Pokerbots 2024

Lecture 7: GTO Wizard AI

Sponsors

HRT

CITADEL
CITADEL | Securities

DRW

Jane Street

FIVE RINGS

jumptrading

TREX QUANT

SEVEN EIGHT CAPITAL, LP

SIG
SUSQUEHANNA

HAPCAPITAL

CTC
CHICAGO TRADING COMPANY

2σ
TWO SIGMA

akuna capital

Announcements

Lightning Tournament 3 Last Night

Strategy Proof Turngate	$400
DKE Sophomores	$300
ssad_people	$225
The Fish	$175
Curtis James Jackson III	$150

Team Strategy Reports

Due Yesterday at 11:59pm

- Required to pass this class
- Communicate with us if you haven’t submitted yet (we tend to be lenient on extensions)

Today (Wednesday 1/31)

- GTO Wizard Guest Lecture (Now)
- GTO Wizard Poker Social (immediately after)
- Final bots must be uploaded by 11:59pm

Friday 2/2

Pokerbots Final Event - 4:30-7pm in Kresge

- 4:30-6:00 Sponsor Networking Session and Puzzle Contest in Kresge Lobby
- 6:00-7:00 Awards Presentation and Closing Ceremony in Kresge Auditorium

RSVP at pkr.bot/rsvp to be eligible for dinner, swag, raffle, Pokerbots T-Shirts, and more!

Saturday 2/3

GTO Wizard Poker Tournament- 5pm in BC Porter Room

- $3500 Cash Prize Pool
- Limited to 150 - RSVP to secure your spot @ **pkr.bot/gto**

Today’s Raffle:
RSVP to win Sony Headphones
(must be present at end of lecture to claim)

GTO Wizard AI

Marc-Antoine (Marco) Philippe

gtowizard.com

GTO Wizard

POKER SOLVER LANDSCAPE

GTO Wizard AI

About us

- Biggest educational application for poker players
- Founded 3 years ago and grew to 50 employees
- Strong community and partner with some of the biggest players in the industry

Espen Jorstad, WSOP 2022 Main Event Champion

About us

- Sponsors major events such as the World Series of Poker and Triton series

WSOP 2022 Main Event Final Table

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

	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

Table 3: Median thinking time in seconds. Thinking time for humans are taken from the DeepStack paper.

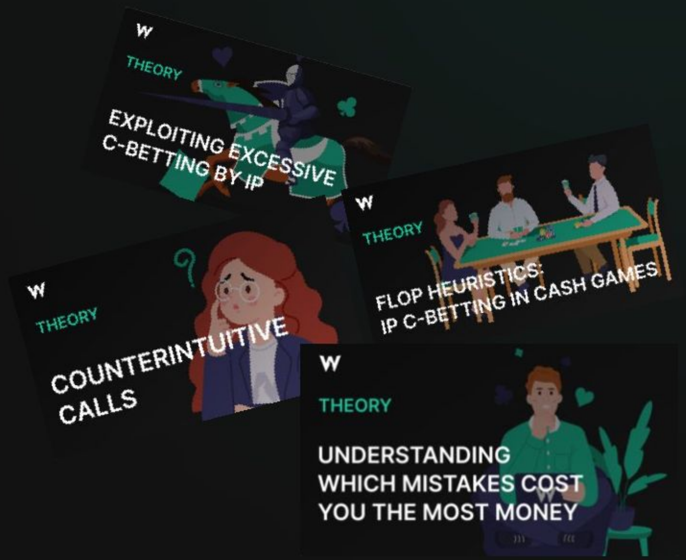
What is GTO?

Game Theory Optimal (GTO)

- Refers to a Nash equilibrium strategy
- Unexploitable and balanced strategy
- Doesn’t rely on tells or intuition
- Helps understand a baseline strategy
- Recognizing that baseline helps you know when and how to exploit mistakes

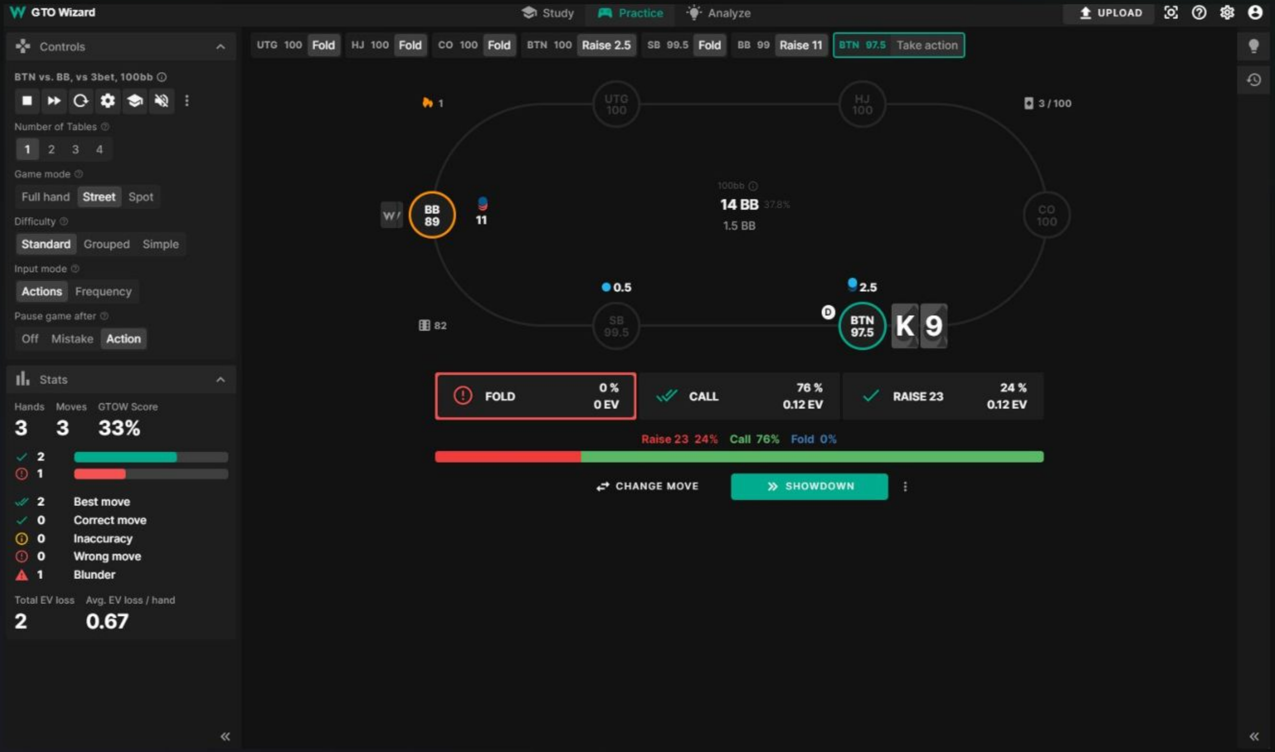
How do poker players study?

- Understand GTO strategies by finding heuristics
- Exploiting their opponents (nodelocking, MDA)
- Simple strategies with minimal exploitability
- Analyze their games to find their own leaks
- Drilling scenarios where they are weaker



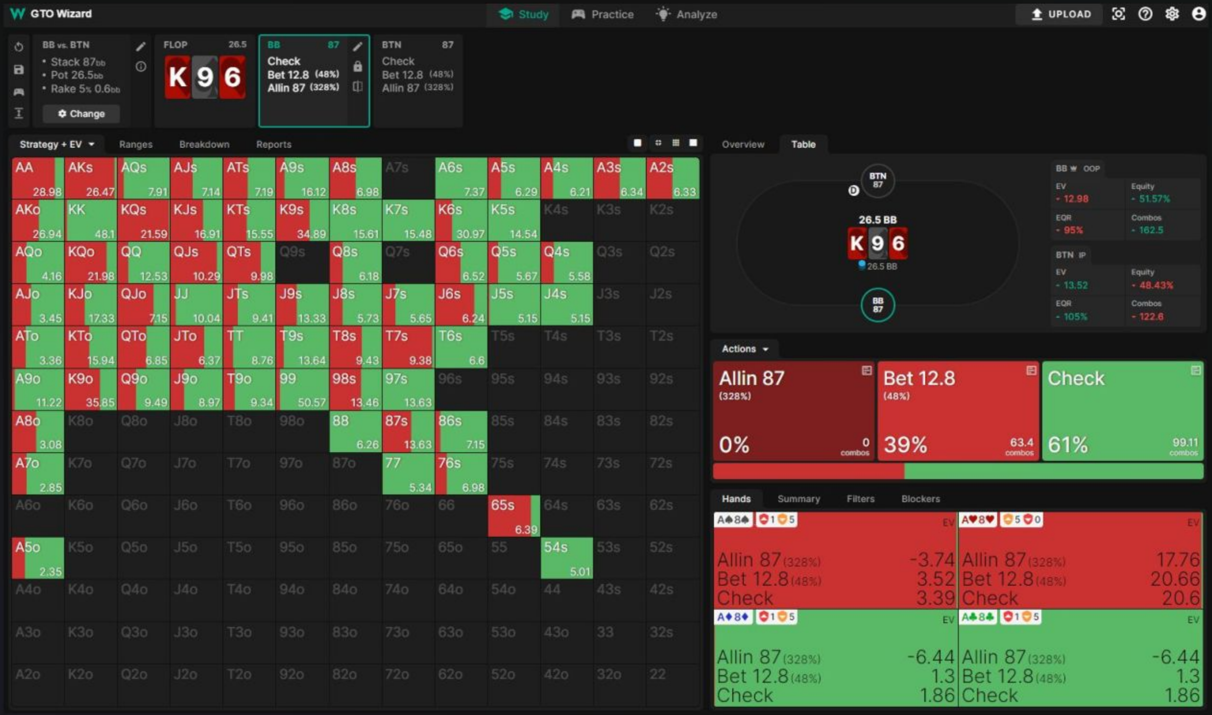
GTO Wizard's App

- 3 mains functions: Practice, Study, and Analyze



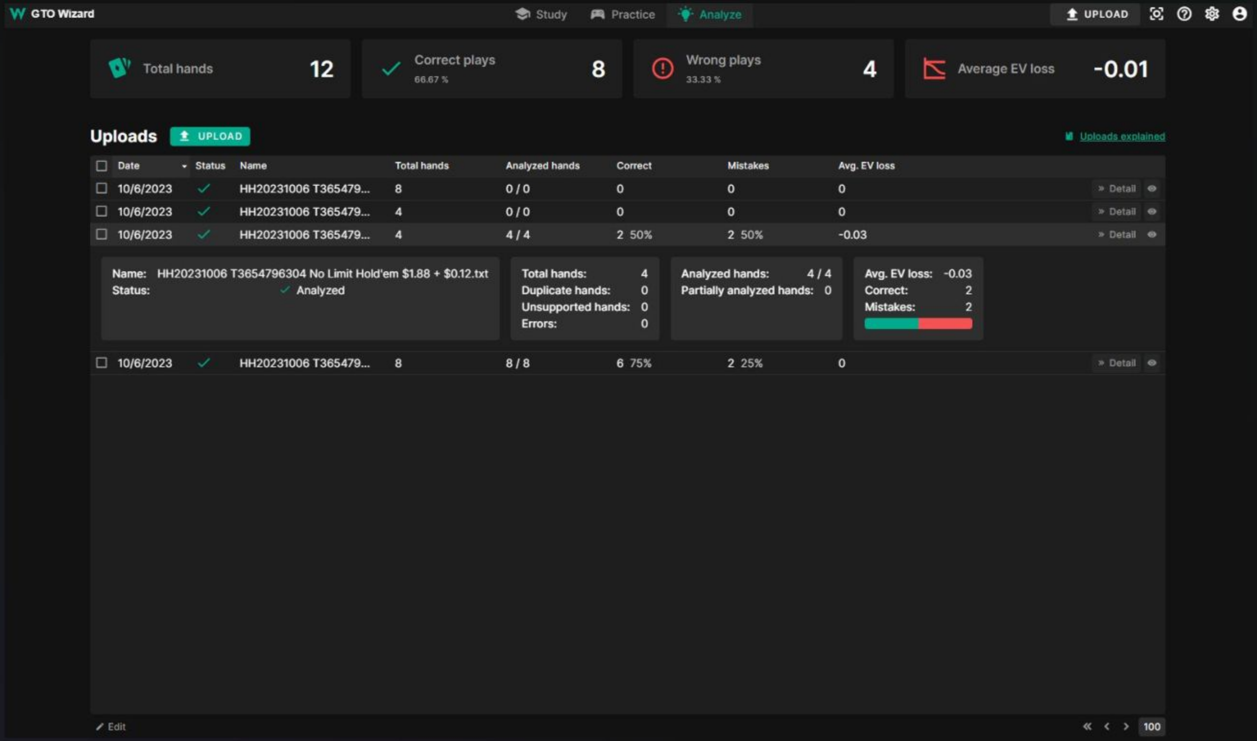
GTO Wizard's App

- 3 mains functions: Practice, Study, and Analyze



GTO Wizard's App

- 3 mains functions: Practice, Study, and Analyze



Product vision

- Solve any poker variant in a few seconds
- Turn amateurs in professional poker players
- Fast feedback loop and explainable outputs

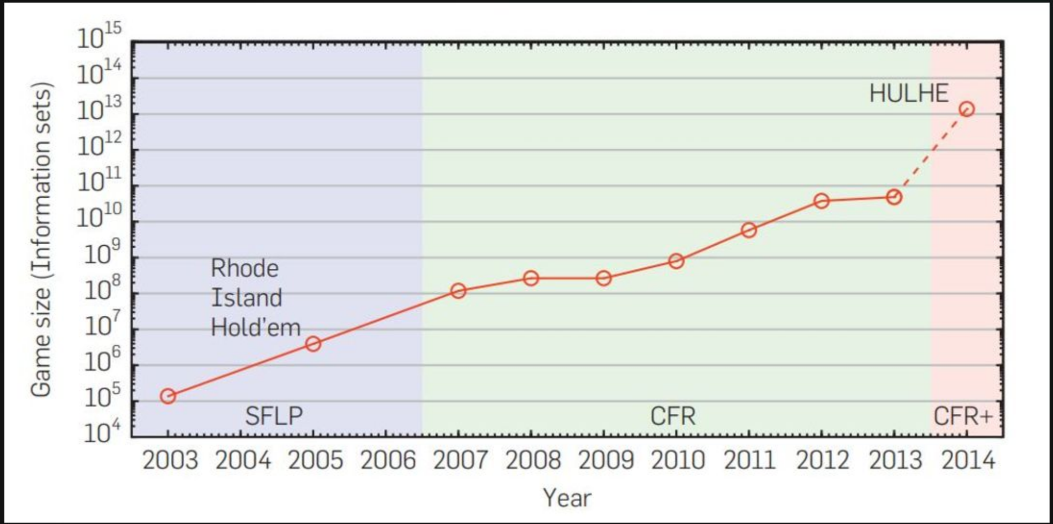
History of Poker Bots

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

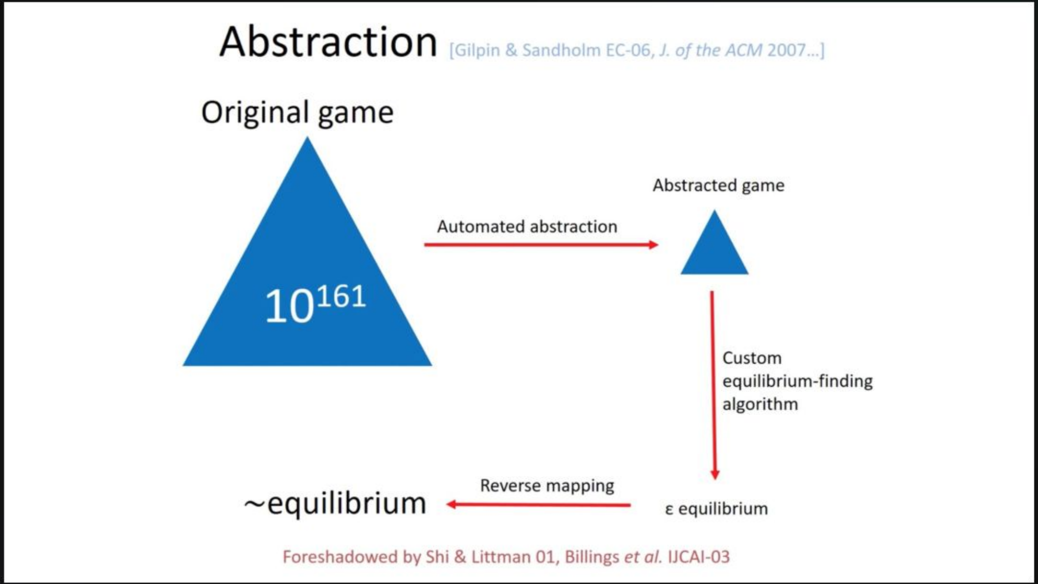


Increasing sizes of imperfect information games solved over time (Bowling et al., 2015)

Solving with Abstractions

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

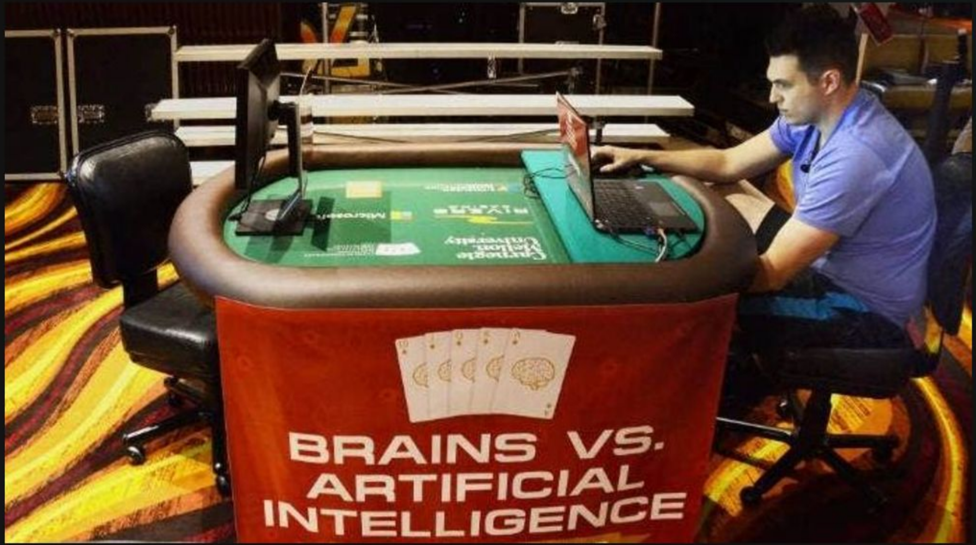
- Solve an abstracted version of the game
- Use this strategy at inference time as a lookup table



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**



Claudico Brain vs AI (2015)

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.
- Superhuman performance against top professionals in Heads-up NLH
- Safe solving techniques to deal with off-tree actions

Pluribus (Brown & Sandholm, 2019)

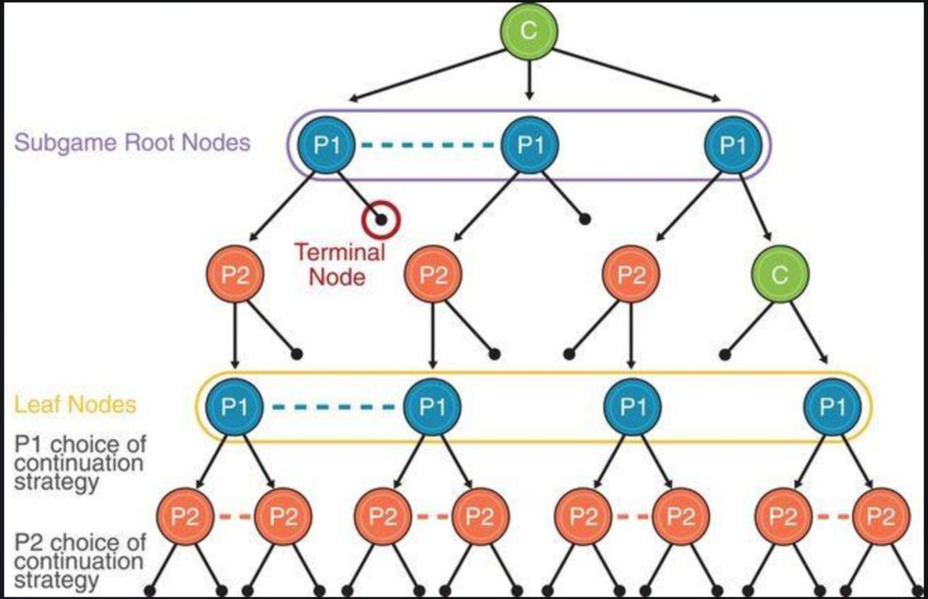
- Superhuman performance in multiplayer poker
- Cost only $150 to train

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

Pluribus (Brown & Sandholm, 2019) - Multi-valued states

Uses a novel search technique, based on **multi-valued states**, where a player is allowed to switch between different strategies beyond the depth-limit



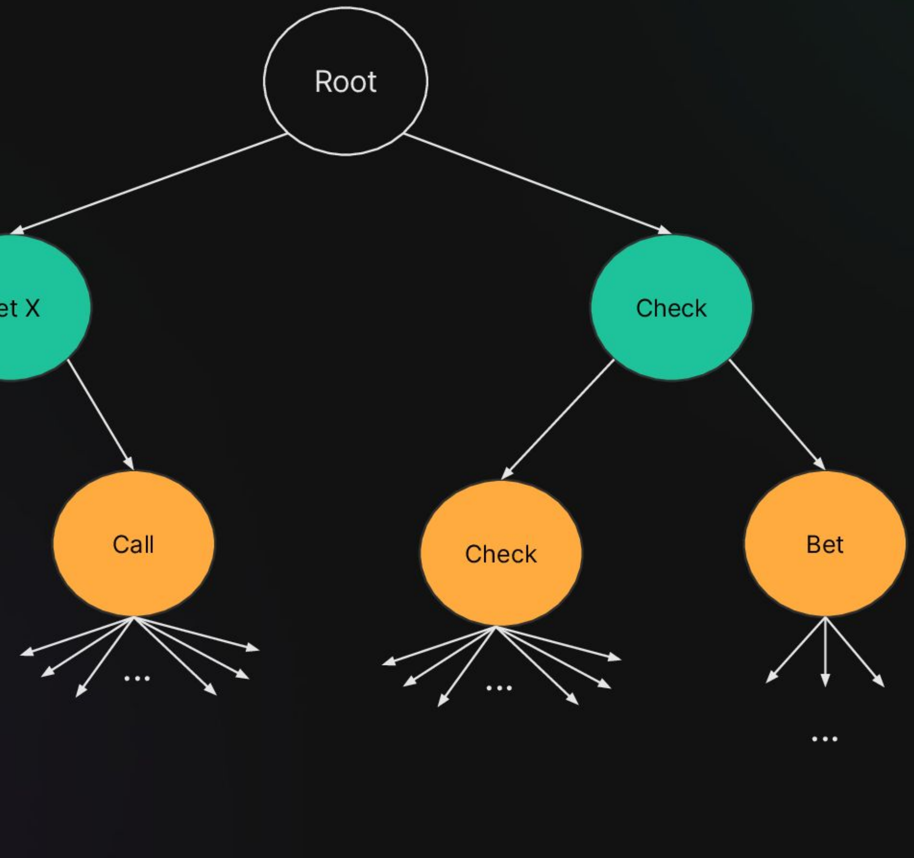
DeepStack (Moravčík et al., 2017)

Algorithm that successfully implement heuristic search in imperfect-information games

- Introduces a way to decompose the game in depth-limited subgames with CFR-Decomposition (CFR-D).
- 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”.

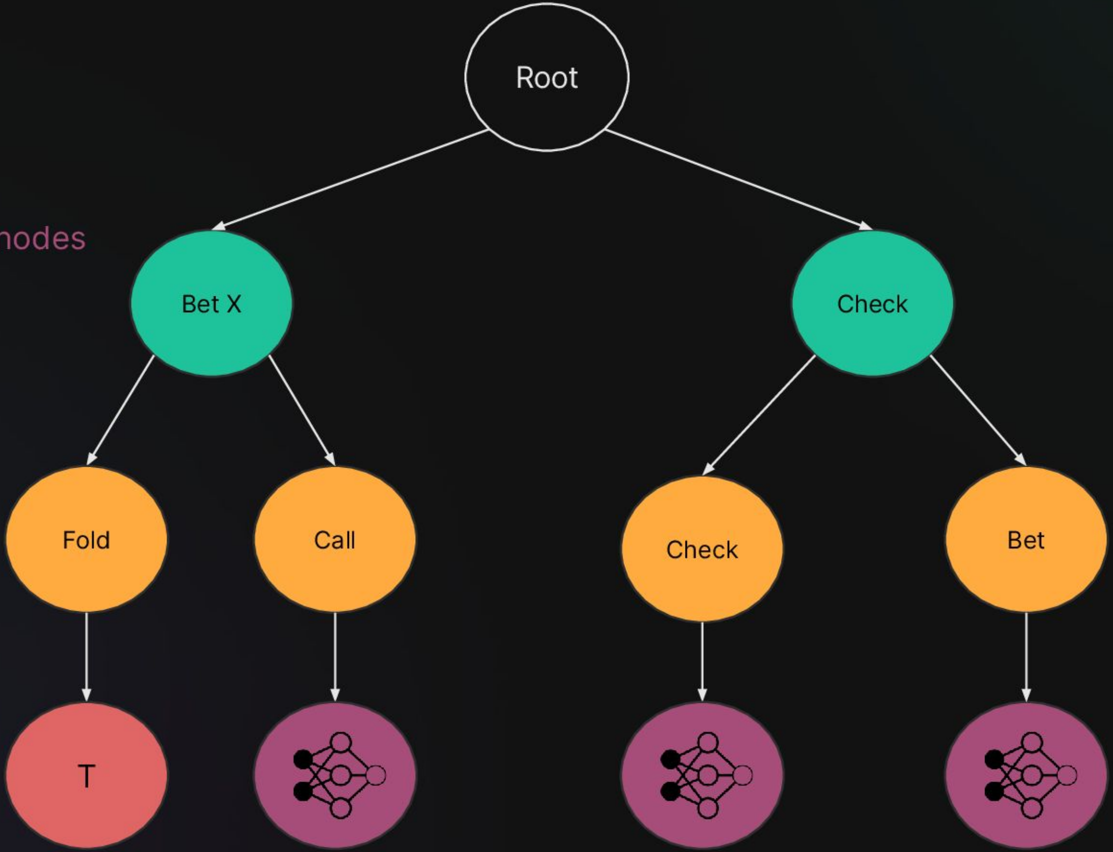
Depth-limited solving with value networks

Player 1
Player 2
Terminal nodes



Depth-limited solving with value networks

Player 1
Player 2
Terminal nodes
Pseudo-terminal nodes

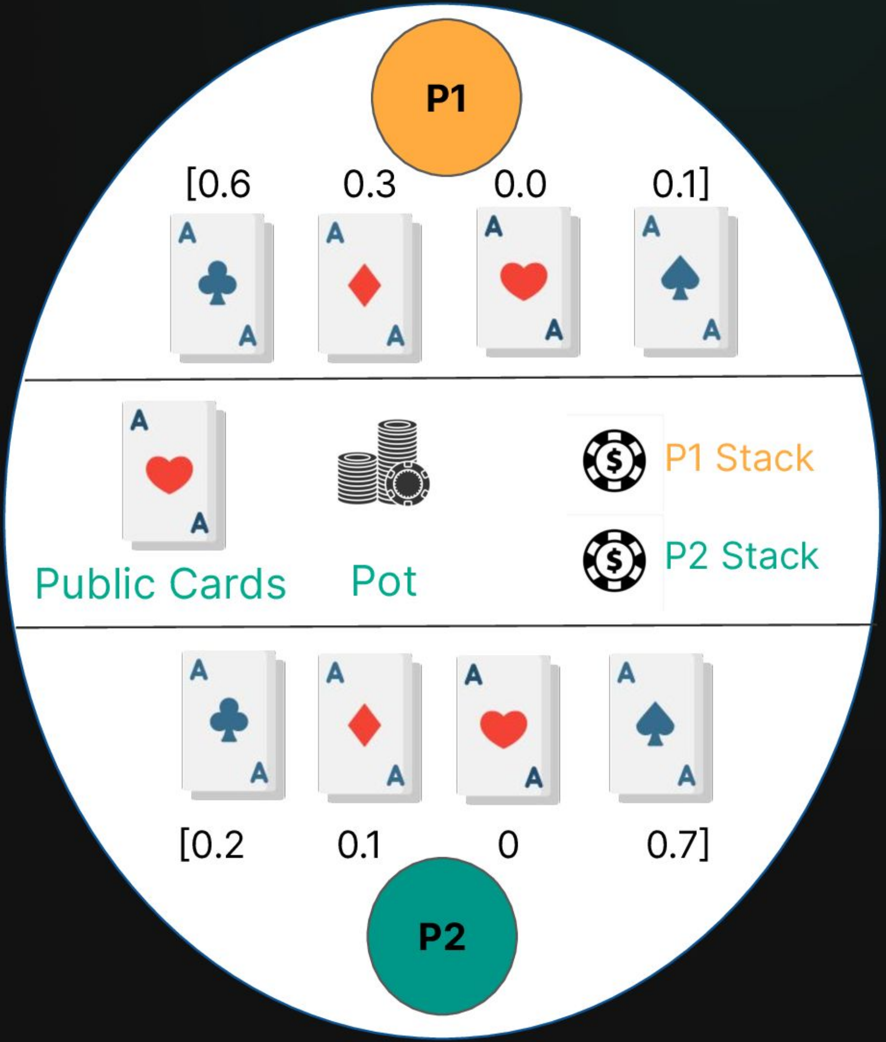


Public Belief States (PBS)

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



DeepStack (Moravčík et al., 2017) - Value Network

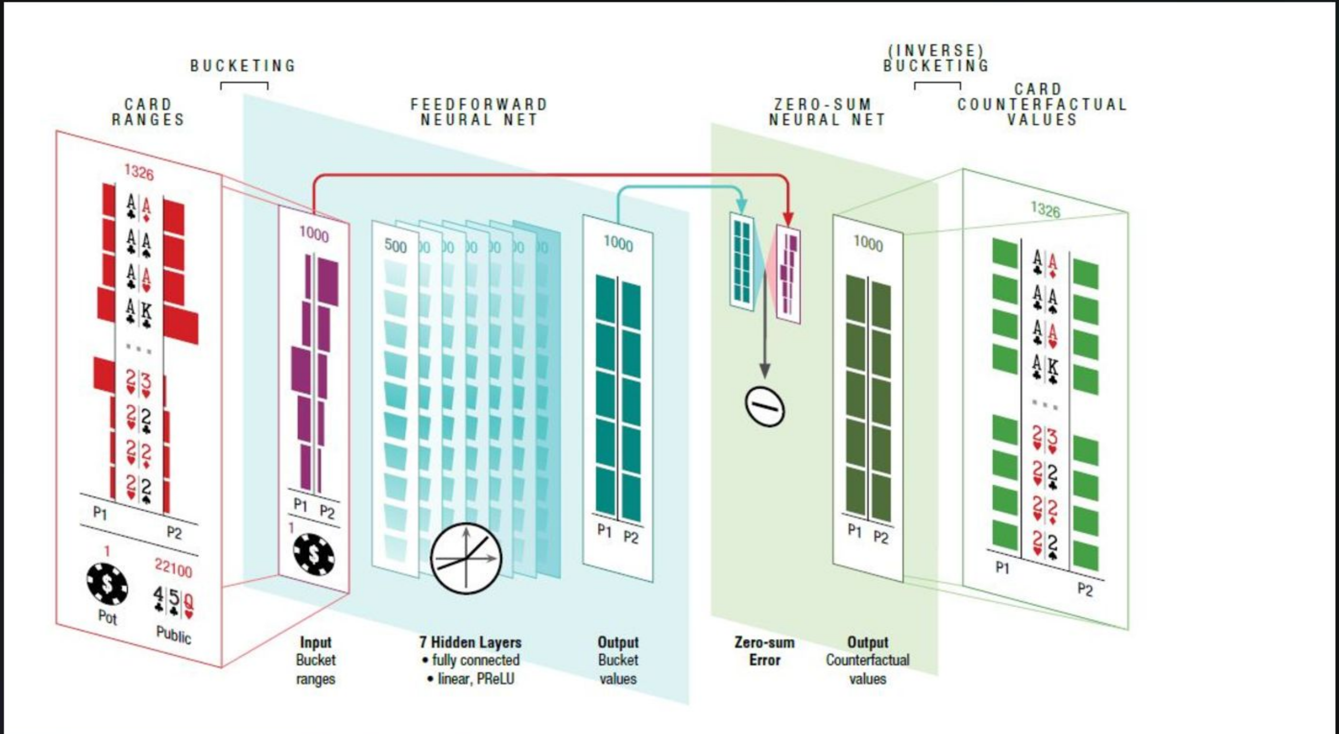


Figure from DeepStack **(Moravčík et al., 2017)**

DeepStack (Moravčík et al., 2017)

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

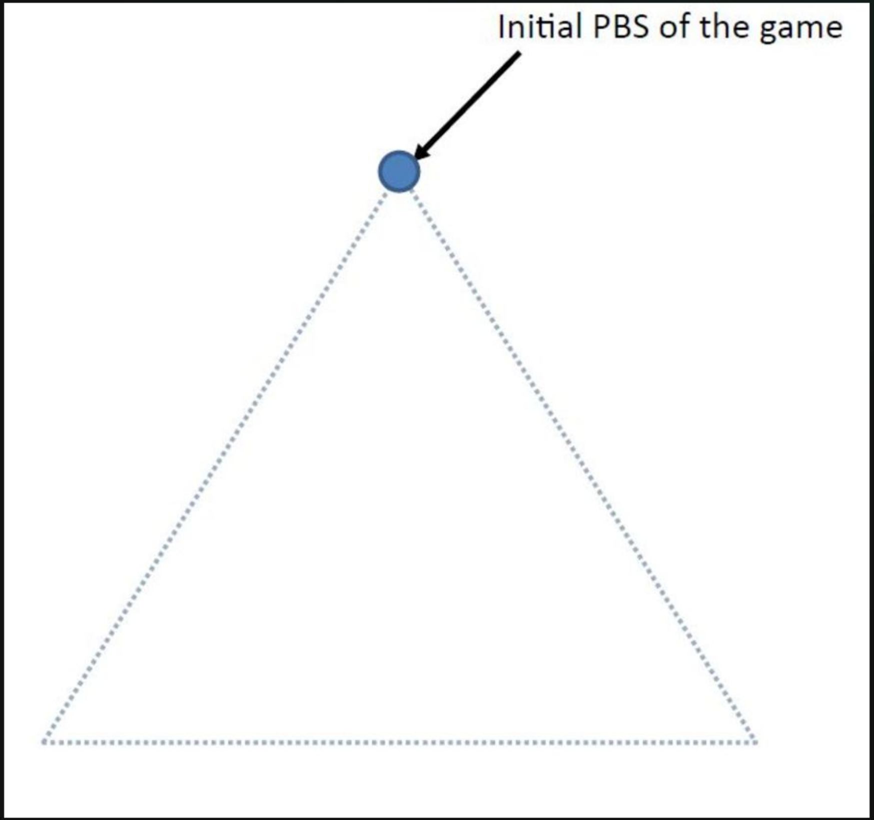
- Evaluated against humans that were not specialized in heads-up poker. Matches were done in an online setting with low financial incentives.
- Not evaluated against other poker bots.
- A reimplementation of DeepStack actually found it to be losing substantially to Slumbot, the last winner of the ACPC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

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 like abstractions
- Achieves superhuman performance
- Along with Supremus (Zarick & al., 2020), proved that *Deep Counterfactual Value Networks* can achieve very strong performance, which was doubted by many at the time

ReBeL (Brown et al., 2020)

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

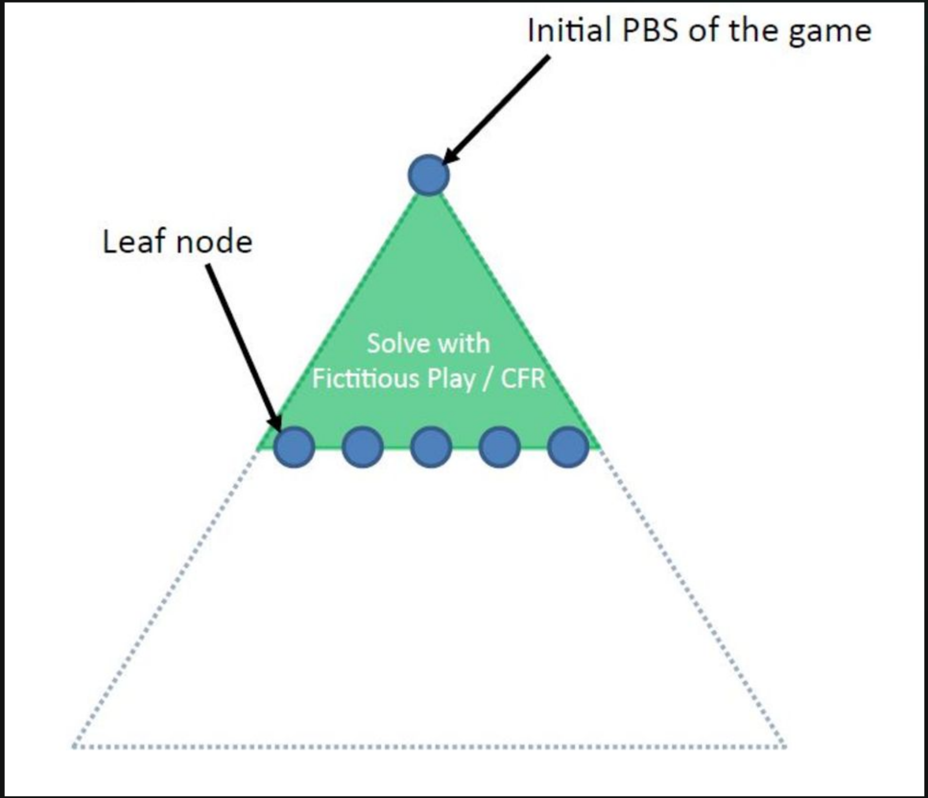


Taken from (Brown, 2020)

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

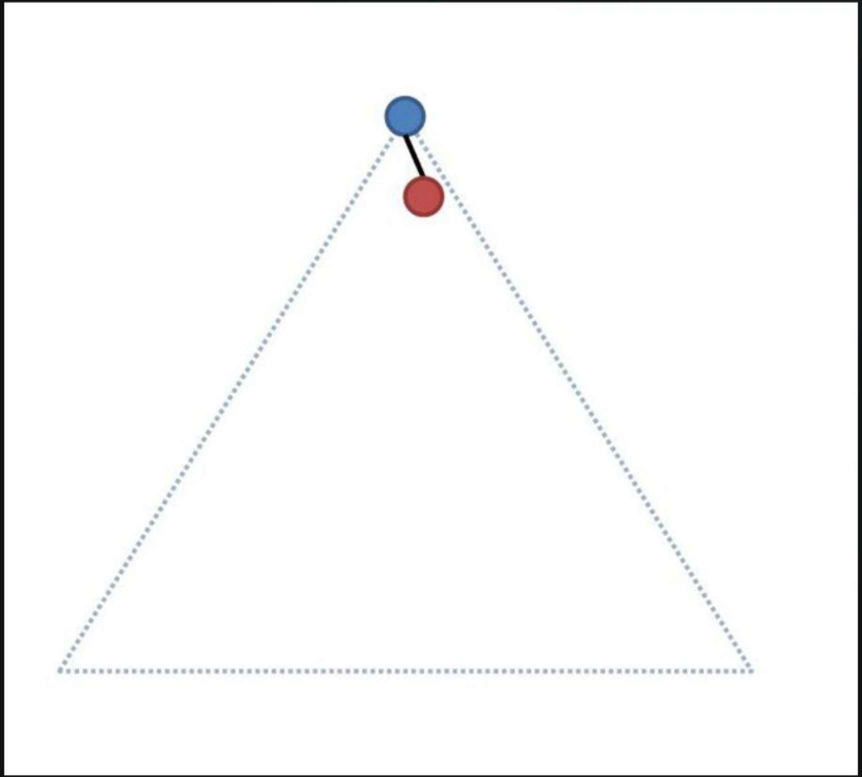


Taken from (Brown, 2020)

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



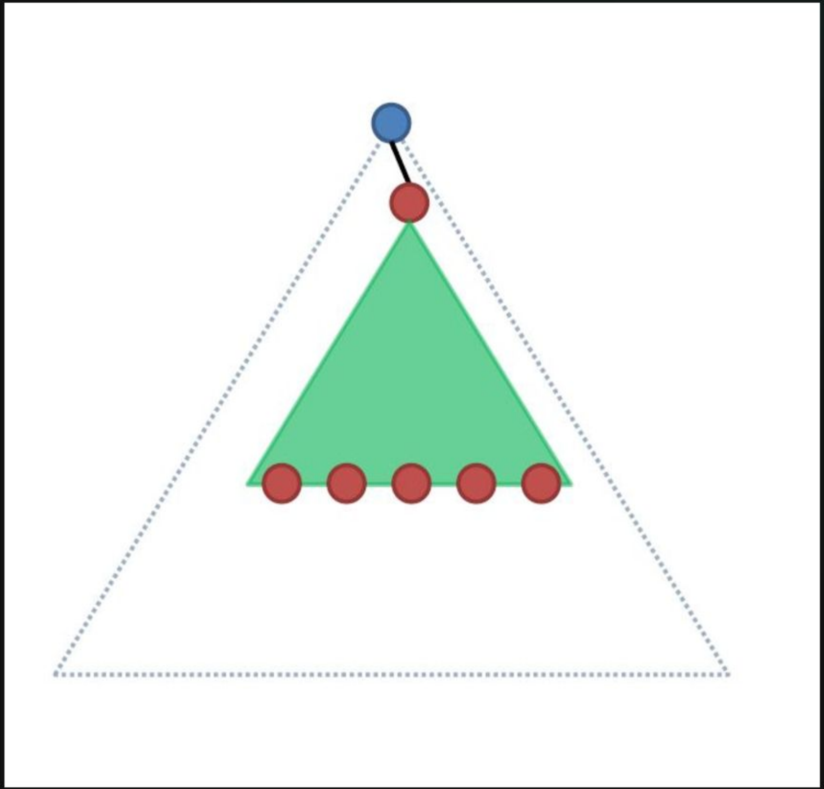
Taken from (Brown, 2020)

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



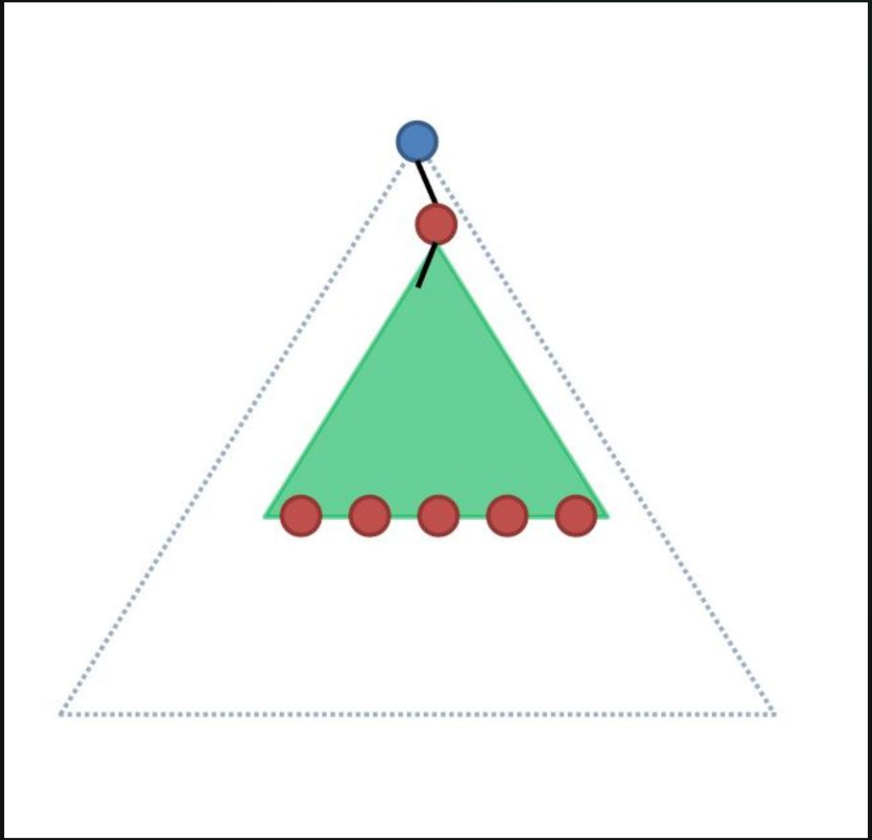
Taken from (Brown, 2020)

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



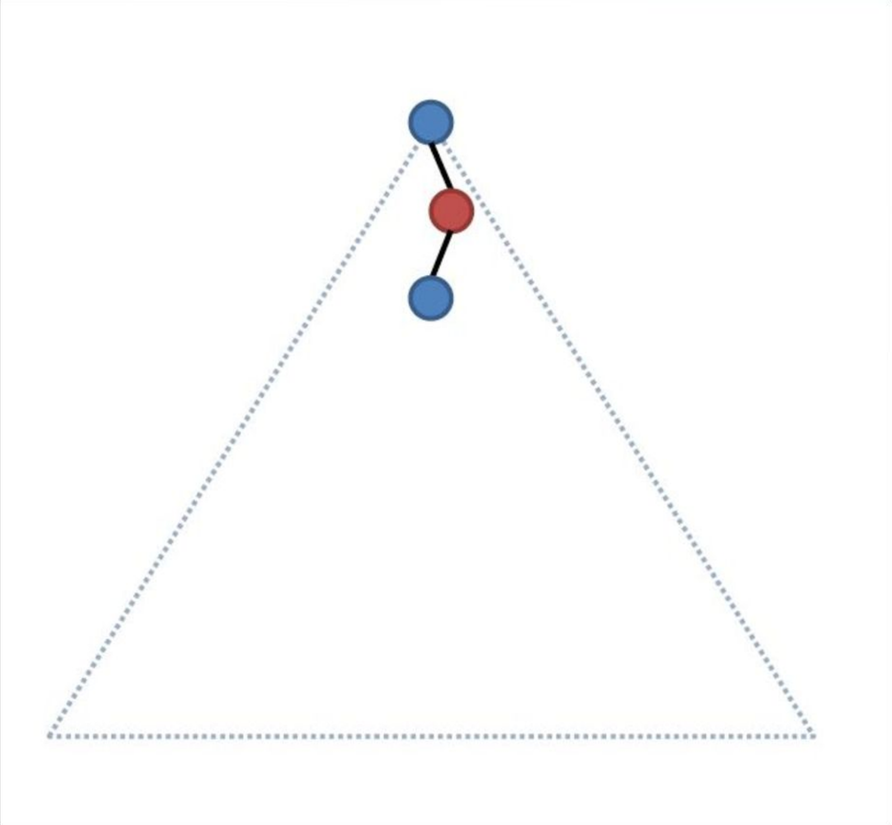
Taken from (Brown, 2020)

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



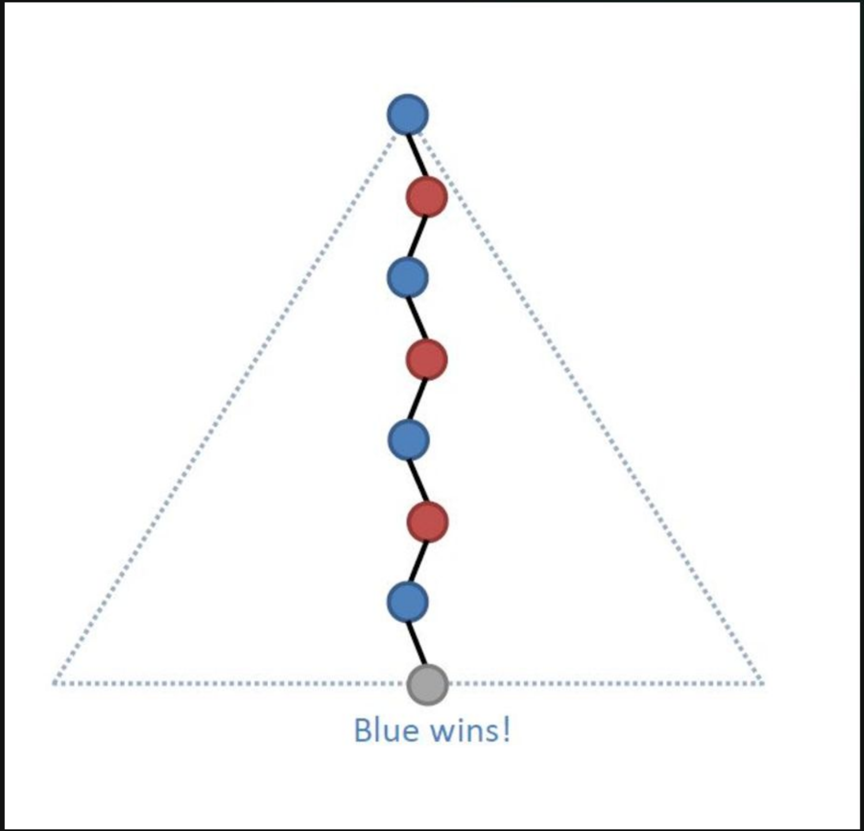
Taken from (Brown, 2020)

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)

Poker timeline

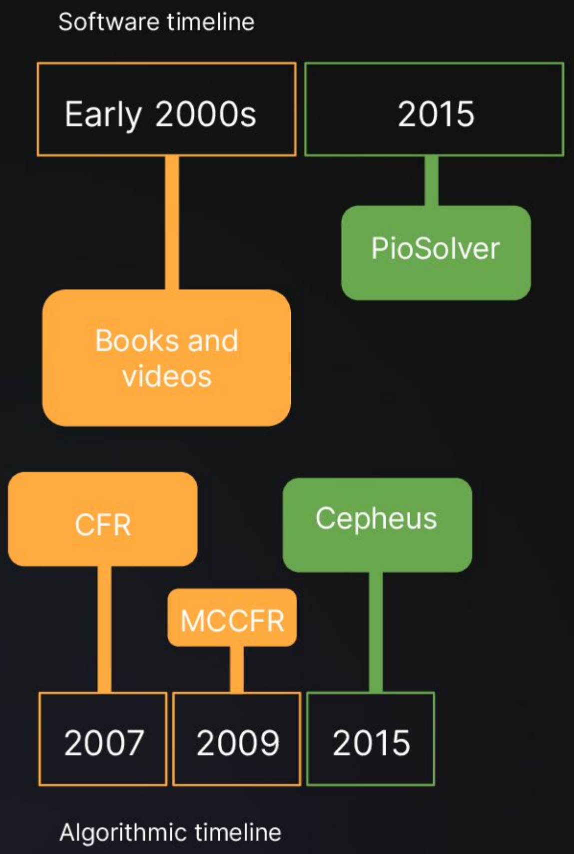
Software timeline

Algorithmic timeline

- Equity, pot odds, and instincts
- Wild west

- State-of-the-art equilibrium finding algorithms
- Algorithmic advances like CFR and MCCFR allowed us to solve much larger games

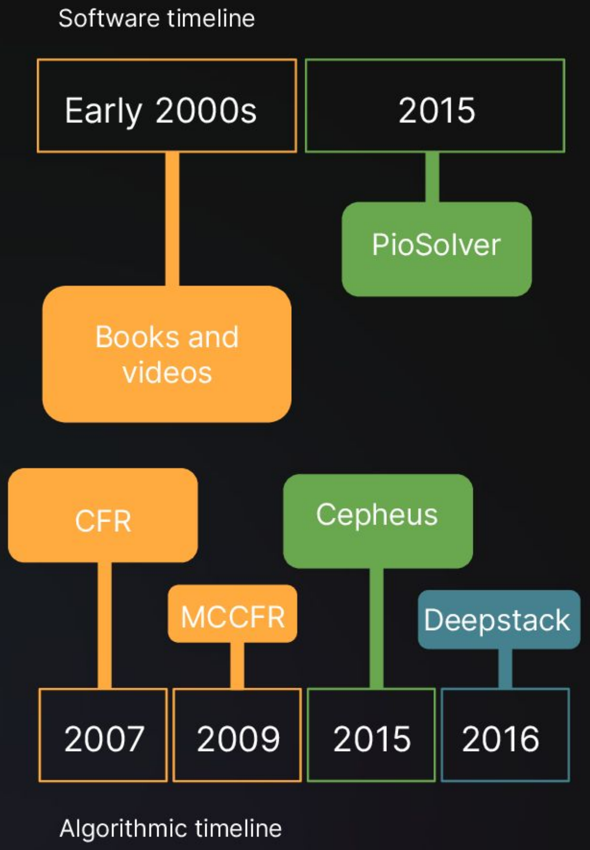
Poker timeline



- New paradigm with tabular solvers
- Dedicated servers and expensive setup
- Hours to converge (HUNL)

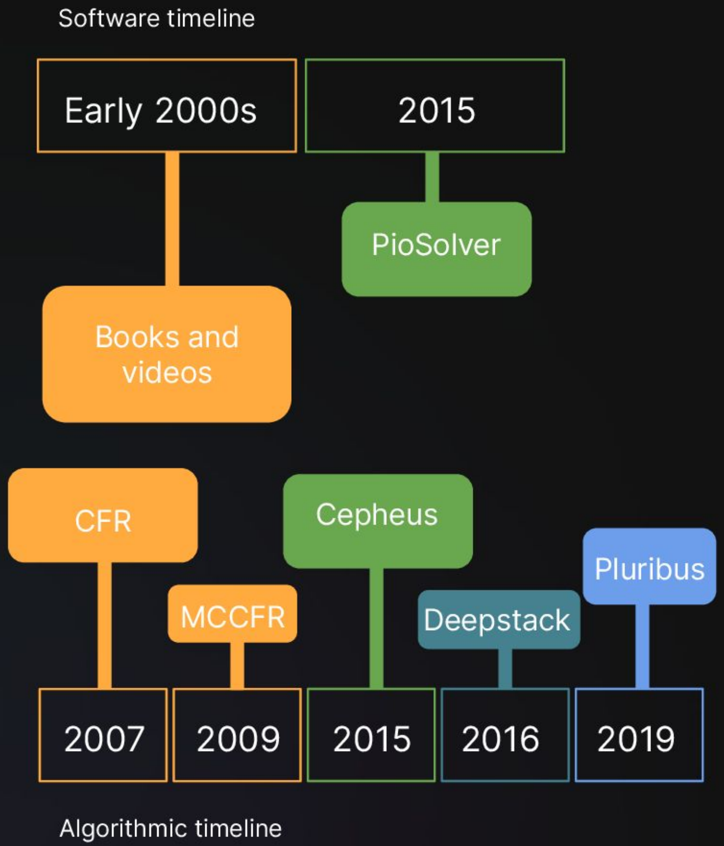
- Essentially solved Heads-Up Limit Hold'em (HULHE), the smallest variant of poker that humans actually play (10^{14} decision points)
- Storing the strategy would require 262 TiB of memory

Poker timeline



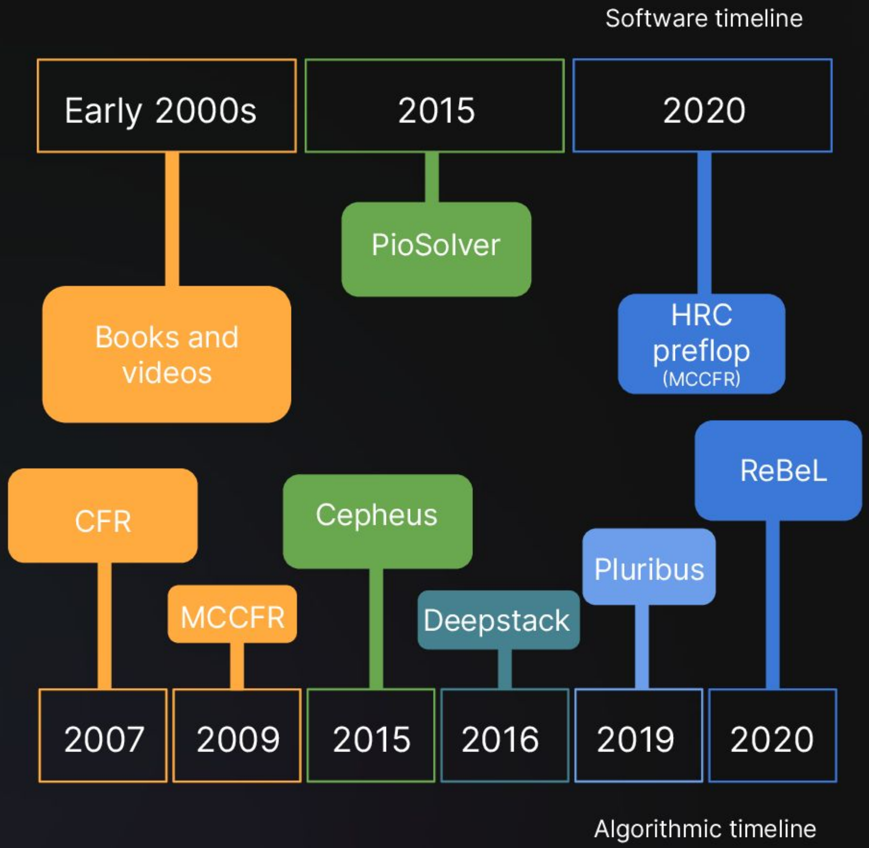
- Depth-limit CFR with neural networks
- Beats human professionals in HUNL

Poker timeline



- Beats human professionals in multiplayer poker
- Inexpensive to train

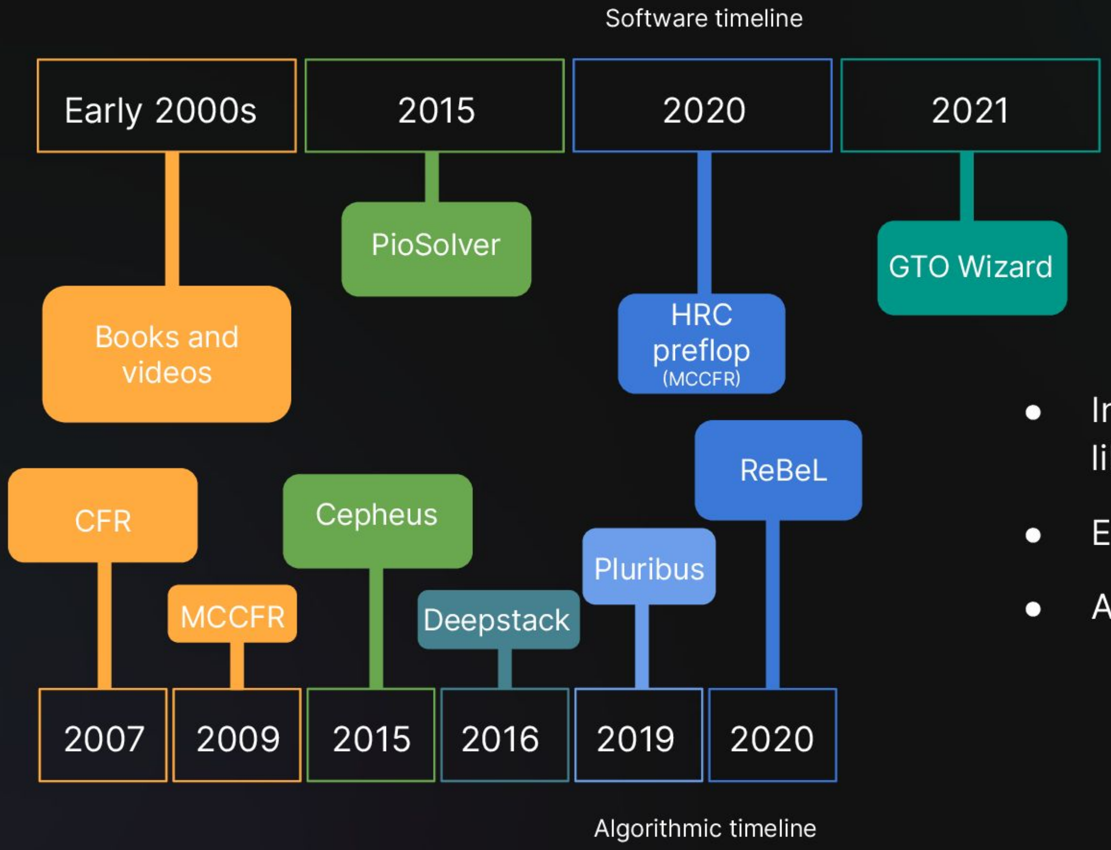
Poker timeline



- Introduction of multiway (3+ players) scenarios
- Still need expensive setup and computational intensive process

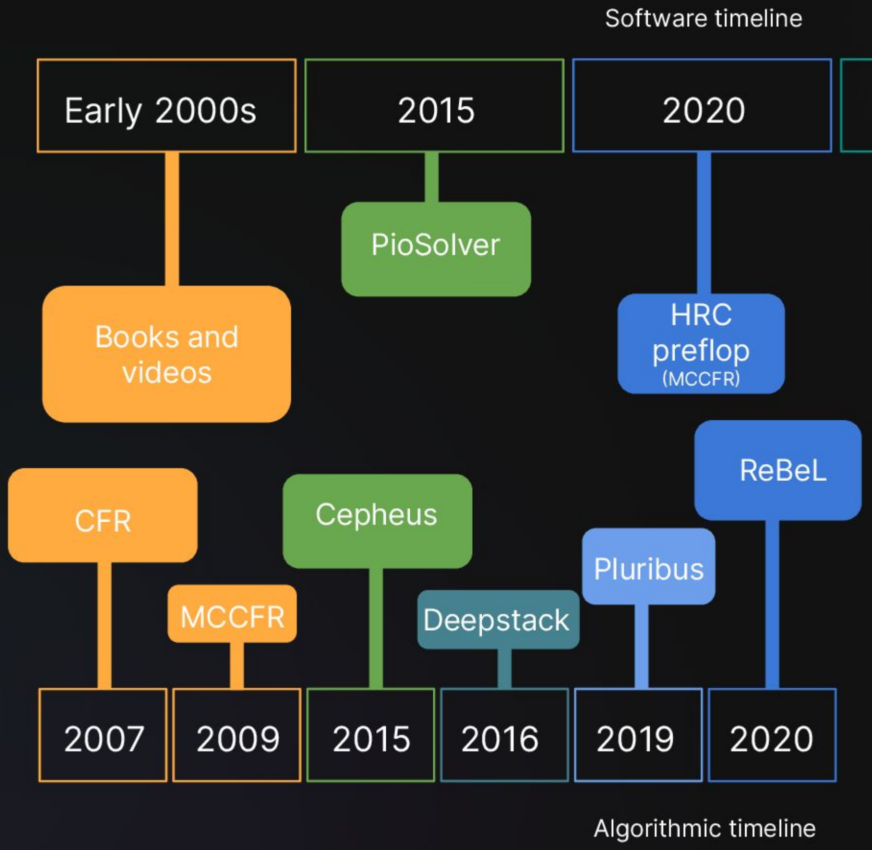
- General approach for self-play reinforcement learning and search for imperfect-information games

Poker timeline



- Introduction of pre-solved libraries
- Entirely web-based
- Accessible software

Poker timeline



- Commercial software, using depth-limited CFR with neural networks
- Based on academic research
- Solving time from hours to seconds

Why haven't we seen adoption in the industry?

Even though search in imperfect-information games was invented in 2016/2017, it took 4-5 years for the first search-based, or depth-limited, solver to be released. Even to this day, tabular solvers are still popular.

There are a few different reasons for this:

- Poker is large family of games
- Cultural reasons - First solvers were invented before search
- Complex players needs
- These techniques relied on abstractions for values estimation

Poker - A family of games

Unlike games like chess or Go, poker, even when only considering No-Limit Hold’em, consists of a large family of games:

- Any number of players from 2 to 10
- Cash games, which different rake structures
- Tournament poker, with different structures (satellites, top-heavy, knockouts, progressive knockouts, etc.) with large strategic implications
- An infinite number of asymmetric stacks variations
- Many custom variants

Issues with blueprint strategies

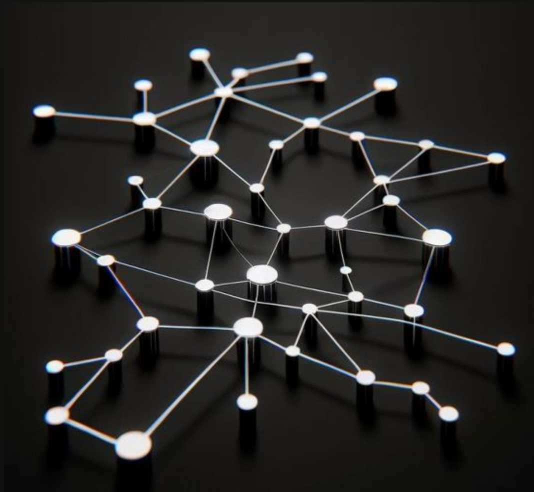
The approach of building a sound blueprint strategy like Libratus/Pluribus makes it hard to satisfy players:

- We’d need to pre-compute thousands of blueprint strategies.
- Even if it was feasible to so, it would be an engineering nightmare to implement, especially for desktop-based applications, since blueprint strategies can be large.

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 diversity of data, it can do a good job of interpolating any PBS 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



GTO Wizard AI

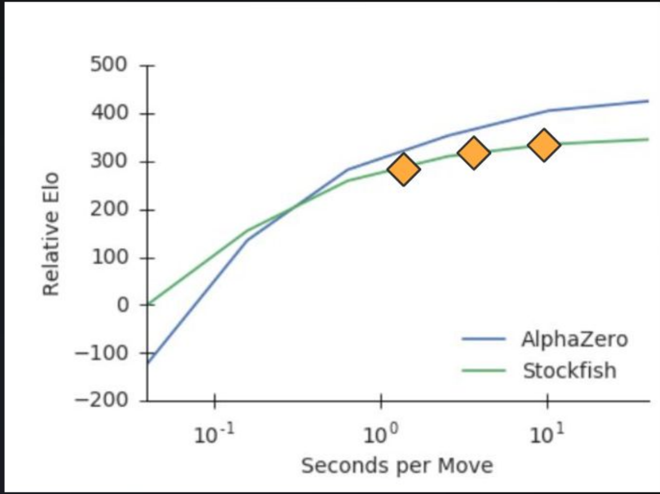
- Based on the latest AI research in imperfect-information games
- Uses reinforcement learning and search, similar to the ReBeL algorithm
- Engineering and algorithmic improvements to achieve state-of-the-art performance, **with minimal use of abstractions** at the depth-limit, and running on CPU in a few seconds
- Can solve any stack size for HUNL, with any rake format, by leveraging the function approximation power of deep neural networks

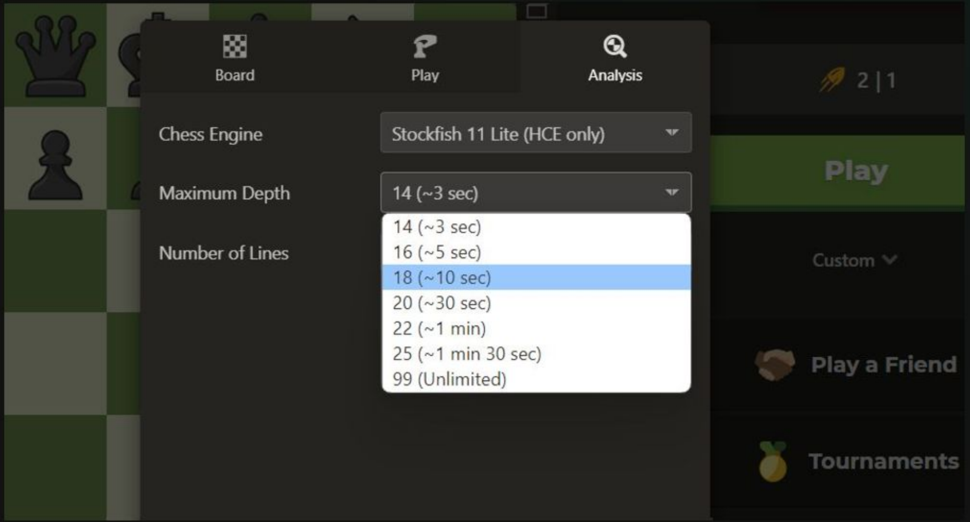
Agent	Slumbot (2018)	Top Humans
DeepStack's reimplementation (2020)	-6.3 ± 4.0	
ReBeL (2020)	4.5 ± 0.5	16.5 ± 6.9
Supremus (2020)	**17.6 ± 4.4**	
GTO Wizard AI (2022)	**19.4 ± 4.4**	

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

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?

Extend our approach to larger games

- Incorporate tournament models in the value network, i.e. predict the mapping from chips → $EV
- Multiplayer games in Hold'em
- 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 in a few seconds

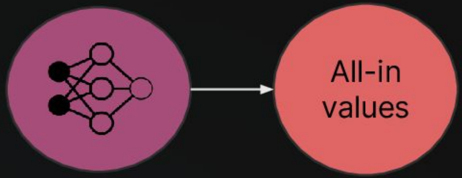
Play against our bot!

Due to the competition constraints (time/memory/compute), the best approach would have been to pre-solve an abstracted game as a blueprint strategy.

However, we wanted to base our bot on the techniques we use in practice and that were outlined in this talk

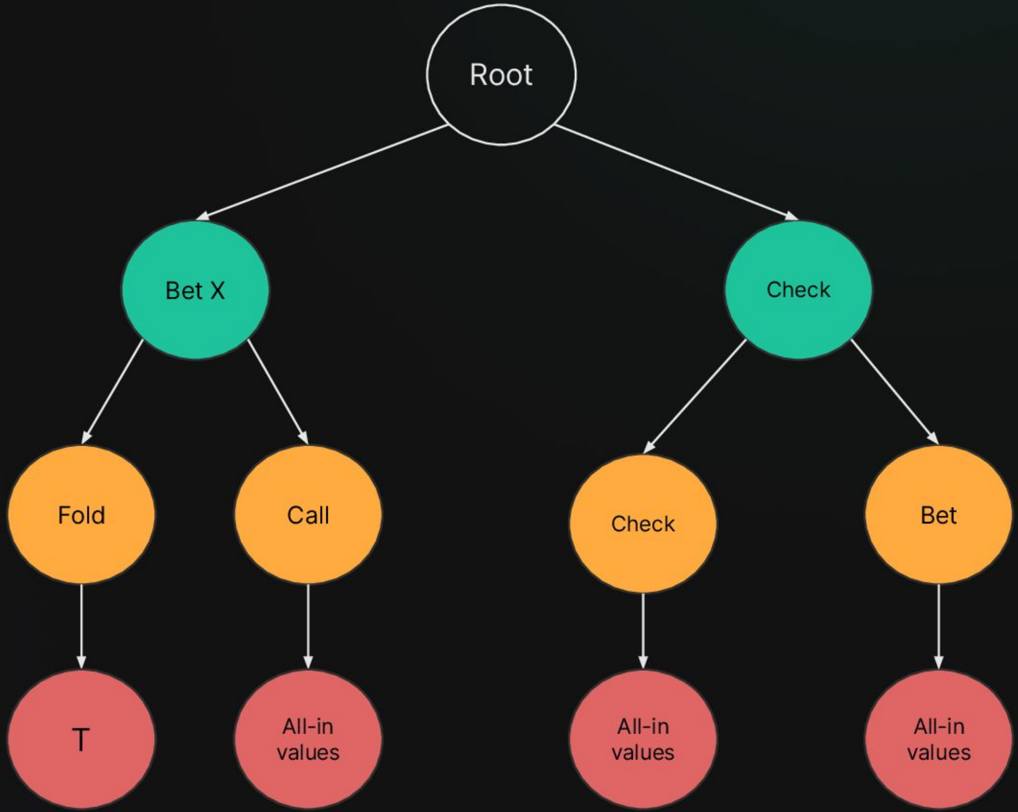
- No information abstraction
- No blueprint strategy
- Solve small depth-limited subgames in real-time, using CFR
- Due to time and compute constraints, instead of using a value network, we assume that players get all-in values at the depth-limit. *Note that this was shown to be a poor heuristic for value estimation*

Play against our bot!



1. Set a small depth-limited subgame based on time constraints (~2 steps lookahead)
2. Replace pseudo-terminal nodes by all-in terminal nodes as an heuristic for value estimation
3. Solve the subgame with CFR

Given more resources, we would have used a learned value network, but everything else would be the same



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, don't hesitate to contact us at **work@gtowizard.com**