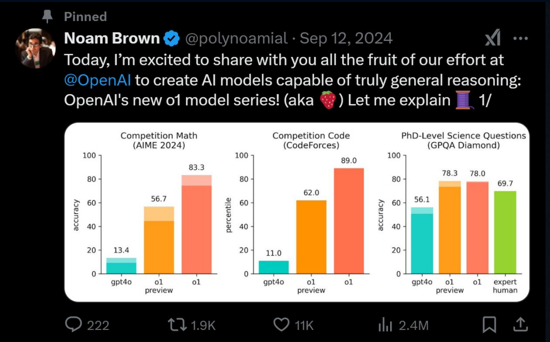
Team Strategy Reports due 11:59PM EST

- No Lecture
- 3-5 pages, double spaced, ≥1 member must submit
- Assignment details and submission on Canvas: **pkr.bot/canvas**
- Description also available in syllabus: **pkr.bot/syllabus**

Wednesday 1/29

Guest Lecture: Dr. Noam Brown

- Won Pokerbots and co-created Deep CFR while PhD at CMU
- Created Libratus and Pluribus, the world's first superhuman poker AIs
- Research scientist currently at OpenAI, previously worked at Meta AI
- Leading role in developing OpenAI's latest GPT o1 and o3 LLMs



Wednesday 1/29 (cont.)

Poker Social After!

- During office hours block
- 32-044, 2-4PM
- Come play with Noam!

Wednesday 1/29 (cont.)

Final Bot Submission due 11:59PM EST

- Upload and select bot as active on scrimmage server
- Both report and bot needed to pass this class!
- Bot will compete in last and final Pokerbots tournament
- Non-secret prize amounts listed on syllabus

Final Tournament Prizes	
First place	$10,000
Second place	$6,500
Third place	$3,500
Fourth place	$2,000
Fifth place	$1,000
First place in language (Python, Java, or C++)	$500 x 3
Second place in language (Python, Java, or C++)	$250 x 3
Third place in language (Python, Java, or C++)	$125 x 3
Best freshman-majority (>51%) team	$2,000

Friday 1/31

Pokerbots Final Event 4:30-7PM in Kresge Auditorium

- Presentation of Awards
- Closing Ceremony
- Sponsor Event and Puzzle Hunt
- Lots of free merch and raffle prizes!
- Dinner provided 😋

Sat 2/1

Jump Trading Poker Tournament!

- 5-8PM in BC Porter Room (tentative)
- $3500 cash prize pool

All in all...

1/27 Today	GTO Wizard Talk
1/28 Tomorrow	Final Report Due
1/29 Wednesday	Noam Brown Talk Social Event Final Bot Due
1/31 Friday	Final Event
2/1 Saturday	Jump Tournament

Giveaways!

Eval Raffle: pkr.bot/evalraffle

- Finish filling out our course evaluation at pkr.bot/eval and drop your kerb!
- Must have proof of completion to claim prize
- Two winners selected at random
- Prize: JBL Charge 5 and GTO Wizard Subscription

Guest: GTO Wizard

GTO Wizard

History of AI in Poker

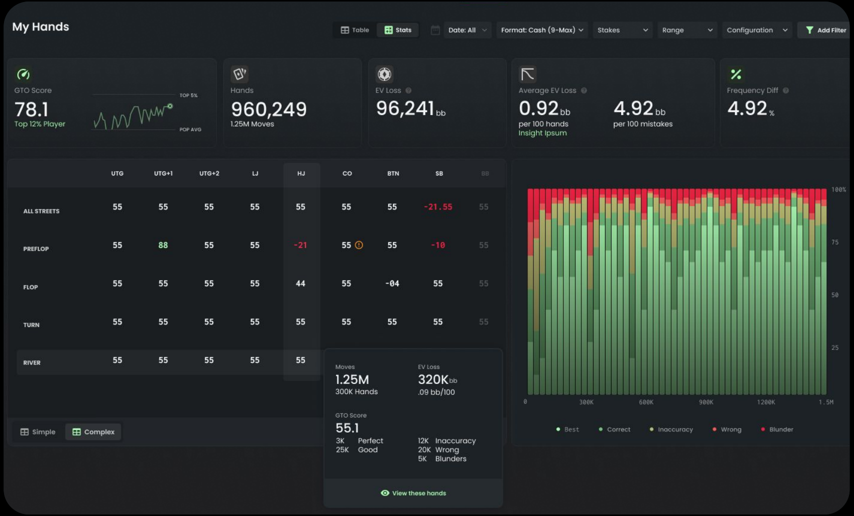
The last 10 years of AI progress in poker

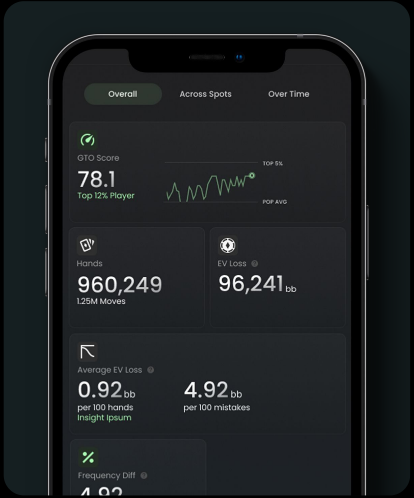
GTO Wizard

About us

- Biggest educational application for poker players
- Founded 4 years ago and grew to 70 employees

GTO Wizard





No.1 App for poker players

About us

- Biggest educational application for poker players
- Founded 4 years ago and grew to 70 employees
- Strong community and partner with some of the biggest players in the industry
- Sponsor major events such as the World Series of Poker and Triton series

GTO Wizard

About us

- About GTO Wizard AI - Team of 7 researchers - and rapidly expanding
- Built the strongest AI for heads-up poker
- Our goal is to **solve any poker variant to a very high accuracy in a few seconds**
    - From 2 to 9 players
    - Hold’em, Omaha, Short-deck, etc.
    - Any cash game and tournament variations

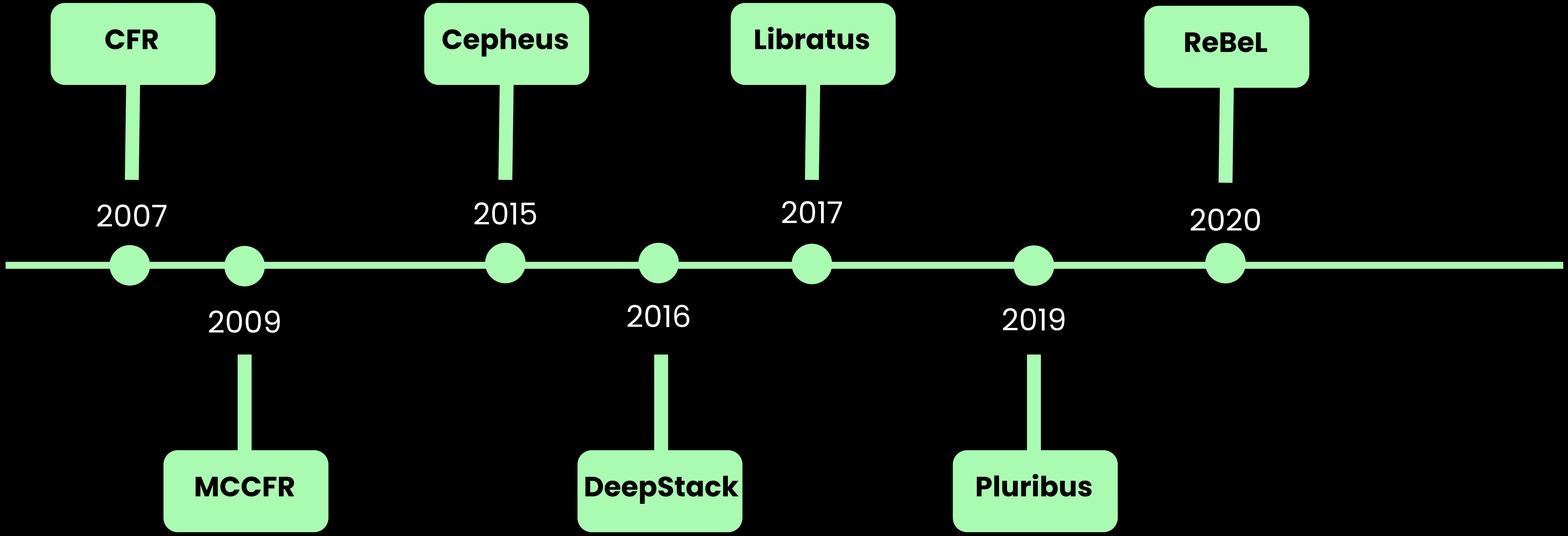
GTO Wizard

History of AI in Poker

The last 10 years of AI progress in poker

GTO Wizard

Timeline of Major Breakthroughs

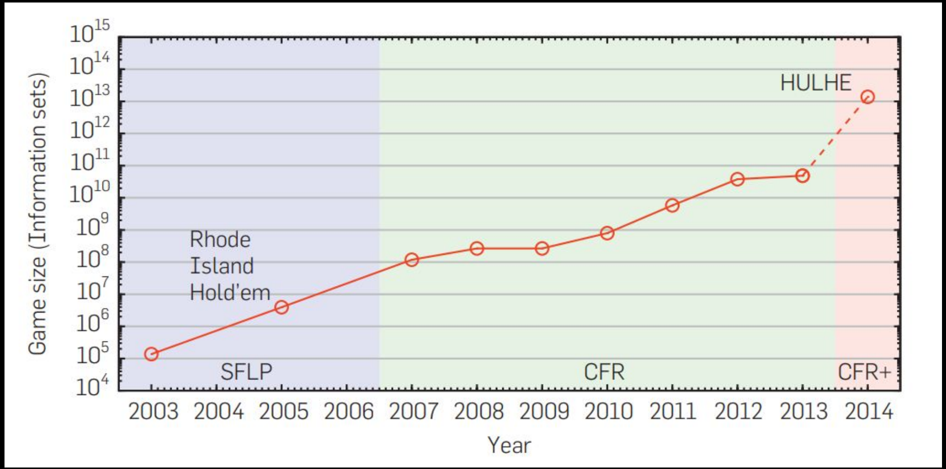


Cepheus (Bowling et al., 2015)

Essentially solved Heads-Up Limit Hold'em, the smallest variant of poker that humans actually play (10^{14} decision points).

Thanks to algorithmic advances like CFR+, that allowed to solve games a few order of magnitude larger.

Storing the strategy would require 262 TiB of memory with 4-byte values.



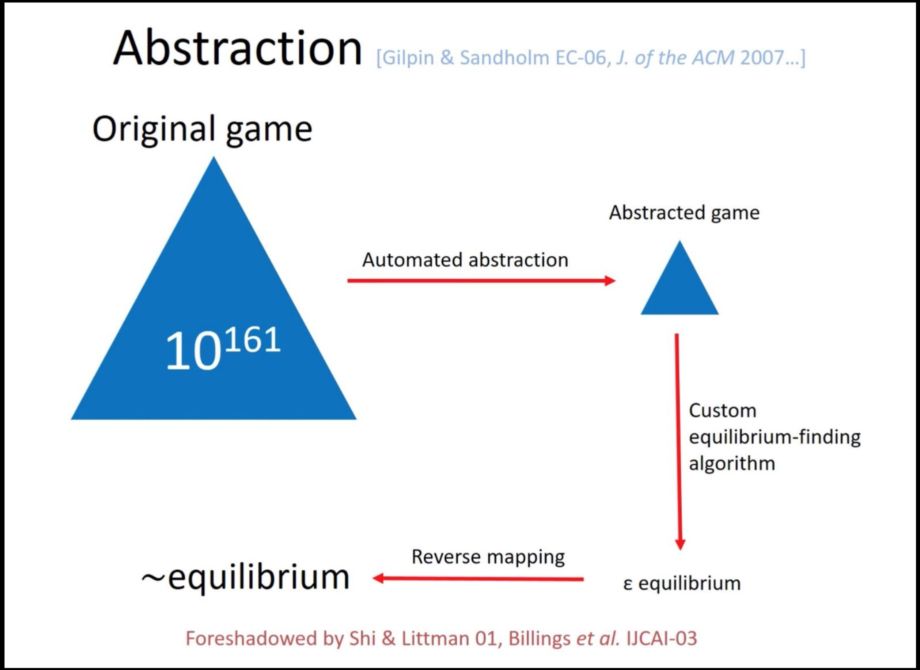
Size of imperfect-information games that we can exactly solve, plotted over time (Bowling et al., 2015)

→ Problem! No-limit Hold'em has around 10^{160} decision points!

Solving with Abstractions

Technique used by every winner of the Annual Computer Poker Competition (ACPC) for NLH, until the last year (2018)

- Solve an abstracted version of the game
- At inference time, simply use this as a lookup table



GTO Wizard

Information Abstraction

The idea is to group strategically similar decision points

In Hold’em, this represents a mapping from strategically similar {hand, board} combinations to buckets.

When using very large abstractions, these buckets can represent pretty sophisticated categories. Here are two real examples of buckets used by our turn’s neural network:

1. 9-high flush draws on a two-broadway board where one board card is lower than the side card
2. King-high flush draws on a Ace-high board where the lowest board card is higher than the side card

Bucket 1
(1400 combinations)

9c6c on AdKd7c4c

9c4c on KdJd7c3c

9c6c on AdQd7c3c

9c4c on JcTc5d2d

9d5d on AdQd8c2c

9c3c on AdJd6c2c

Bucket 2
(1100 combinations)

Kc4c on AcQc6d5h

Kc2c on AcJc6d4h

Kc3c on Ac9d7c4h

Kc5c on 7cAd6hJc

Kc6c on AcJh9d7c

Kc4c on AhJc7d5c

How are these abstractions created?

- Define a set of features for each information set in the original game
- Define a distance function $d_{\{i,\, j\}}$ between pairs of information sets
- Run a clustering algorithm (e.g. k-means)

➔ Obtain a many-to-one mapping between the full game's information sets and the information sets in a smaller abstracted game

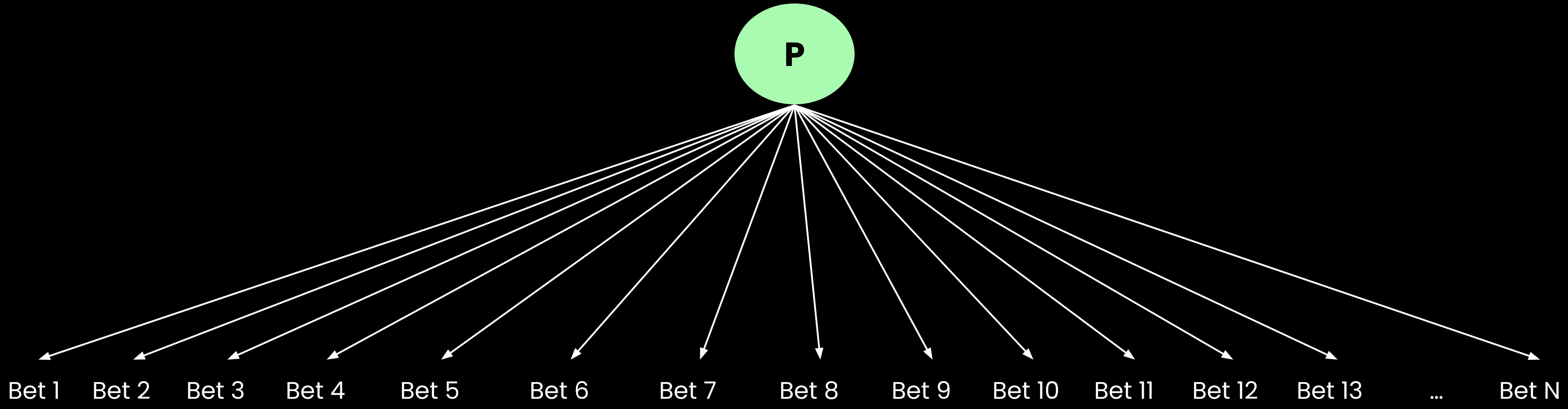
Example: Expectation-based abstractions

Idea: Group similar states together according to some metric

The most common and obvious one is the expected hand strength E[HS], commonly called the equity in poker, which is equal to $P(win) + P(tie)/2$ against a uniform random range.

{Hand, Board}	E[HS]
{2♠2♣, K♠8♦6♥2♥5♦}	0.9394
{8♠8♦, J♠8♣6♥A♠T♥}	0.9424
{J♠2♠, K♠8♦6♥2♥5♦}	0.4192
{6♠4♠, J♠8♣6♥A♠T♥}	0.4202

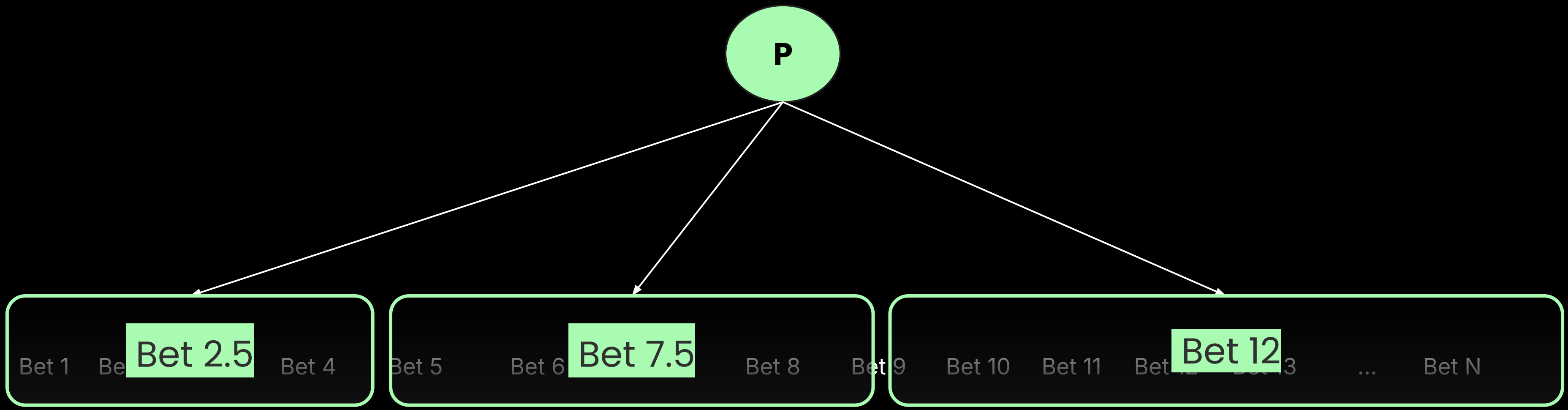
Action Abstraction



The action space in no-limit hold'em is huge, but most actions are very strategically similar

GTO Wizard

Action Abstraction



In poker, it's straightforward to group similar actions, but that's not usually the case for other games

There exists techniques called **action translation** to deal with actions that are out of your abstraction at test time

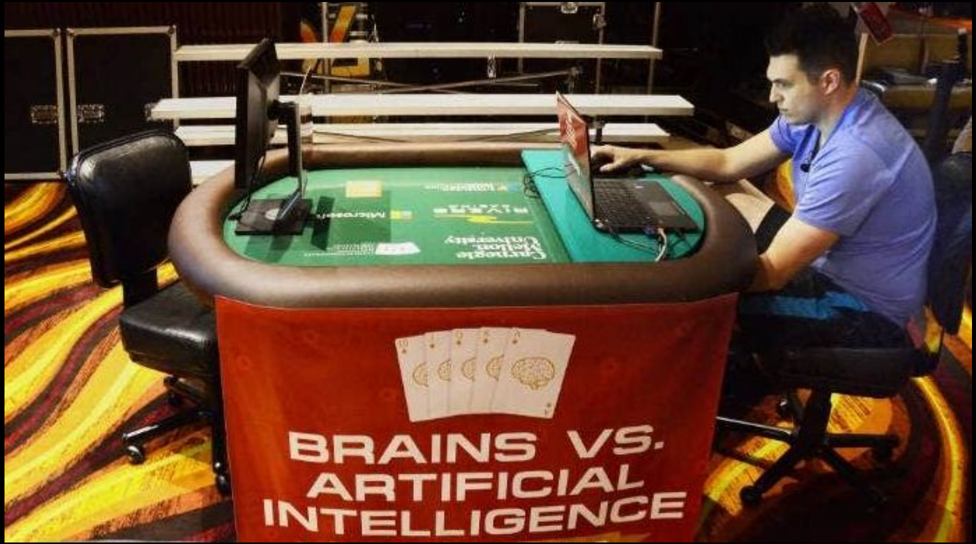
GTO Wizard

Solving with Abstractions

Not enough to achieve superhuman performance

Claudico, a bot from CMU leveraging these techniques, was tested against four top Heads-up NLH professionals.

- Lost by 9 BB/100, a significant margin



Claudico Brain vs AI (2015)

Problem with abstraction-based agents

Abstraction-based agents have been shown to be highly exploitable. Against their worst-case opponent, they are worse than the Always Fold strategy!

Local Best Response (LBR) is a technique for finding a lower-bound on a strategy's exploitability

Agent	LBR performance (bb/100)
Always Fold	75.0 ± 0
Slumbot (2016)	402.0 ± 11.5
Act1 (2016)	259.7 ± 14.0
DeepStack (2017)	-38.3 ± 21.9

GTO Wizard

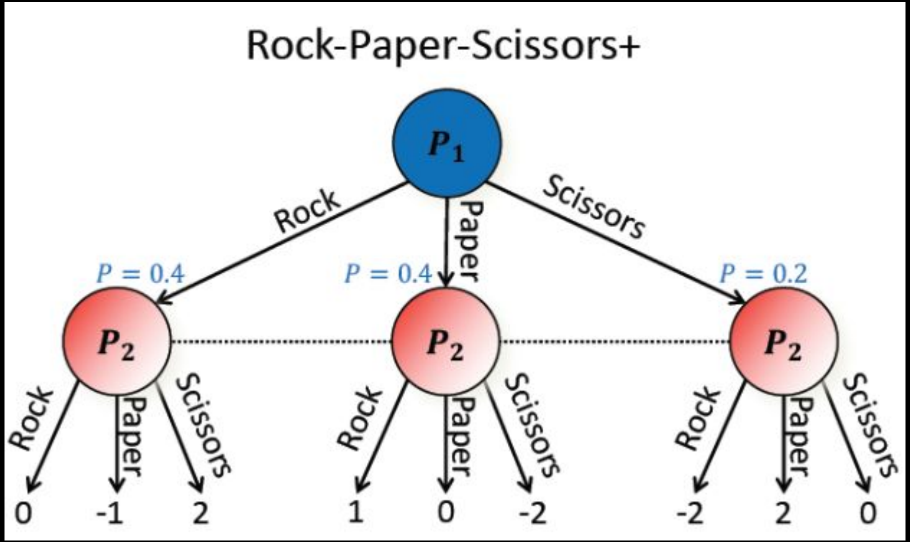
➔ Solution: Search (Real-time reasoning)

Why Search is Hard

Search has been used for decades in perfect-information games, so why is it just recent in imperfect-information games?

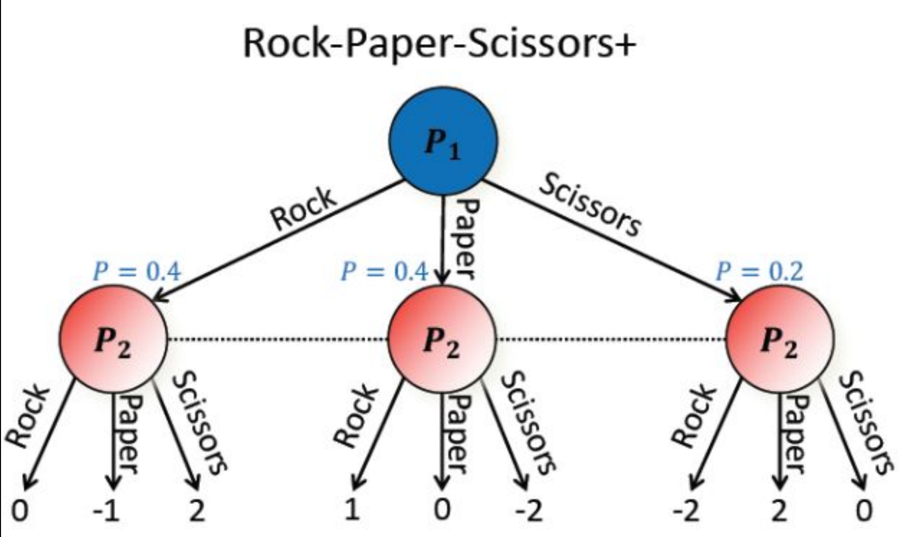
Problem: In imperfect-information games, states don't have unique values

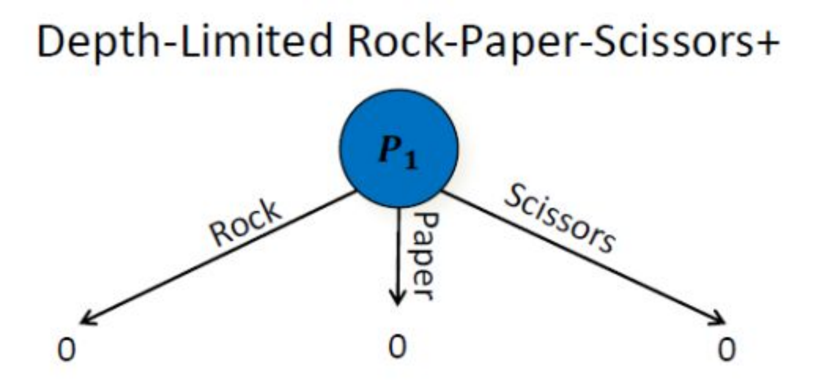
Depth-Limited Search



Taken from (Brown, 2020)

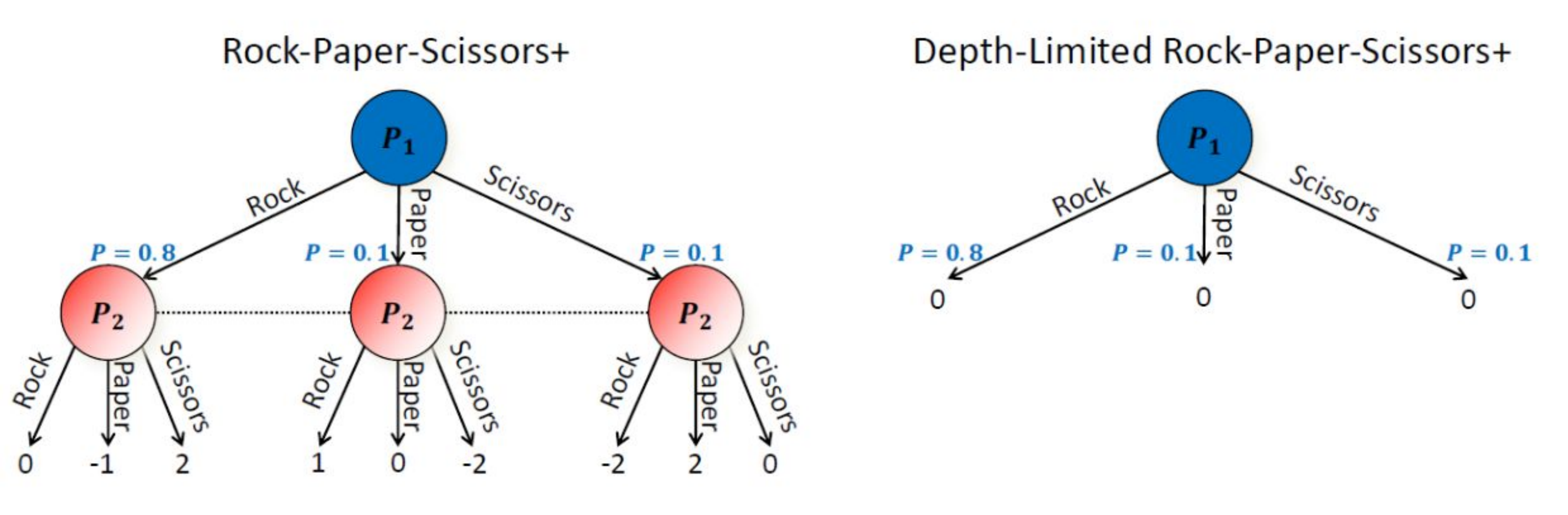
Depth-Limited Search





Taken from (Brown, 2020)

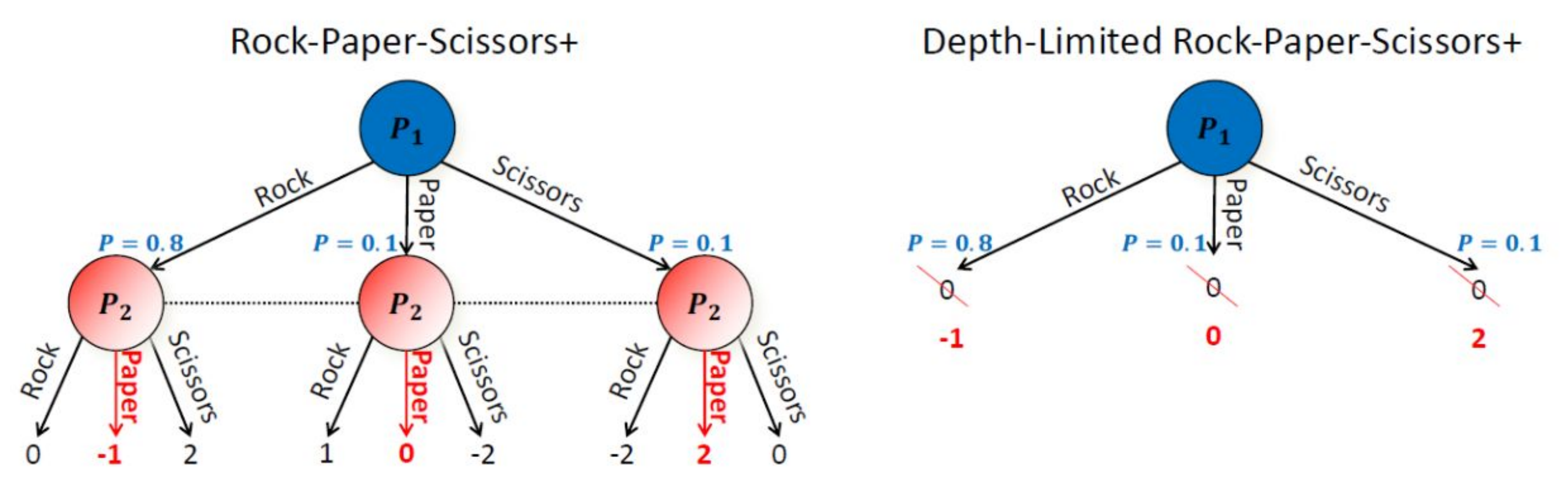
Depth-Limited Search



GTO Wizard

Taken from (Brown, 2020)

Depth-Limited Search



One solution is to condition the value on the probability distribution over states

- $v(Rock)$ is not well defined
- $v(0.8\ Rock, 0.1\ Paper, 0.1\ Scissors) = -0.6$

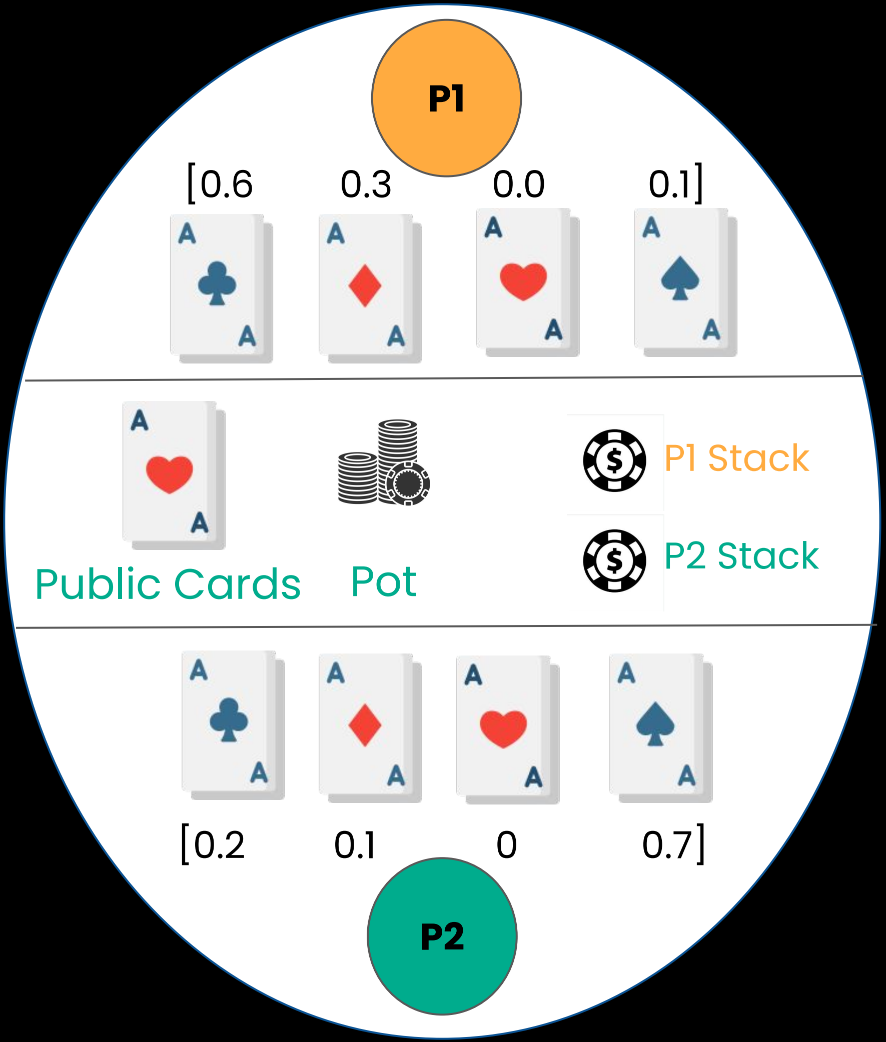
Taken from (Brown, 2020)

Public Belief States (PBS)

PBS: A common-knowledge probability distribution over states in some public state.

Properties:

- Identical to perfect-info states in perfect-info games
- Have a unique value in two-player zero-sum games



GTO Wizard

DeepStack (Moravčík et al., 2017)

Algorithm that successfully implement heuristic search in imperfect-information games

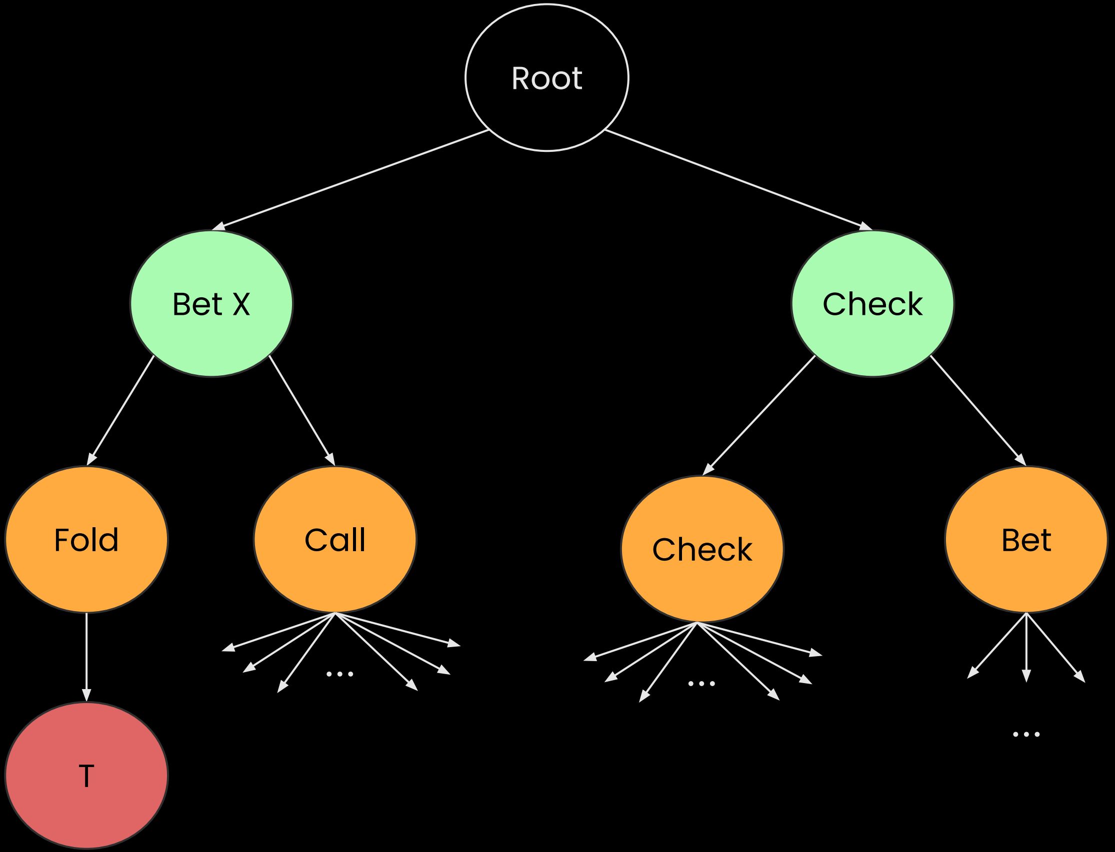
- Introduces a way to decompose the game in depth-limited subgames with CFR-Decomposition.
- Uses a value network to estimate the leaf nodes at a given depth limit.
- The depth-limited subgame is small enough to be solved in real-time "without the need for abstraction".

Depth-limited solving with value networks

Player 1

Player 2

Terminal



GTO Wizard

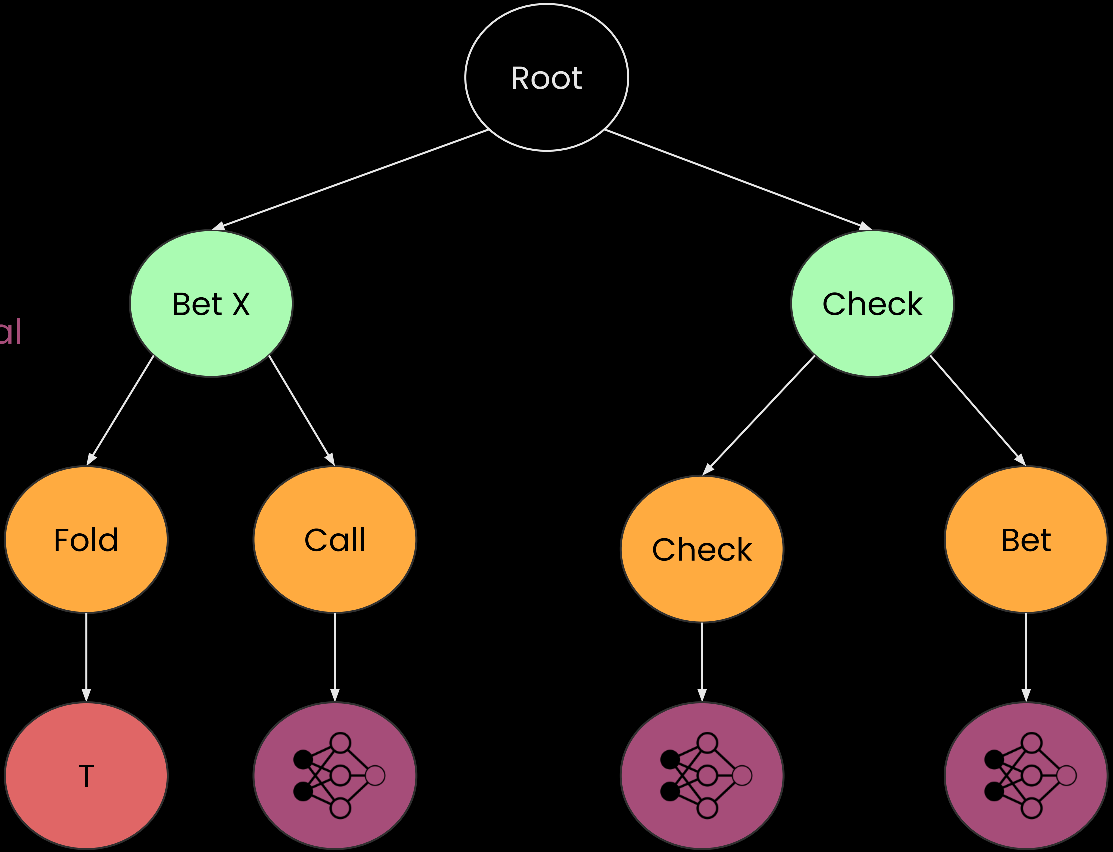
Depth-limited solving with value networks

Player 1

Player 2

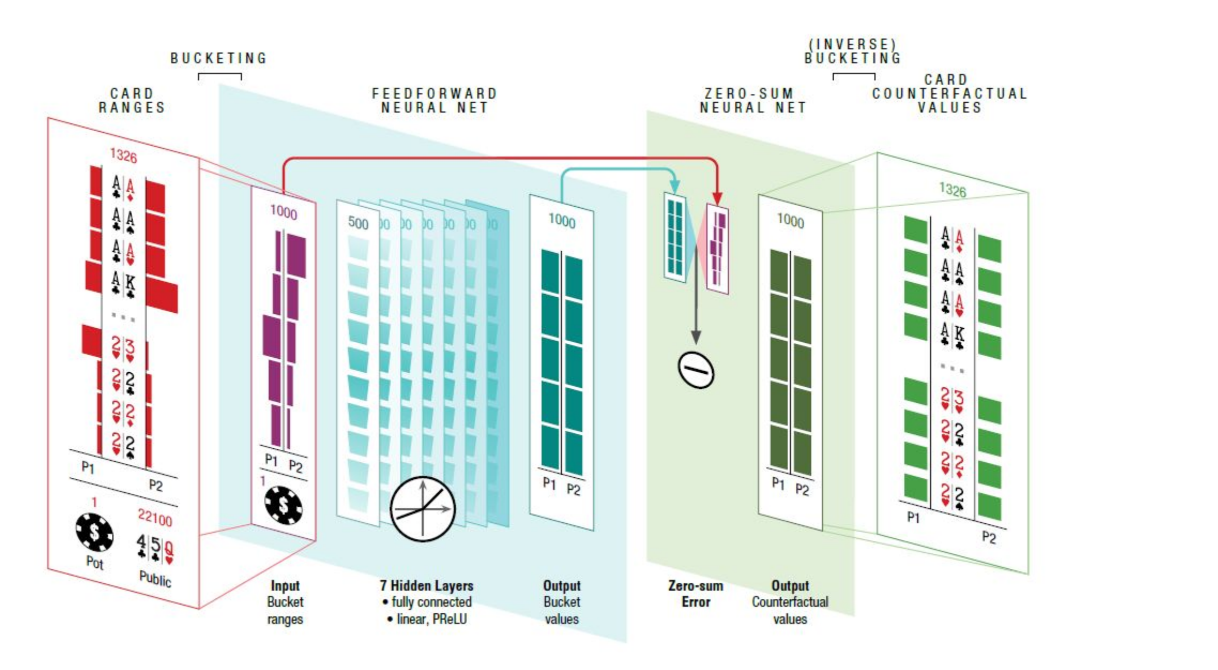
Terminal

Pseudo-terminal



GTO Wizard

DeepStack's Value Network (Moravčík et al., 2017)



GTO Wizard

Figure from DeepStack **(Moravčík et al., 2017)**

DeepStack (Moravčík et al., 2017)

While DeepStack showed a general framework to do search in imperfect-information games, its performance was unclear.

- Evaluated against humans that were not specialized in heads-up poker. Matches were done in an online setting with low financial incentives.
- Not evaluated against other poker bots.
- A reimplementation of DeepStack actually found it to be losing substantially to Slumbot, the last winner of the ACPC (Zarick & al, 2020)
- Still used a lot of domain knowledge to achieve high performance
- Required multiple steps (far from end-to-end learning)

Libratus (Brown & Sandholm, 2017)

Search with blueprint strategies

- Precomputes the strategy for a very large abstracted version of the game ($100,000, 50 TB), the blueprint strategy
- Uses search to refine this strategy at inference time when getting “near” the end of the game.
- Safe solving techniques to deal with off-tree actions
- Superhuman performance against top professionals in Heads-up NLH

GTO Wizard

Pluribus (Brown & Sandholm, 2019)

- Superhuman performance in multiplayer poker
- Cost only $150 to train!

High-level summary:

- Pre-computes a blueprint strategy to estimates the values of different states of the game.
- Blueprint strategy was computed with state-of-the-art abstraction techniques
- Uses a novel search technique, based on multi-valued states, where a player is allowed to switch between different strategies beyond the depth-limit

Issues with blueprint strategies

Building a sound blueprint strategy like Libratus/Pluribus is impossible in more realistic settings where the initial state of the game is constantly changing every hand.

- Poker is a family of games → We'd need to pre-compute thousands of blueprint strategies.
- That would be an engineering nightmare to implement since blueprint strategies can be large.
- Complex players needs - Players want to customize the game tree when studying, e.g. by making assumptions about their opponents (node-locking) or removing/adding actions at certain nodes.

→ You likely won't find any state in the blueprint strategies that will be similar to the state that the player is studying

Solution - Learning

The elegant solution to this problem is to use function approximation with deep neural networks. Instead of pre-computing values via a blueprint strategy, we'll ask a neural network to predict them during search.

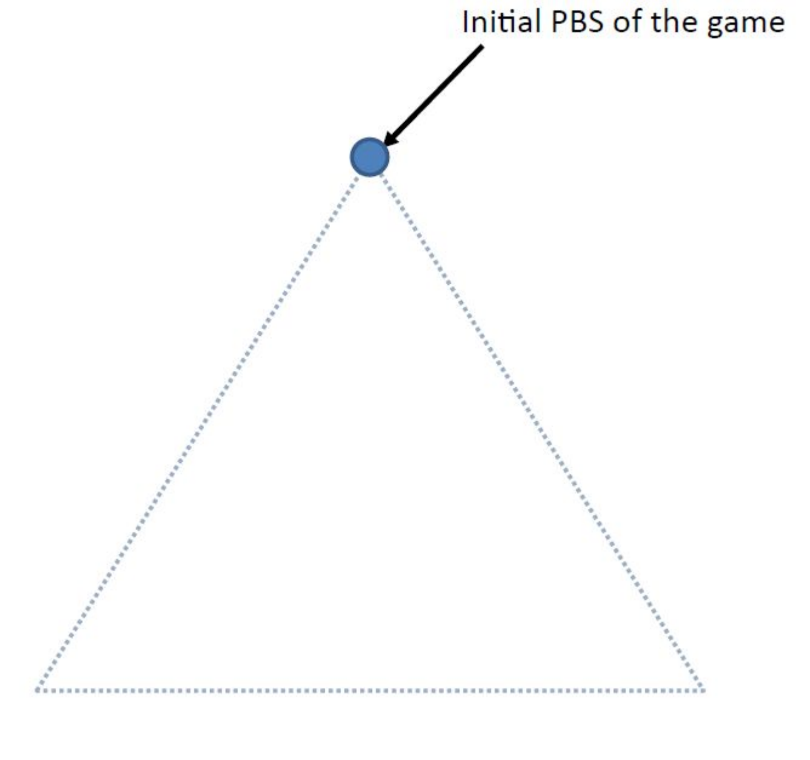
- ➔ If the neural network is trained on a large and diverse dataset, it can do a good job of interpolating any state that we can throw at it
- ➔ Aligned with the methodology of DeepStack, ReBeL, and others
- ➔ It took until 2020 for researchers to convincingly show that this approach could achieve the same level of performance as Libratus

ReBeL (Brown et al., 2020)

- General approach for self-play reinforcement learning and search for imperfect-information games
- The algorithm reduces to an algorithm similar to AlphaZero in perfect-information games
- Uses neural networks to predict values at the depth-limit, like DeepStack
- Doesn’t use any domain knowledge in the form of information abstraction
- Achieves superhuman performance
- Along with Supremus (Zarick & al., 2020), ReBeL proved that Deep Counterfactual Value Networks can achieve very strong performance, which was doubted by many at the time

ReBeL (Brown et al., 2020)

Generate a subgame rooted at the start of the game and solve it



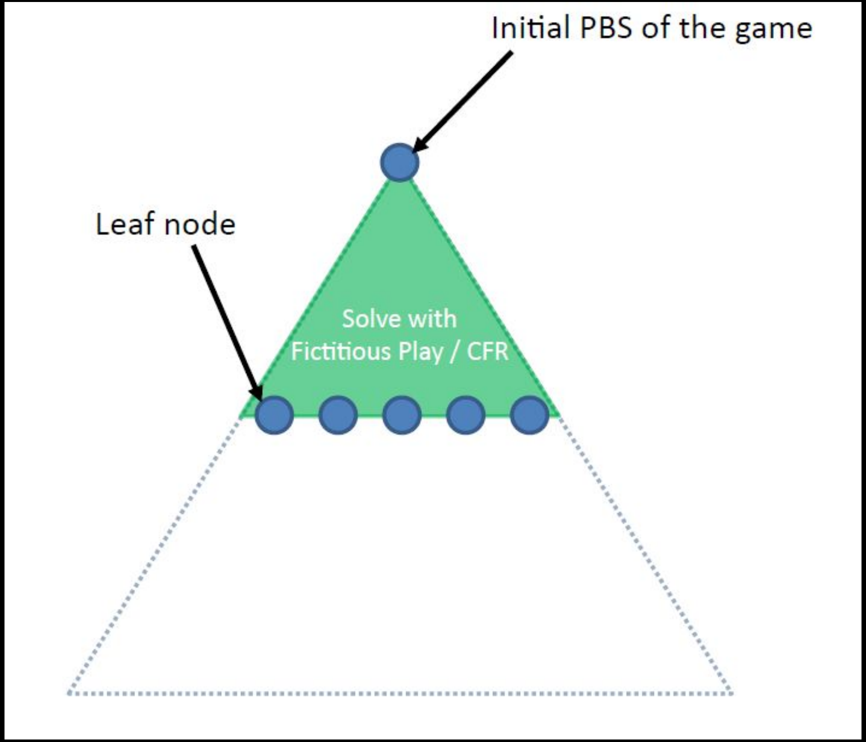
Taken from (Brown, 2020)

GTO Wizard

ReBeL (Brown et al., 2020)

Generate a subgame rooted at the start of the game and solve it

- Solve using CFR
- Leaf values come from value network
- Take next action

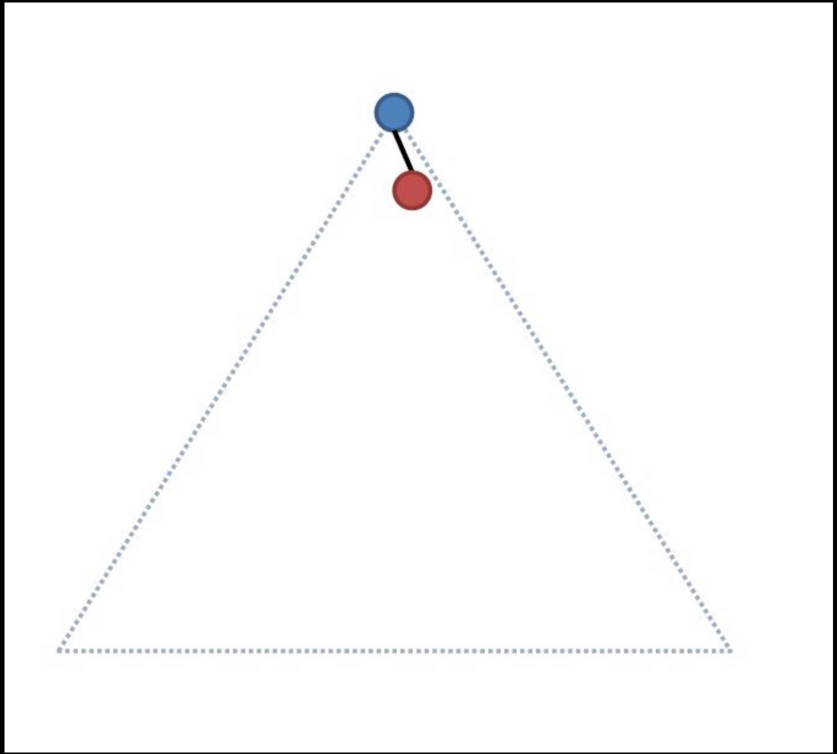


Taken from (Brown, 2020)

ReBeL (Brown et al., 2020)

After an agent's action, generate a subgame and solve it

- Solve using CFR
- Leaf values come from value network
- Take next action



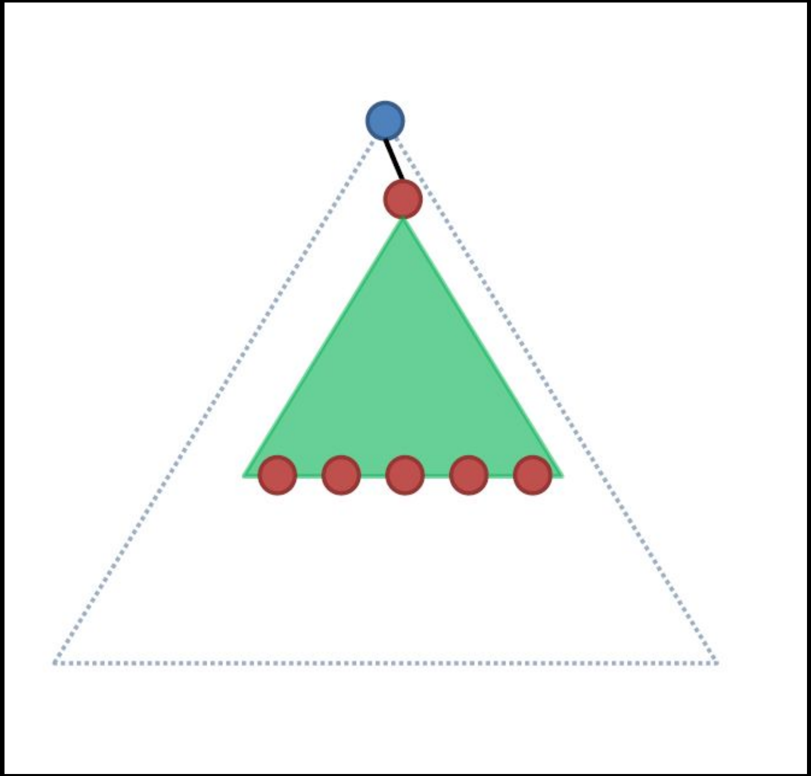
Taken from (Brown, 2020)

ReBeL (Brown et al., 2020)

After an agent's action, generate a subgame and solve it

- Solve using CFR
- Leaf values come from value network
- Take next action

Repeat until end of the game



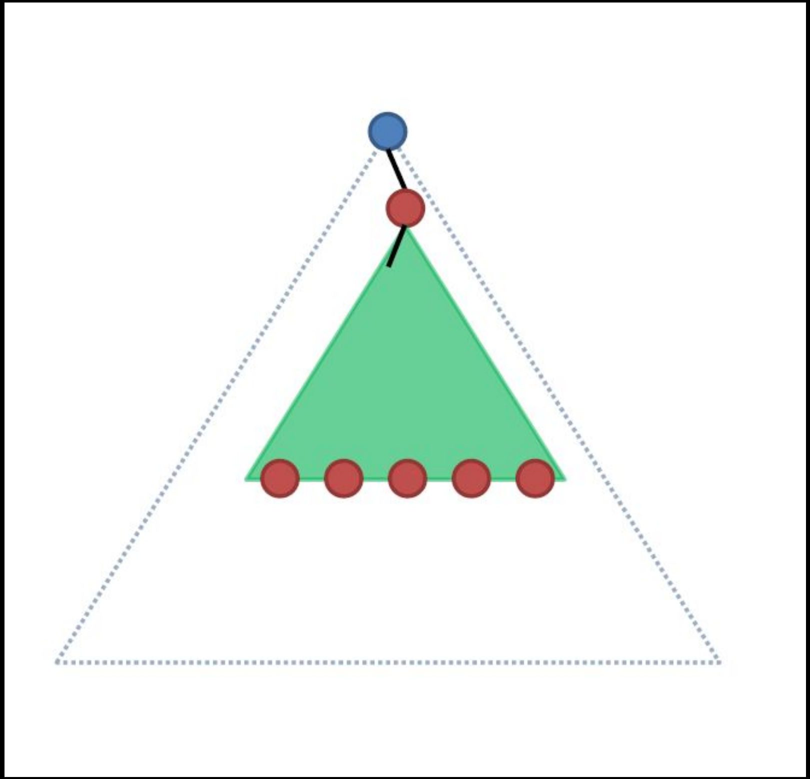
Taken from (Brown, 2020)

ReBeL (Brown et al., 2020)

After an agent's action, generate a subgame and solve it

- Solve using CFR
- Leaf values come from value network
- Take next action

Repeat until end of the game



Taken from (Brown, 2020)

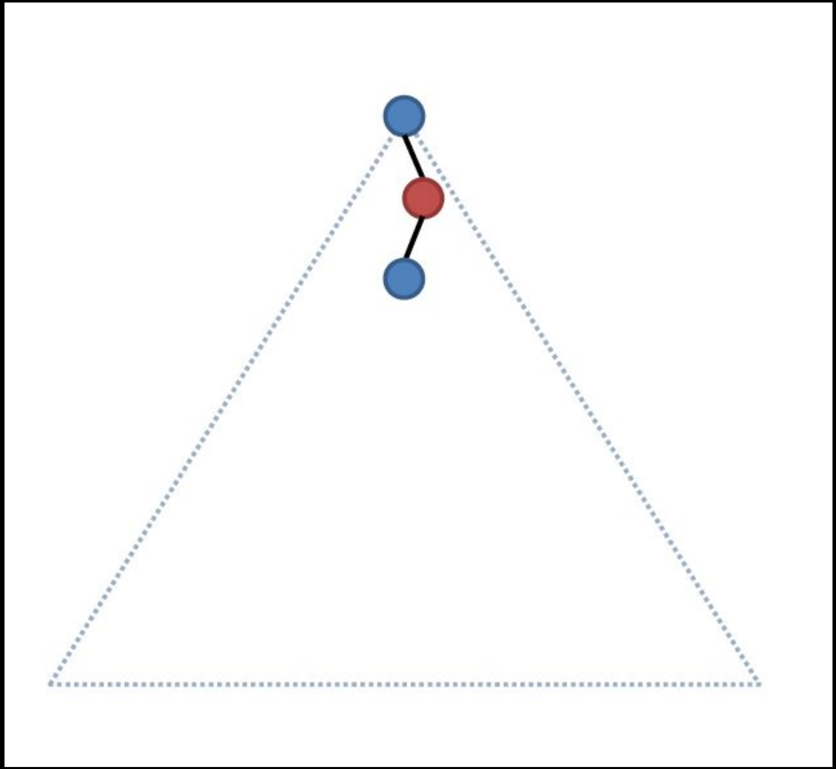
GTO Wizard

ReBeL (Brown et al., 2020)

After an agent's action, generate a subgame and solve it

- Solve using CFR
- Leaf values come from value network
- Take next action

Repeat until end of the game



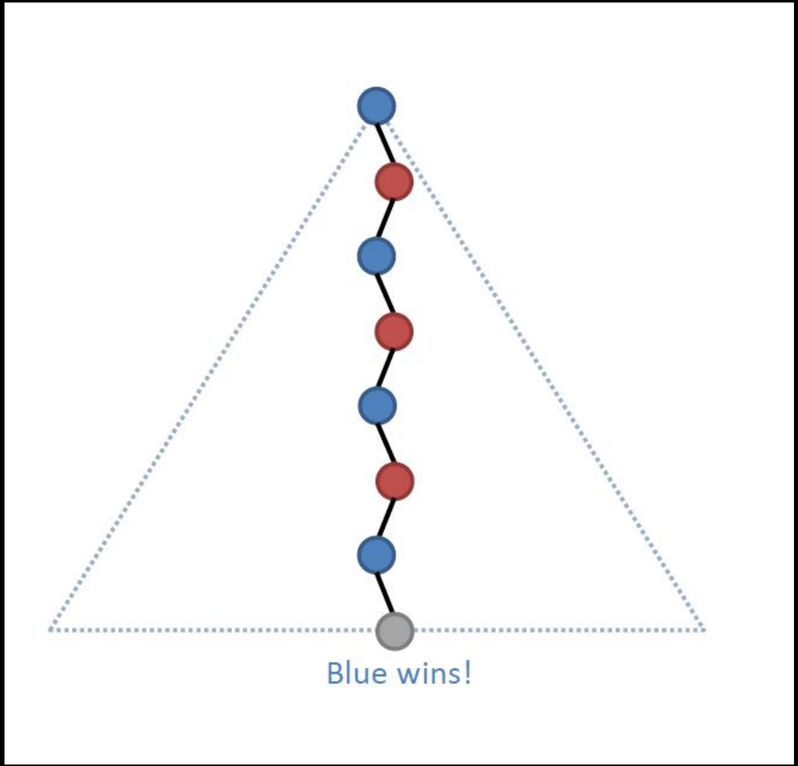
Taken from (Brown, 2020)

ReBeL (Brown et al., 2020)

After an agent's action, generate a subgame and solve it

- Solve using CFR
- Leaf values come from value network
- Take next action

Final values are used as a training sample for the value network



Taken from (Brown, 2020)

GTO Wizard AI

- Pushing the state-of-the-art of AI research in imperfect-information games
- Uses reinforcement learning and search, similarly to ReBeL
- Engineering and algorithmic improvements to achieve state-of-the-art performance, and running on CPU in a few seconds
- Can solve any stack size for 2-player games, with any rake format, by leveraging the function approximation power of deep neural networks
- Support for solving under the Independent Chip Model (ICM), which is the most accepted model for translating chips into dollars in tournaments

GTO Wizard AI

Agent	Slumbot (2018)	Top Humans
DeepStack's reimplementation (2020)	-6.3 ± 4.0	
ReBeL (2020)	4.5 ± 0.5	16.5 ± 6.9
Supremus (2020)	**17.6 ± 4.4**	
GTO Wizard AI (2022)	**19.4 ± 4.4**	

Table 2: Head-to-head results showing expected winnings (bb/100).
The ± shows one standard deviation.

GTO Wizard AI

Results vs. Slumbot across 150k hands

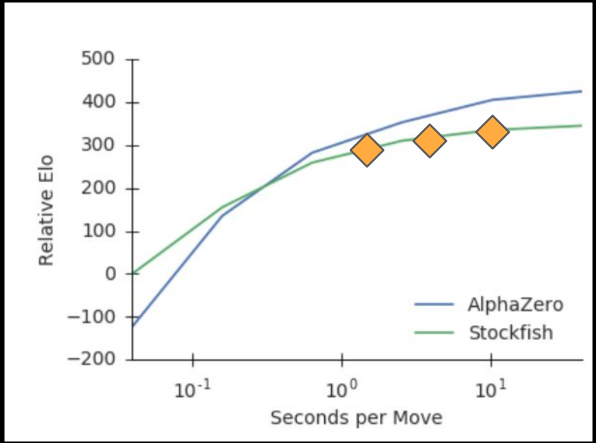
GTO Wizard AI

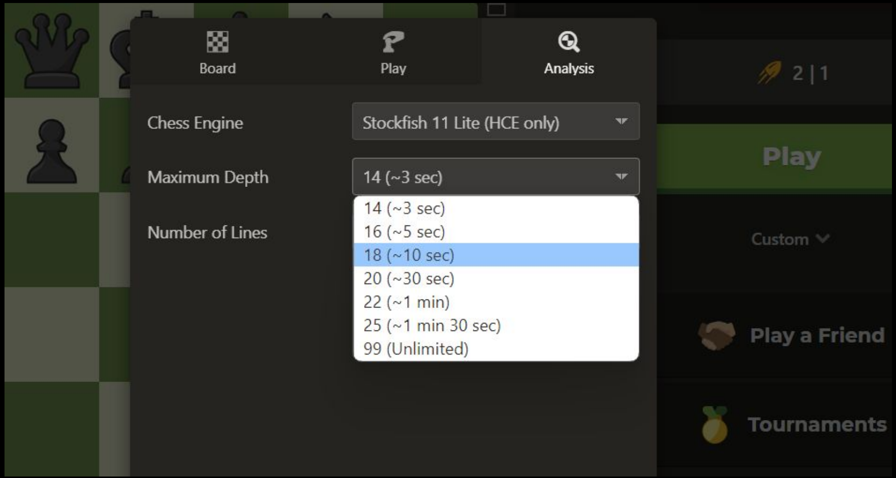
	Median thinking time (s)		
Round	Humans	DeepStack	GTO Wizard AI
Flop	9.1	5.9	2.0
Turn	8.0	5.4	1.4
River	9.5	2.2	1.6

Median thinking time in seconds. Thinking time for humans are taken from the DeepStack paper.

What's next?

Depth-limited solving with a dynamic/growing depth-limit so that users can choose different solutions on the speed-accuracy trade-off curve





Chess.com's approach to setting the thinking time - Maximum Depth

What’s next?

- Scale our algorithms to up to 9 players - Exciting releases soon coming in that direction
- Scale our algorithm to larger games:
  - Omaha: 270,725 or 2,598,960 possible hands instead of 1326 for Hold’em
  - Drawing games like Seven Card Stud, where you can draw up to 7 cards, for a total of 133,784,560 possible hands!

→ Solve any poker game to a high accuracy in a few seconds

Thank you!

There's still a lot of exciting research and development to be done in the field.

If being at the forefront of technology sounds interesting to you, please reach out to me at
phil@gtowizard.com

GTO Wizard

Giveaway Winners!

Thanks for coming!

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

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

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

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

GTO RSVP Raffle: kerbs “nxliu” (1st) and “zamanova”