Equilibrium Finding for Large Adversarial Imperfect-Information Games

Noam Brown

“And that’s why there’s never going to be a computer that will play World Class Poker. It’s a people game.”
-Doyle Brunson, *Super/System* 1979

“The analysis of a more realistic poker game than our very simple model should be quite an interesting affair.”
-John Forbes Nash, 1951


Imperfect-Information Games
Perfect-Information Games
"The Golden Crescent"
"The Golden Triangle"
from SE Asia
to Asia
to USA
from SA
Major Trafficking Routes
opiates
cocaine
Growing Areas
opium poppy
coca

No-Limit Texas Hold'em Poker

- Long-standing challenge problem in AI and game theory
- Massive in size (two-player has 10^{161} decision points)
- By far the most popular form of poker

2017 Brains vs AI

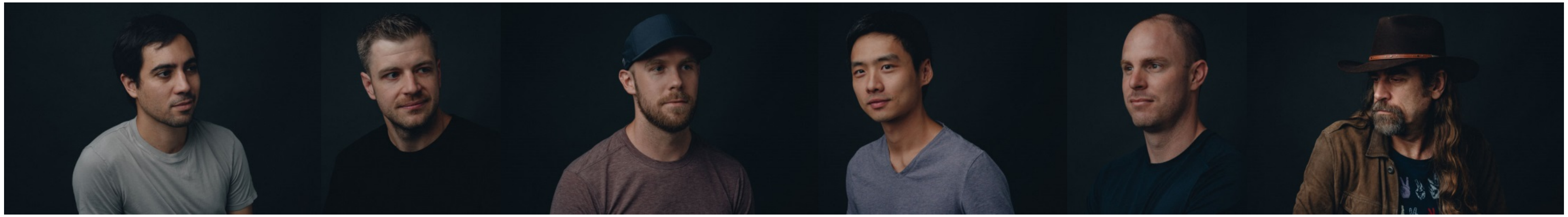
- Libratus (our 2017 AI) against four of the ***best*** heads-up no-limit Texas Hold’em poker pros



- 120,000 hands over 20 days in January 2017
- $200,000 divided among the pros based on performance
- Won with 99.98% statistical significance
- Trained purely from self play; no human data
- Training: 3 million core hours (~$100,000); Running: 1,200 CPU cores

2019 Pluribus Experiment

- Pluribus (our 2019 AI) against 15 top professionals in ***six-player*** no-limit Texas Hold'em



- 10,000 hands over 12 days in June 2019
  – Used variance-reduction techniques to decrease luck
  – One bot playing with five humans
- Won with >95% statistical significance
- Cost under $150 to train, runs on 28 CPU cores (no GPUs)

Talk Outline

- Background
- Improving Counterfactual Regret Minimization (CFR)
  - Discounted CFR
  - Best-Response Pruning
- Scaling Equilibrium Finding to Large Games
  - Deep CFR
- Search in Imperfect-Information Games
  - Multi-Valued States
  - ReBeL: Combining Deep Reinforcement Learning and Search
- Conclusion

Nash Equilibrium

Nash Equilibrium: a set of strategies in which no player can improve by deviating

In two-player zero-sum games, playing a Nash equilibrium ensures you will not lose in expectation

Exploitability: How much we'd lose to a best response

Critical assumption: Our strategy is common knowledge, but the outcomes of random processes are **not** common knowledge

Nash Equilibrium

Nash Equilibrium: a set of strategies in which no player can improve by deviating

In two-player zero-sum games, playing a Nash equilibrium ensures you will not lose in expectation

Exploitability: How much we'd lose to a best response

	Round 1	**Round 2**	**Round 3**
Us			
Best Response			

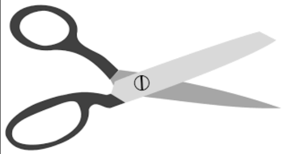
Our Exploitability = 1

Nash Equilibrium

Nash Equilibrium: a set of strategies in which no player can improve by deviating

In two-player zero-sum games, playing a Nash equilibrium ensures you will not lose in expectation

Exploitability: How much we'd lose to a best response

	Round 1	Round 2	Round 3
Us			
Best Response			

Our Exploitability = 1

Nash Equilibrium

Nash Equilibrium: a set of strategies in which no player can improve by deviating

In two-player zero-sum games, playing a Nash equilibrium ensures you will not lose in expectation

Exploitability: How much we'd lose to a best response

	Round 1	Round 2	Round 3
Us			
Best Response			

Our Exploitability = 0

Nash Equilibrium

"Poker is simple, as your opponents make mistakes, you profit."

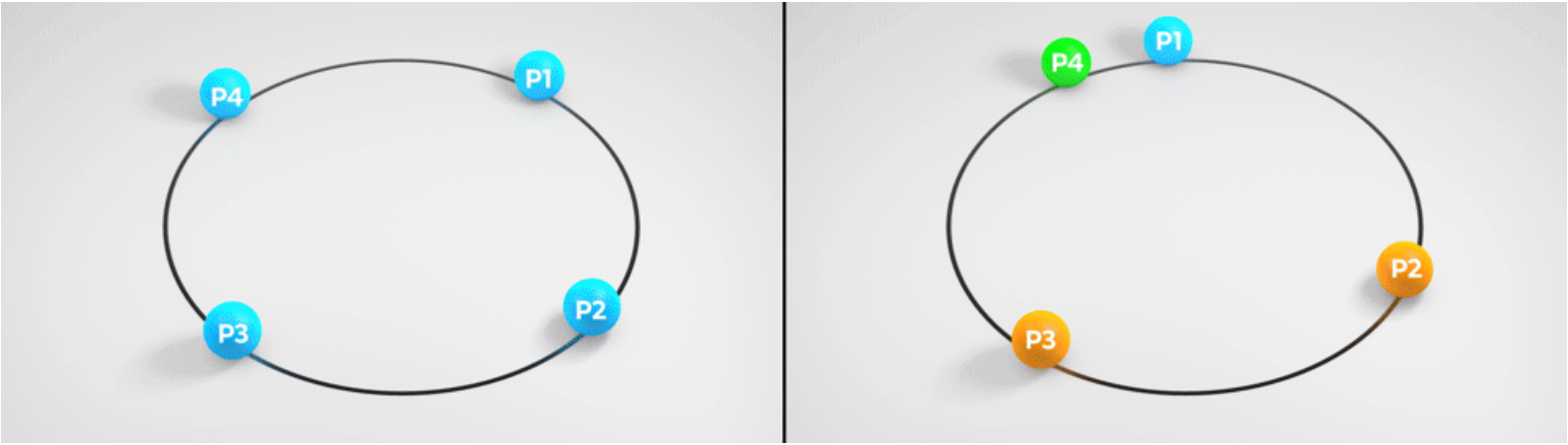
-Ryan Fee's Poker Strategy Guide

	Round 1	Round 2	Round 3
Us			
Best Response			

Our Exploitability = 0

Nash Equilibria in Non-Two-Player Zero-Sum Games

- Cannot be computed in polynomial time
- Even if it could be computed efficiently, might not make sense to play
- But same algorithms ***still work well in practice*** in six-player poker!

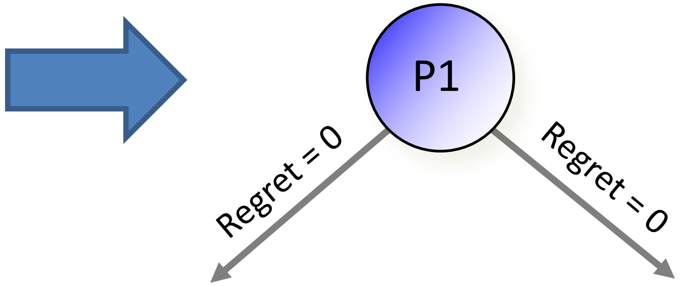


Improvements to
Counterfactual Regret Minimization

Monte Carlo Counterfactual Regret Minimization (MCCFR)

[Zinkevich *et al.* NeurIPS-07, Lanctot *et al.* NeurIPS-09]

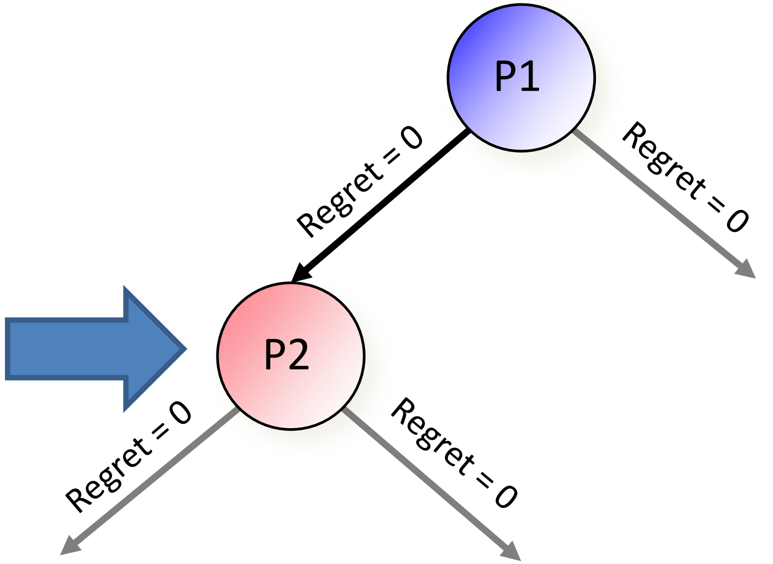
Pick action proportional to **positive regret**



Monte Carlo Counterfactual Regret Minimization (MCCFR)

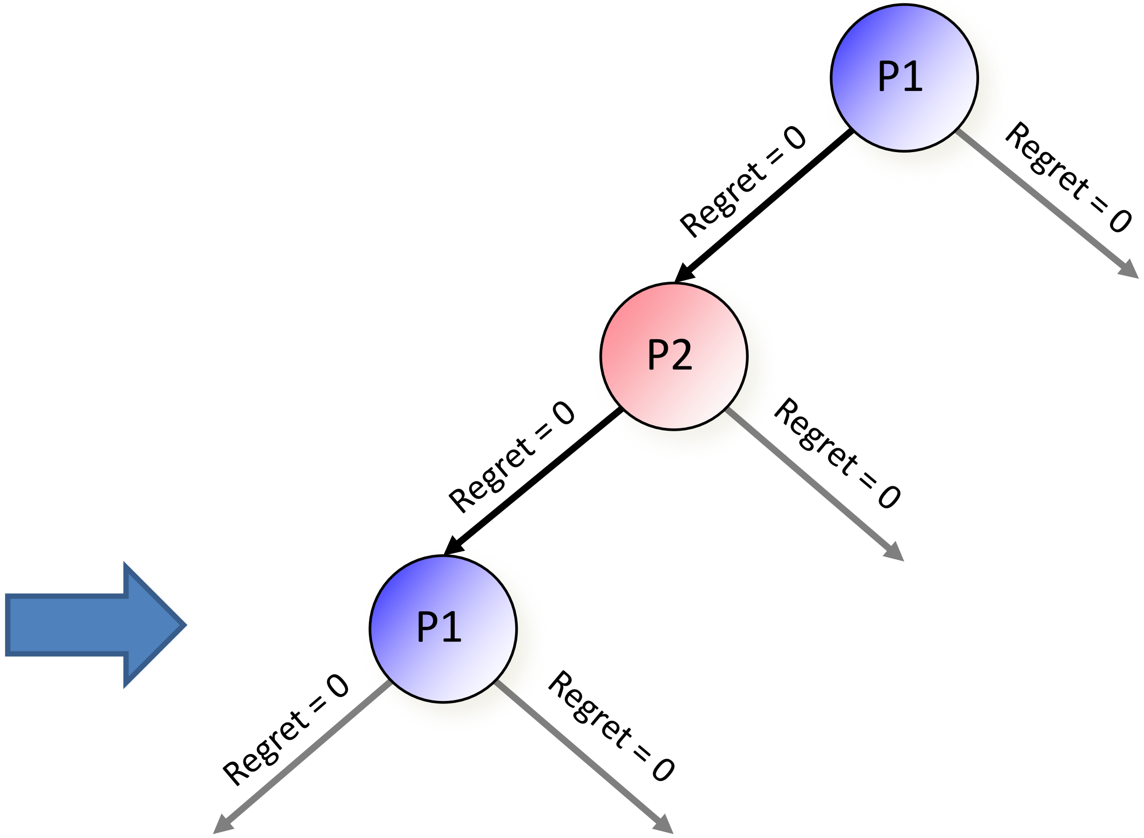
[Zinkevich *et al.* NeurIPS-07, Lanctot *et al.* NeurIPS-09]

Pick action proportional to **positive regret**



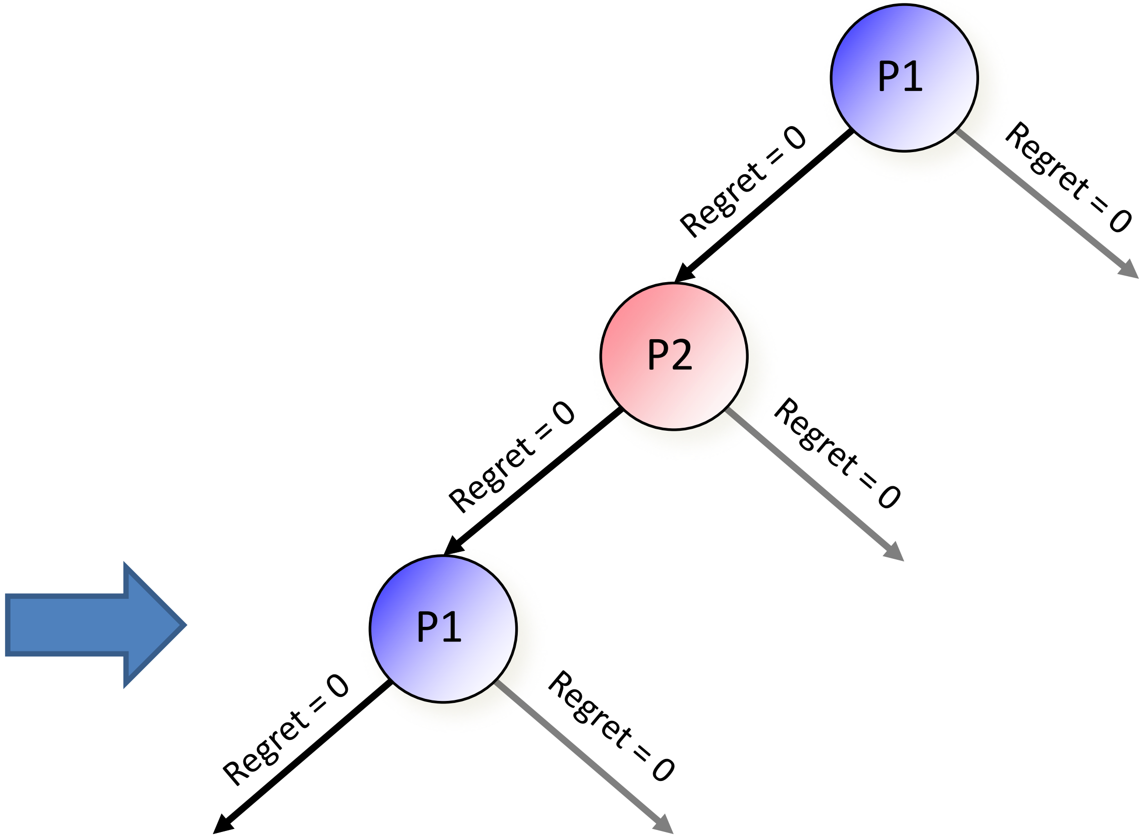
Monte Carlo Counterfactual Regret Minimization (MCCFR)

[Zinkevich *et al.* NeurIPS-07, Lanctot *et al.* NeurIPS-09]



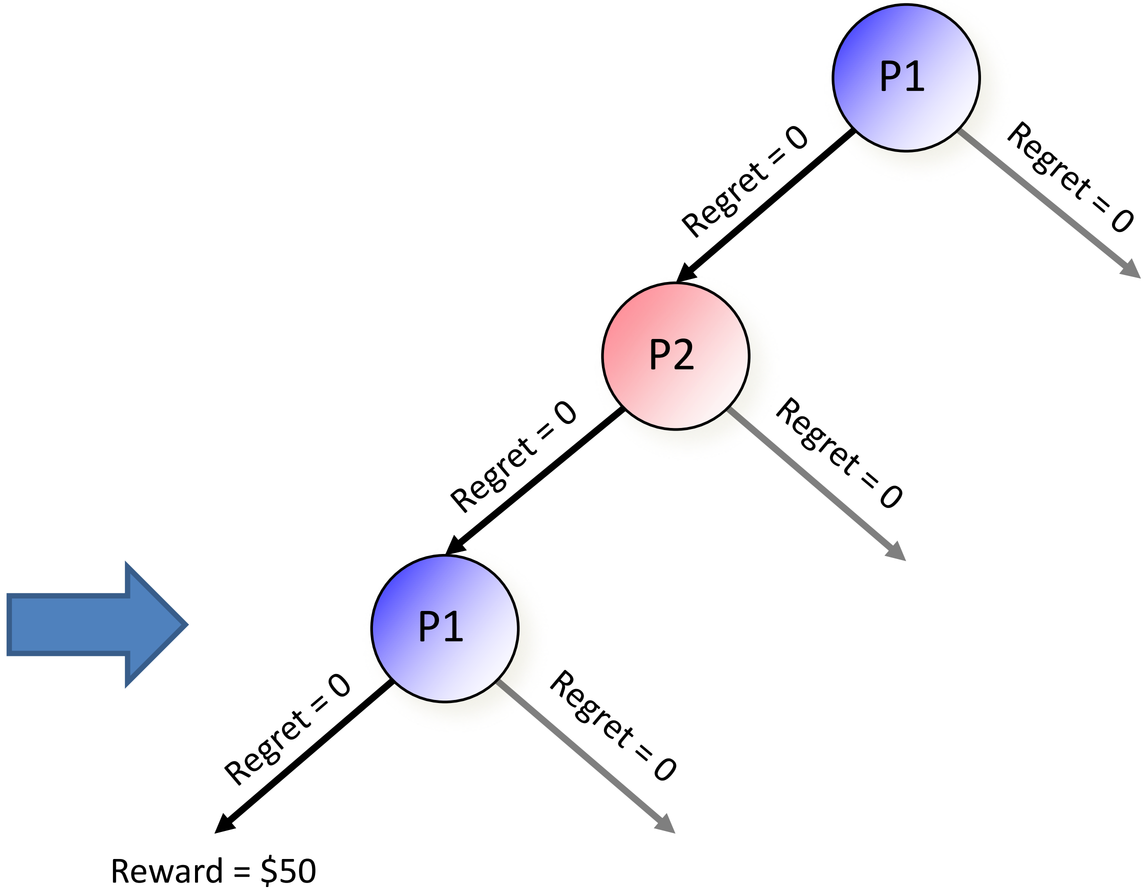
Monte Carlo Counterfactual Regret Minimization (MCCFR)

[Zinkevich *et al.* NeurIPS-07, Lanctot *et al.* NeurIPS-09]



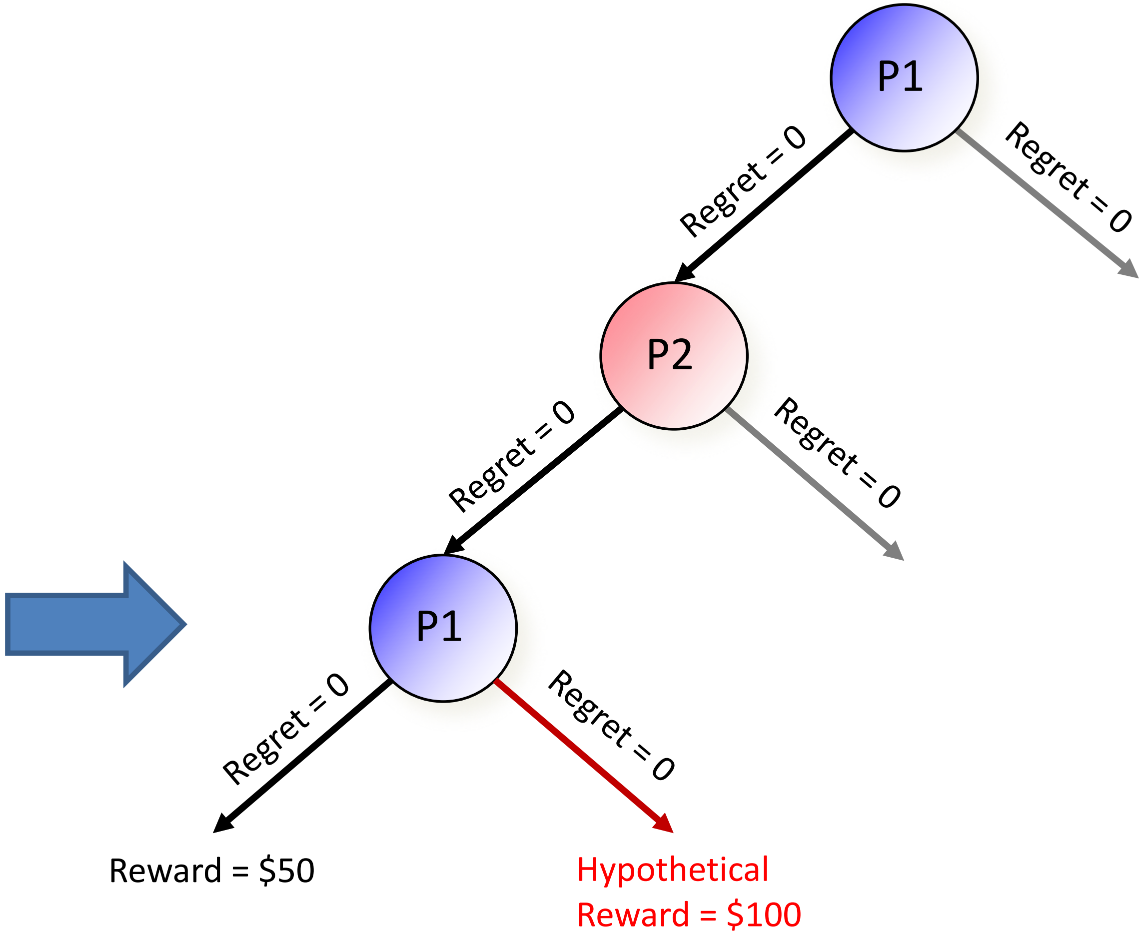
Monte Carlo Counterfactual Regret Minimization (MCCFR)

[Zinkevich *et al.* NeurIPS-07, Lanctot *et al.* NeurIPS-09]



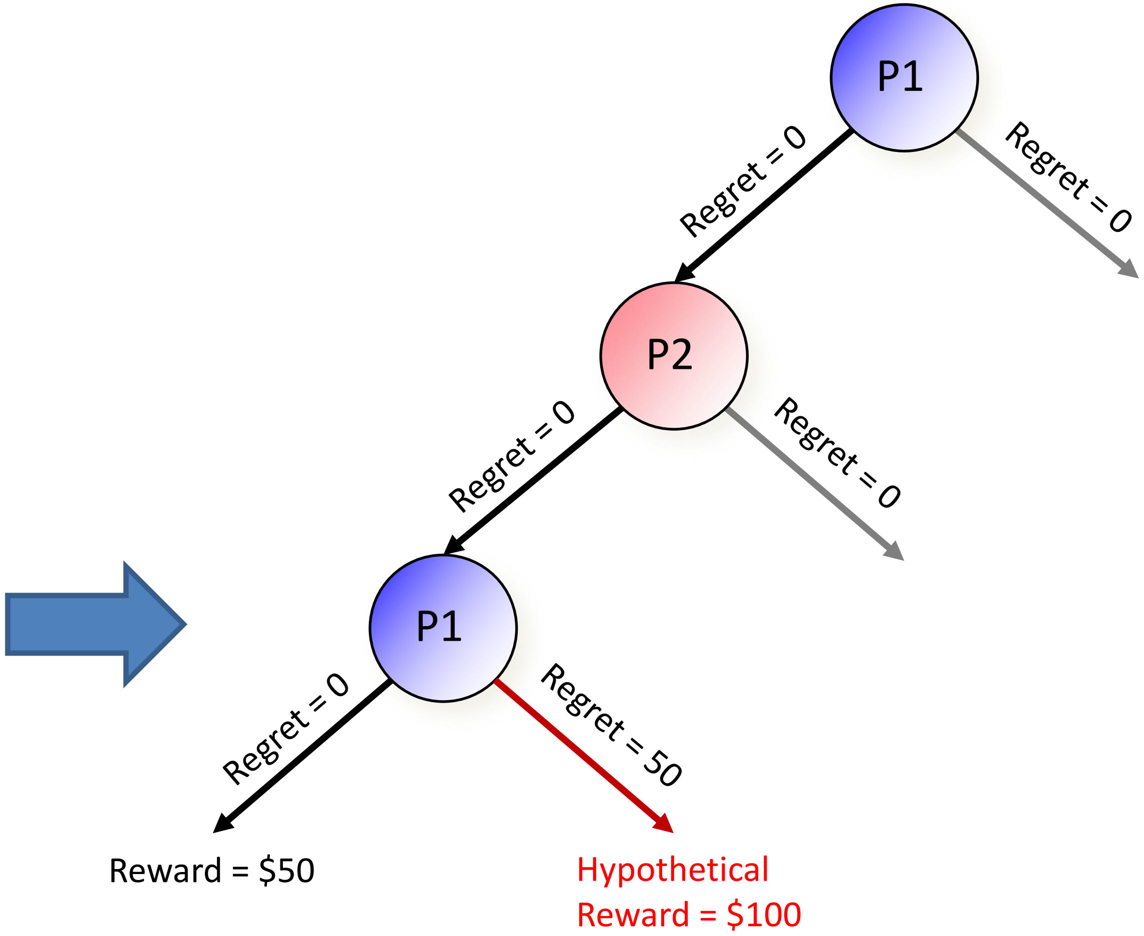
Monte Carlo Counterfactual Regret Minimization (MCCFR)

[Zinkevich *et al.* NeurIPS-07, Lanctot *et al.* NeurIPS-09]



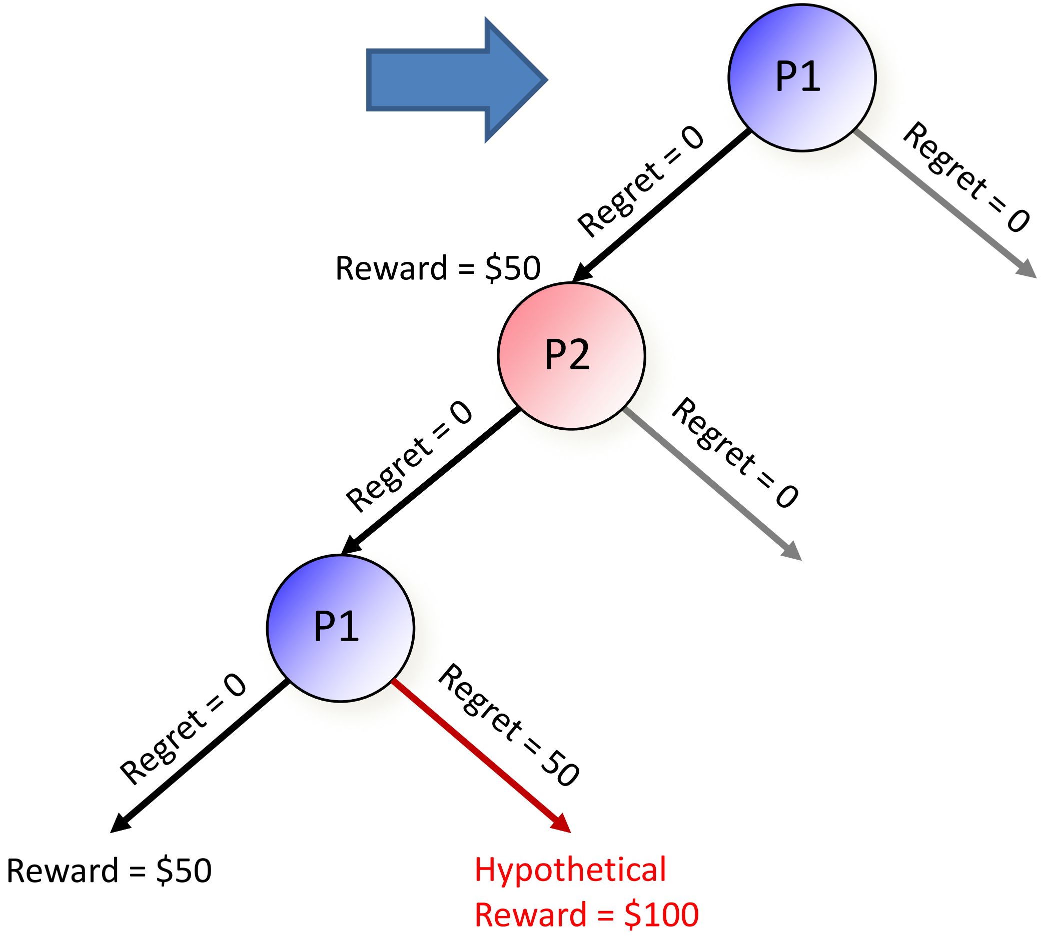
Monte Carlo Counterfactual Regret Minimization (MCCFR)

[Zinkevich *et al.* NeurIPS-07, Lanctot *et al.* NeurIPS-09]



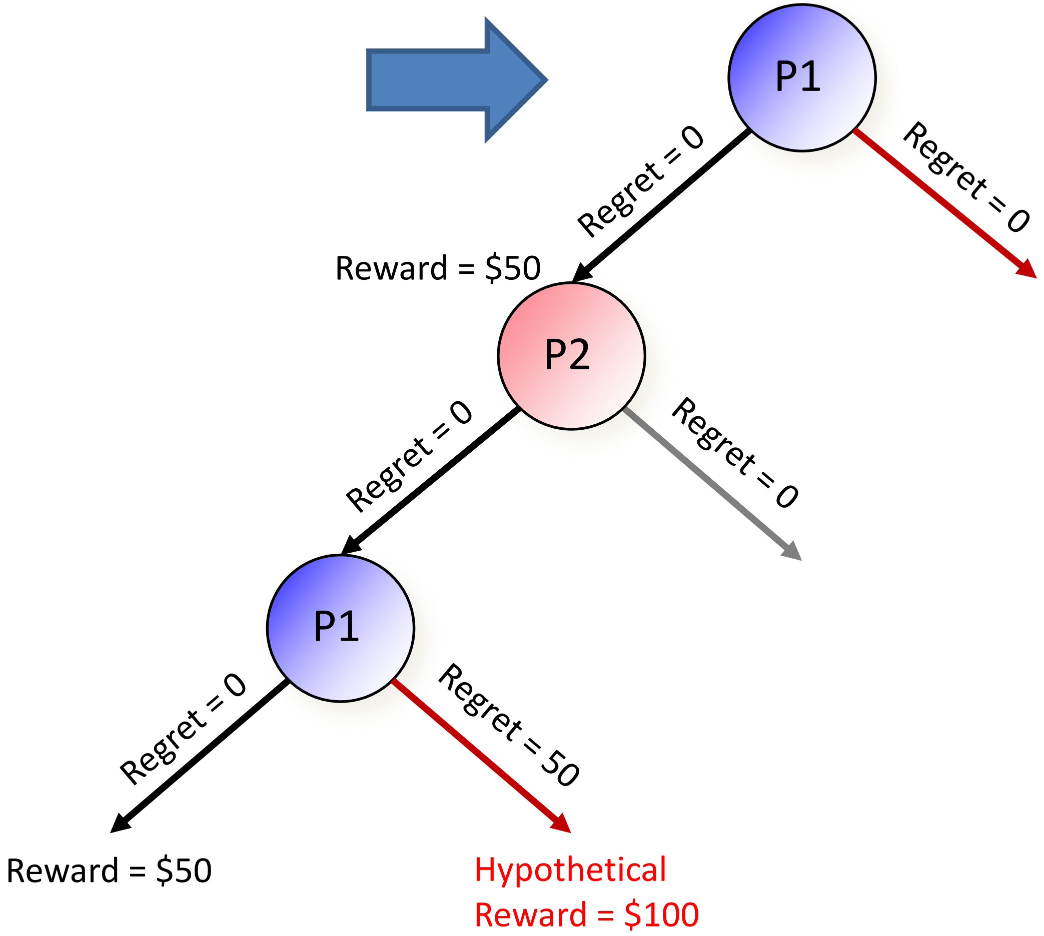
Monte Carlo Counterfactual Regret Minimization (MCCFR)

[Zinkevich *et al.* NeurIPS-07, Lanctot *et al.* NeurIPS-09]



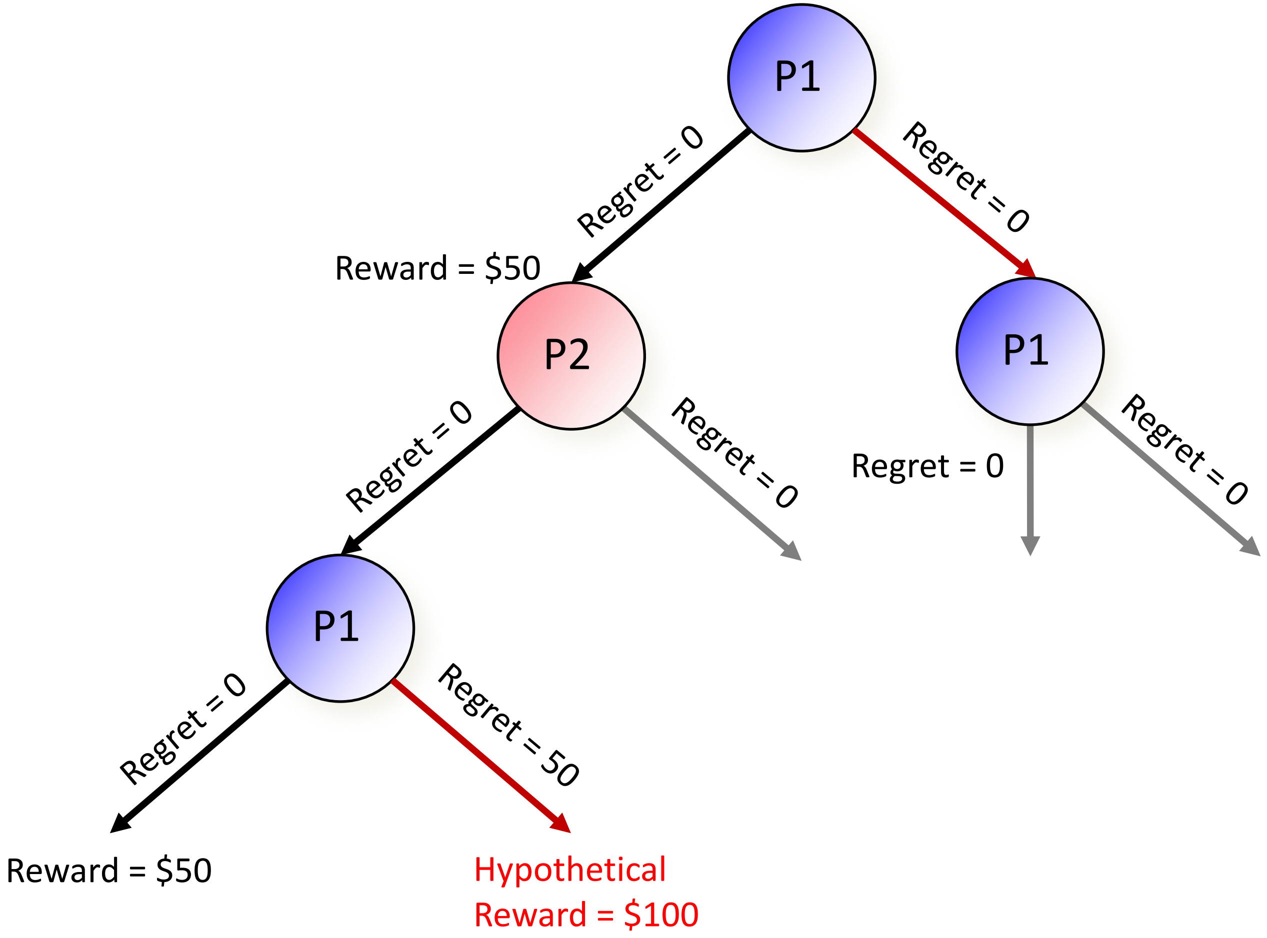
Monte Carlo Counterfactual Regret Minimization (MCCFR)

[Zinkevich *et al.* NeurIPS-07, Lanctot *et al.* NeurIPS-09]



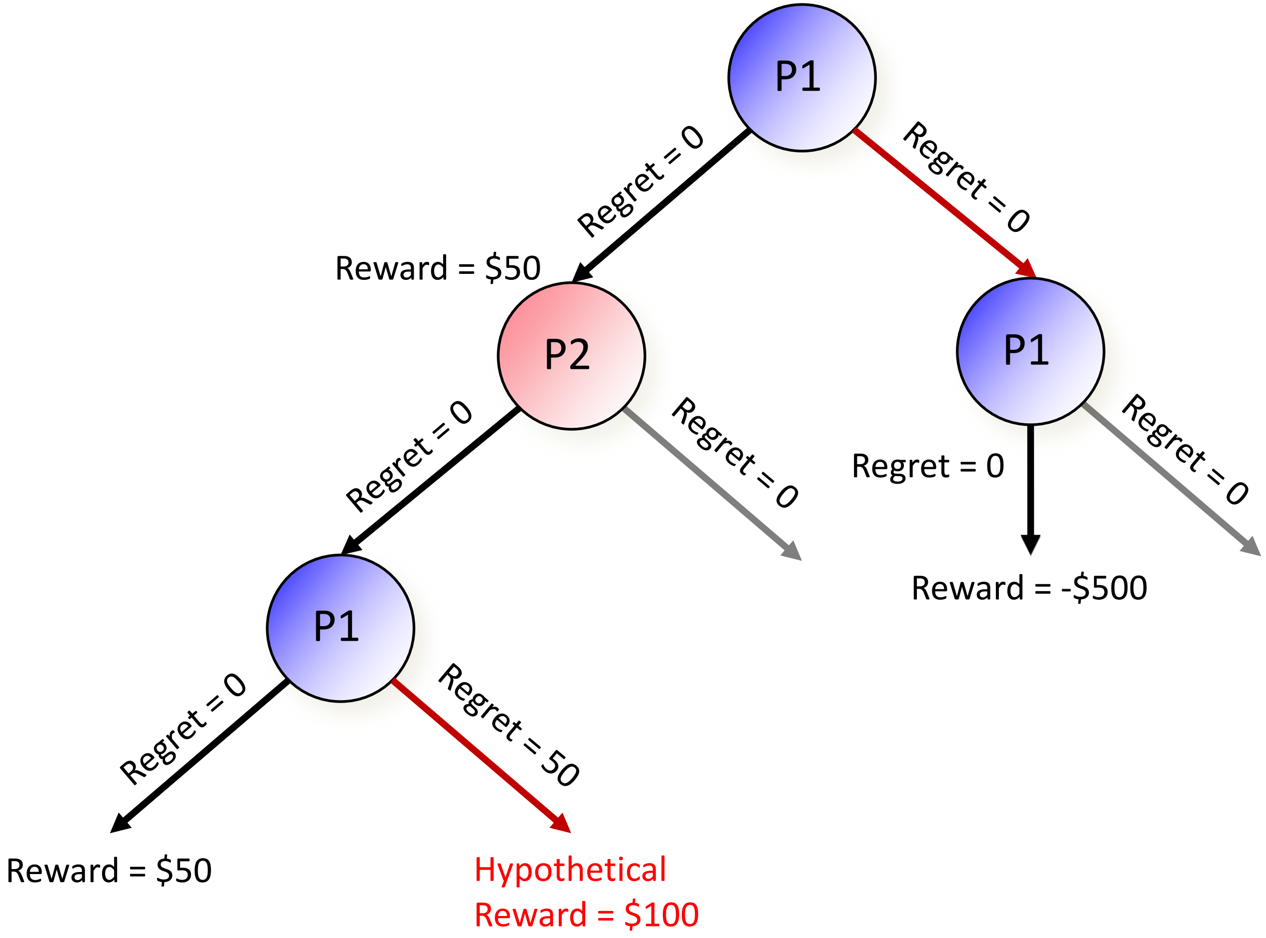
Monte Carlo Counterfactual Regret Minimization (MCCFR)

[Zinkevich *et al.* NeurIPS-07, Lanctot *et al.* NeurIPS-09]



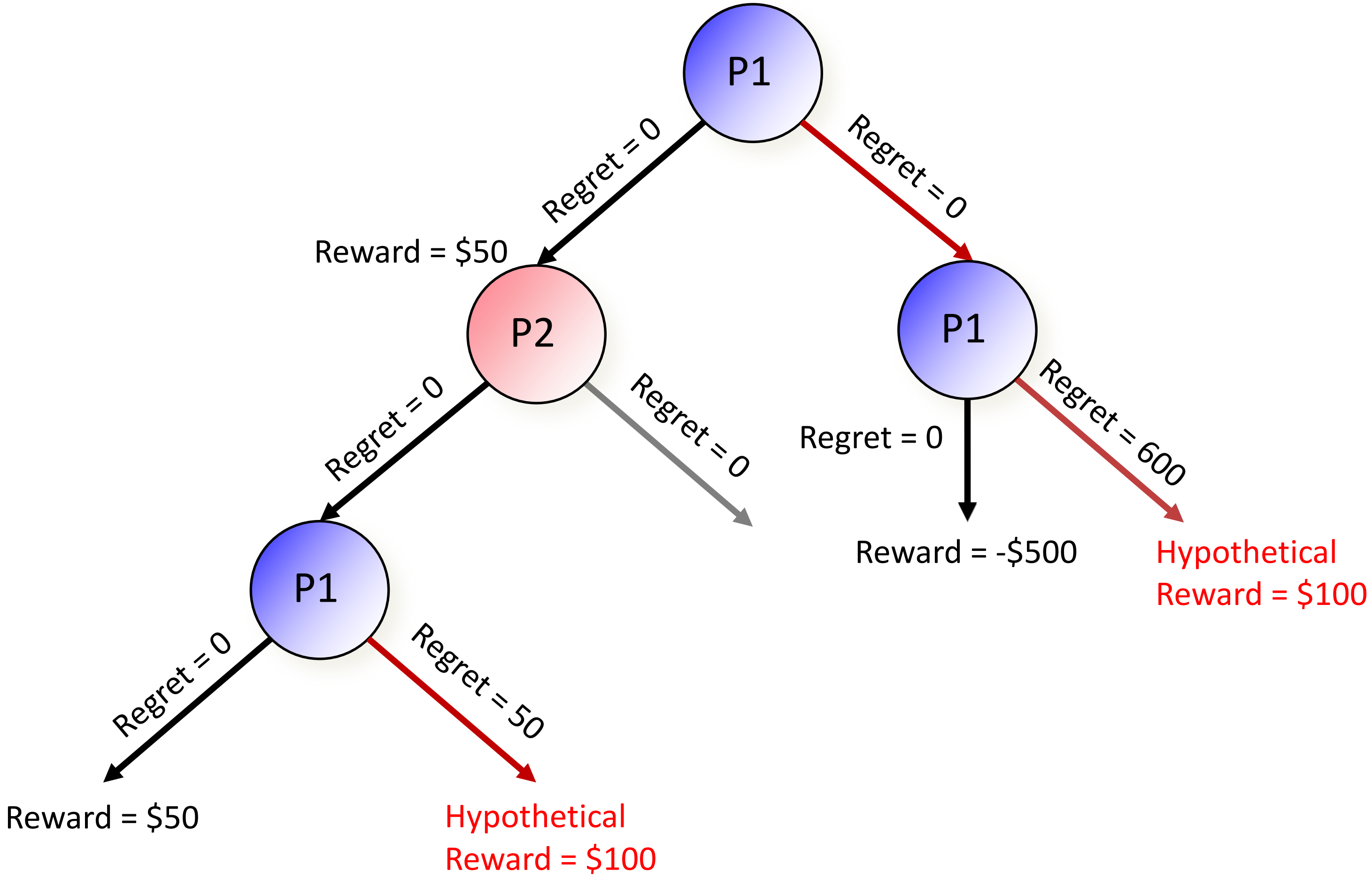
Monte Carlo Counterfactual Regret Minimization (MCCFR)

[Zinkevich *et al.* NeurIPS-07, Lanctot *et al.* NeurIPS-09]



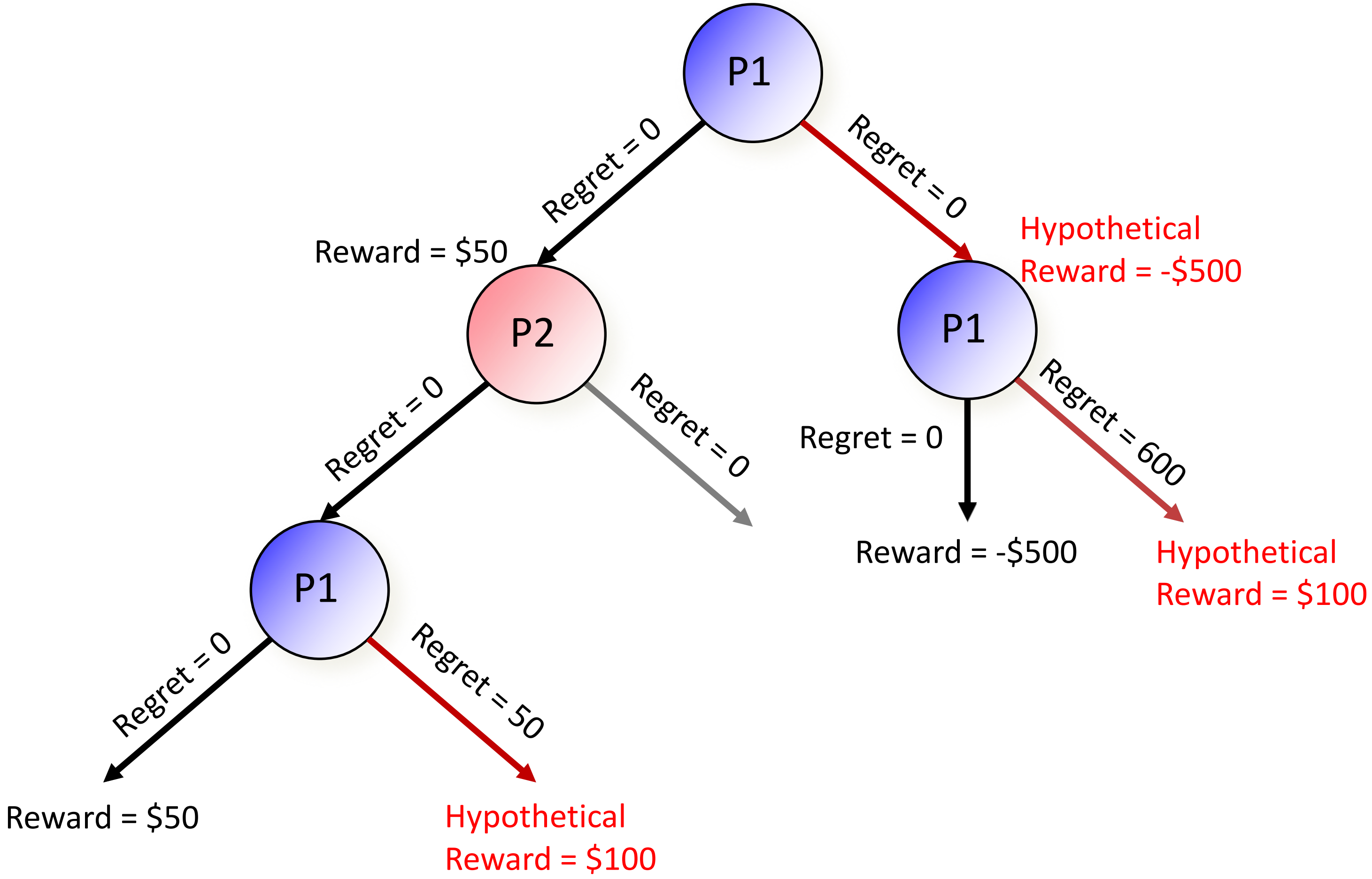
Monte Carlo Counterfactual Regret Minimization (MCCFR)

[Zinkevich *et al.* NeurIPS-07, Lanctot *et al.* NeurIPS-09]



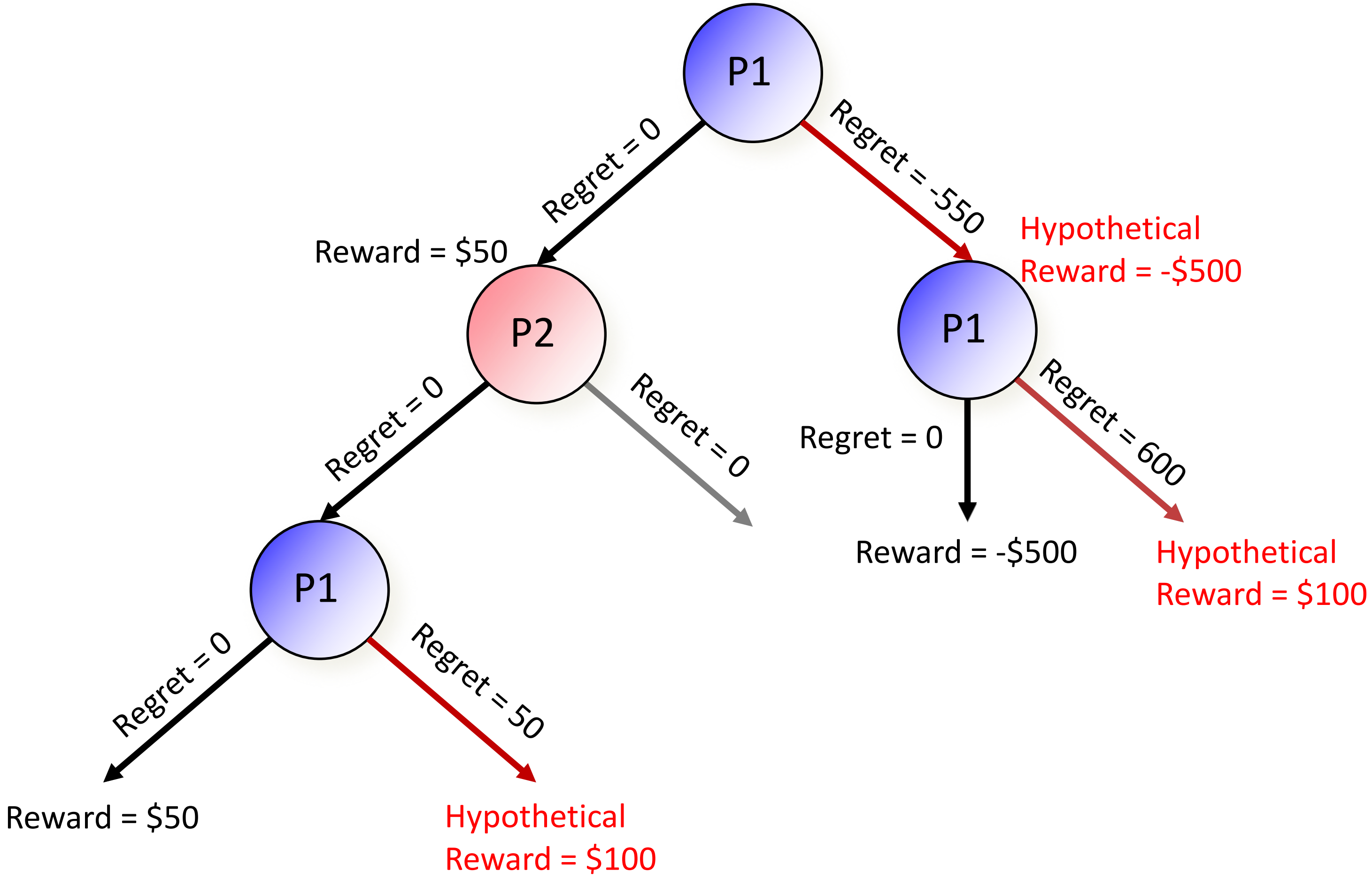
Monte Carlo Counterfactual Regret Minimization (MCCFR)

[Zinkevich *et al.* NeurIPS-07, Lanctot *et al.* NeurIPS-09]



Monte Carlo Counterfactual Regret Minimization (MCCFR)

[Zinkevich *et al.* NeurIPS-07, Lanctot *et al.* NeurIPS-09]



Counterfactual Regret Minimization (CFR)

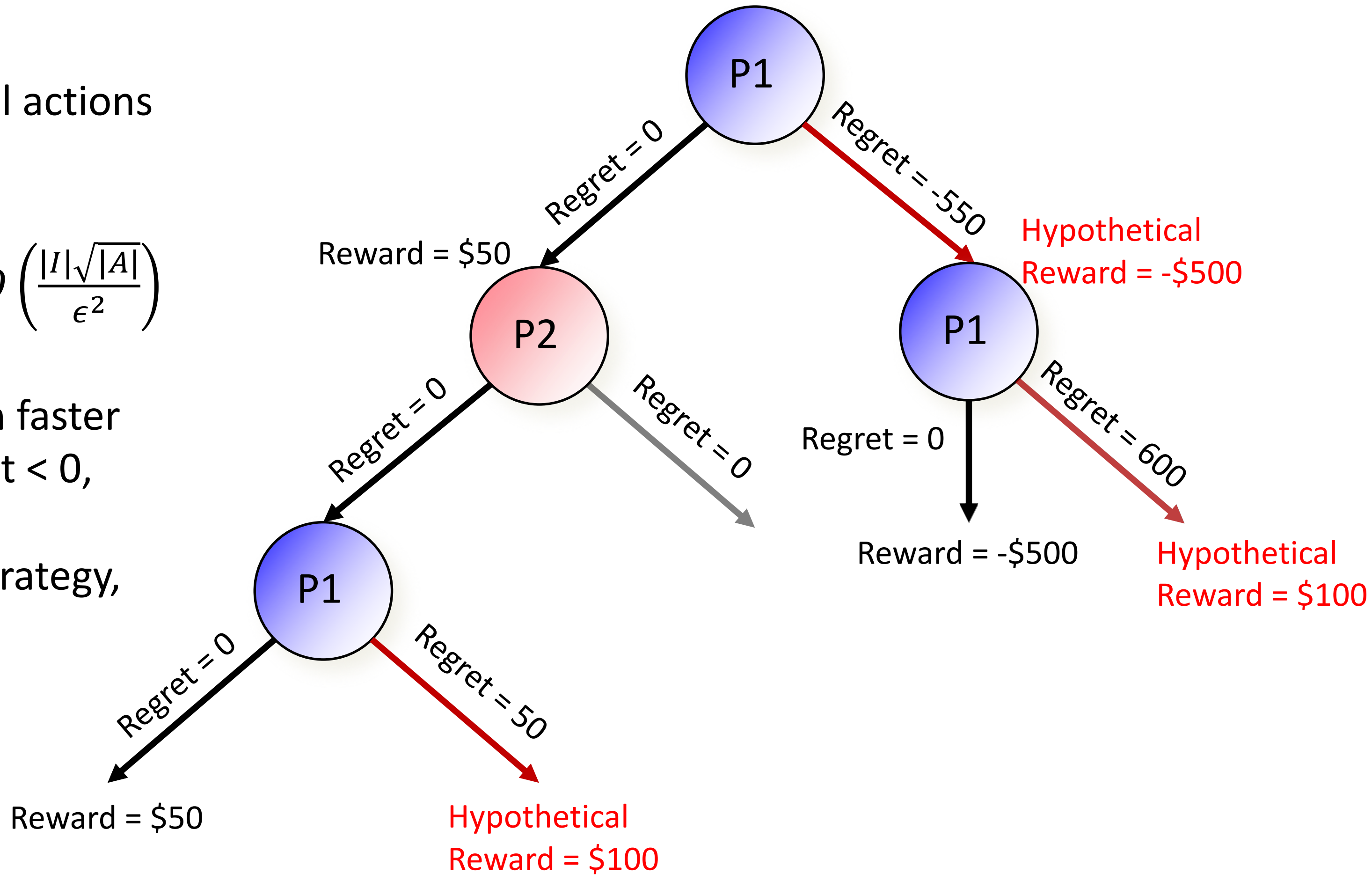
[Zinkevich *et al.* NeurIPS-07]

Similar, but takes the EV over all actions rather than sampling

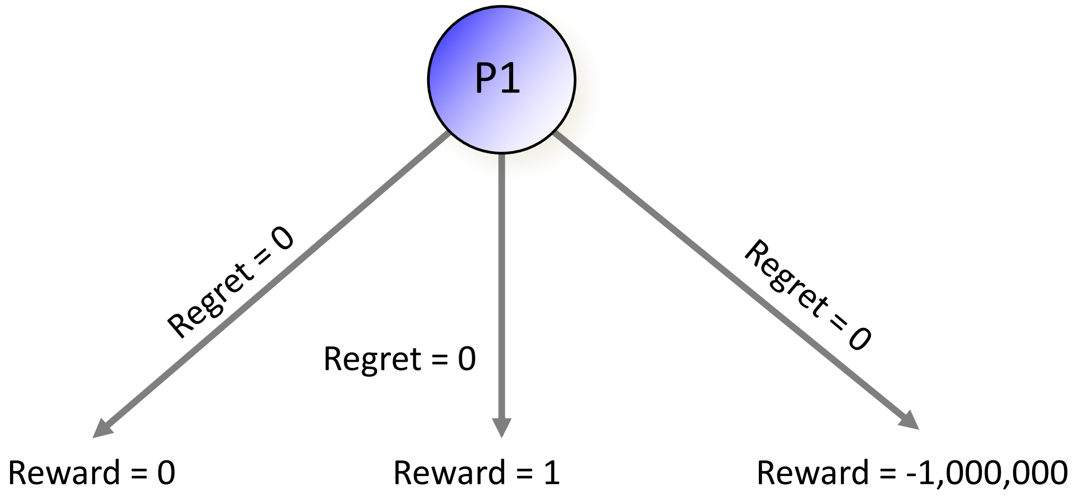
Average converges to Nash in $O\left(\frac{|I|\sqrt{|A|}}{\epsilon^2}\right)$

CFR+: small change that's much faster

- After each iteration, if Regret < 0, set Regret = 0
- When computing **average** strategy, weigh iteration t by t

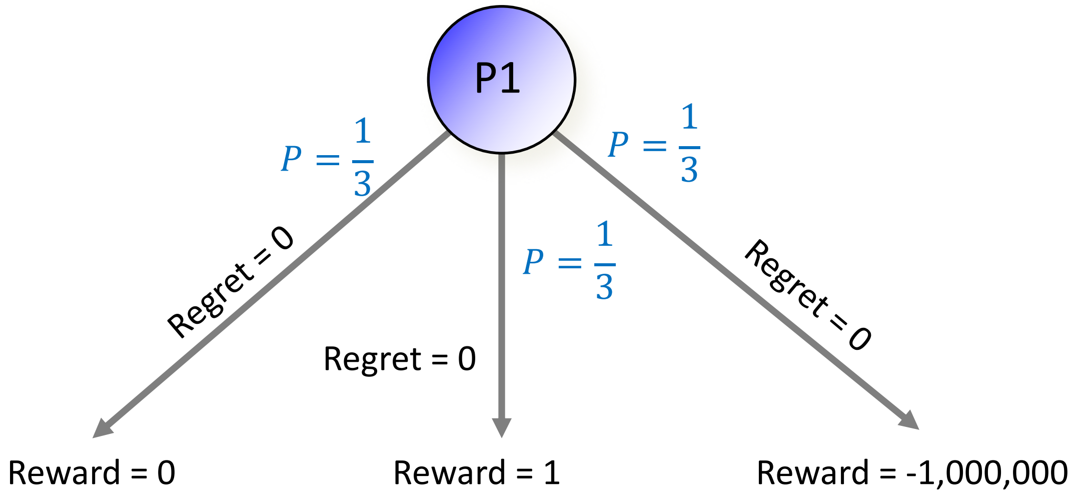


Motivation: limitations of CFR+



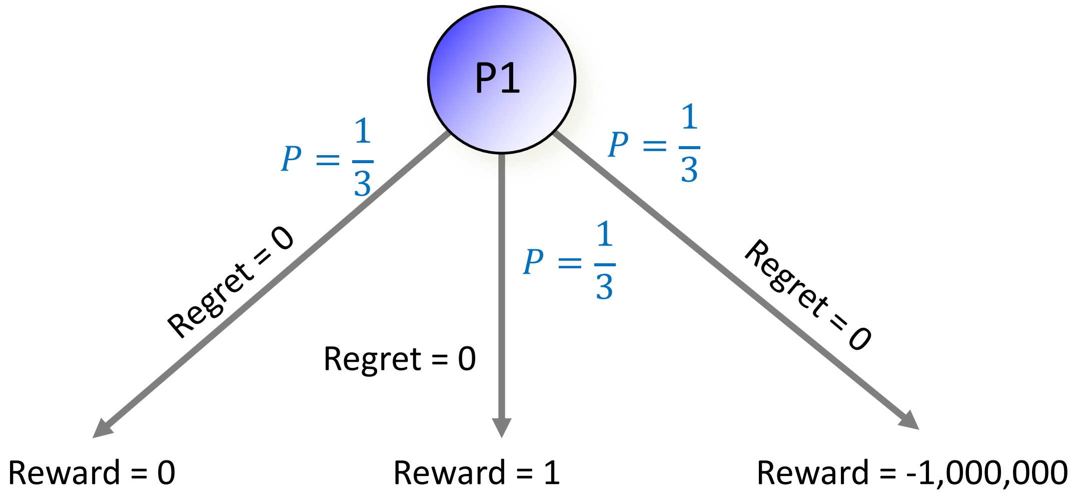
Motivation: limitations of CFR+

- On first iteration, pick all actions with equal probability



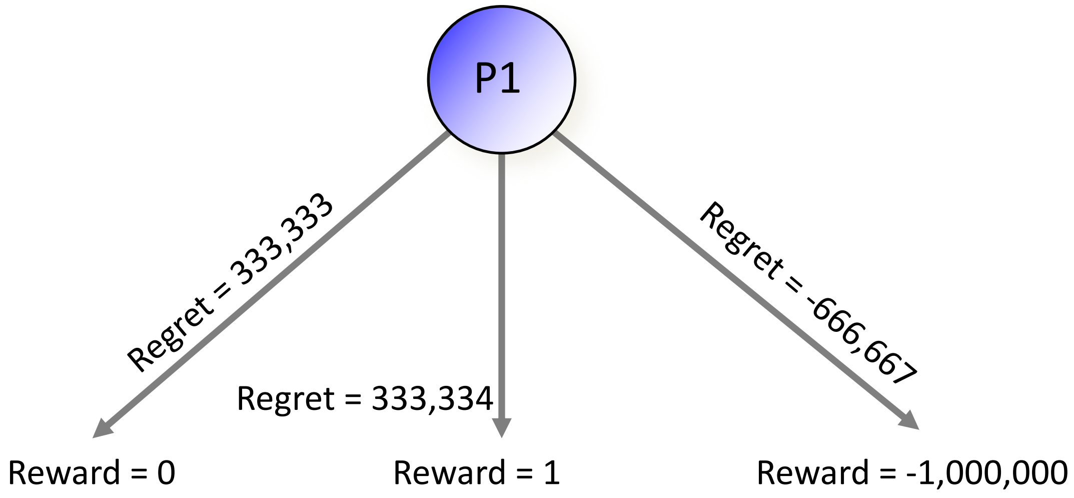
Motivation: limitations of CFR+

- On first iteration, pick all actions with equal probability
- Expected reward is -333,333



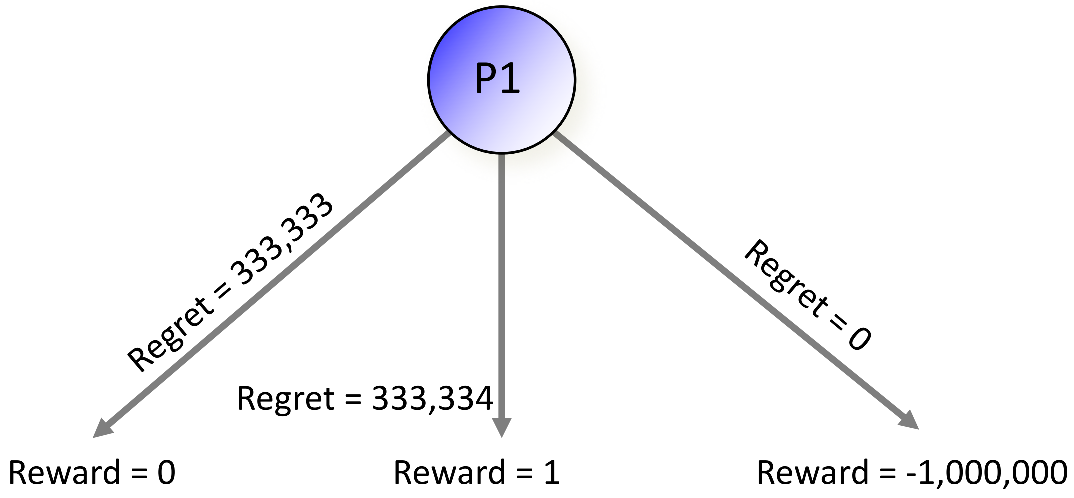
Motivation: limitations of CFR+

- On first iteration, pick all actions with equal probability
- Expected reward is -333,333
- Update regret as Action EV – Achieved EV



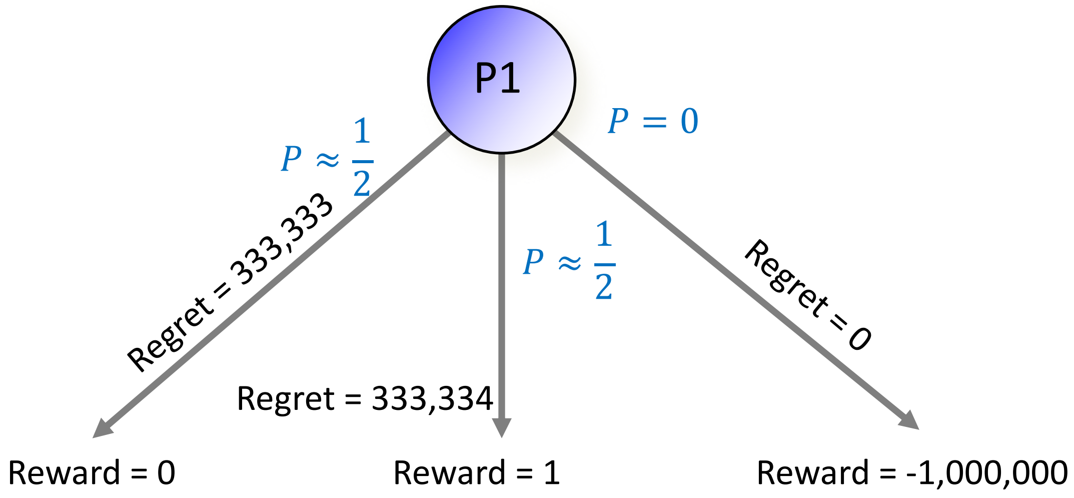
Motivation: limitations of CFR+

- On first iteration, pick all actions with equal probability
- Expected reward is -333,333
- Update regret as Action EV – Achieved EV
- CFR+ floors regret at zero



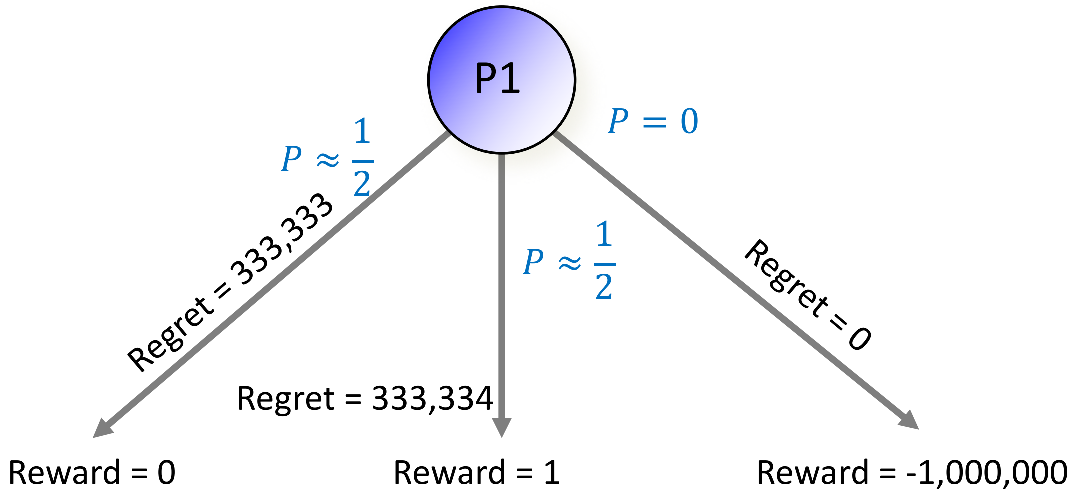
Motivation: limitations of CFR+

- On second iteration, pick actions **proportional to their regret**



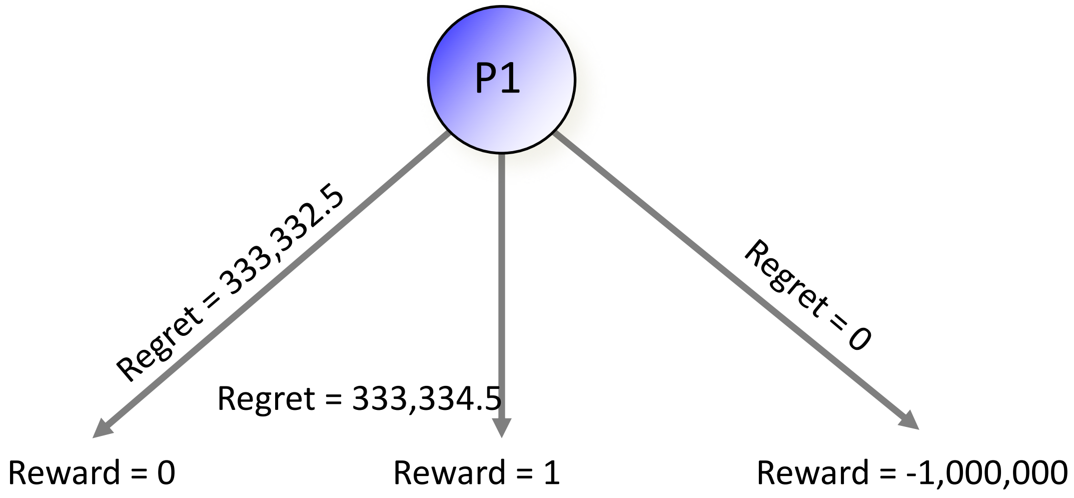
Motivation: limitations of CFR+

- On second iteration, pick actions **proportional to their regret**
- Expected reward ≈ 0.5



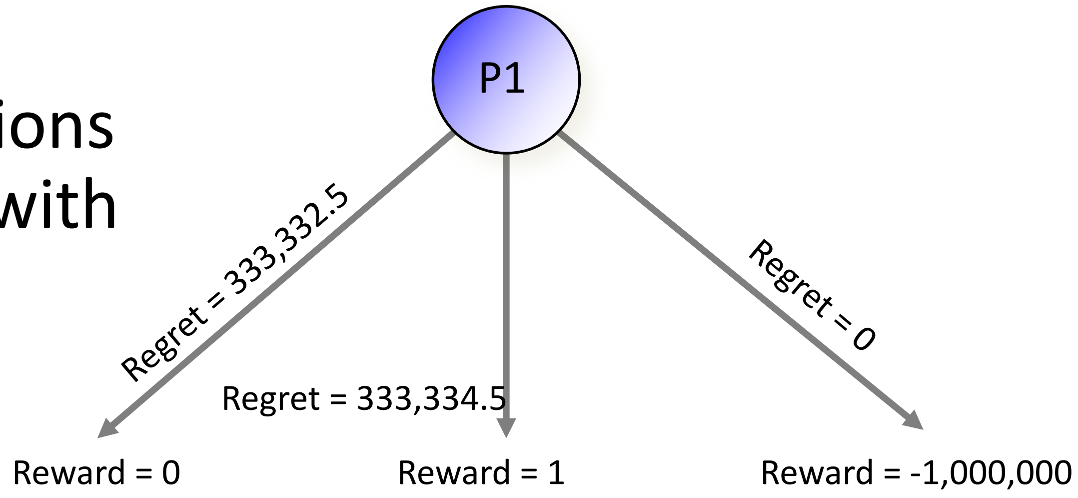
Motivation: limitations of CFR+

- On second iteration, pick actions **proportional to their regret**
- Expected reward ≈ 0.5
- Update regret



Motivation: limitations of CFR+

- Problem: It will take **471,407** iterations for CFR+ to pick the middle action with 100% probability!



- Solution: **Discount** early "bad" iters by weighing iteration t by t
  - Called **Linear CFR**
  - After t iters, first iter only counts for $\frac{2}{t^2+t}$
  - Picks middle action in only **970** iterations
  - Convergence bound increases only by a factor of $\frac{2}{\sqrt{3}}$

Discounted CFR

- Linear CFR: Weigh iteration t by t
- CFR+: Floor regrets at zero
- Can we combine both into Linear CFR+?

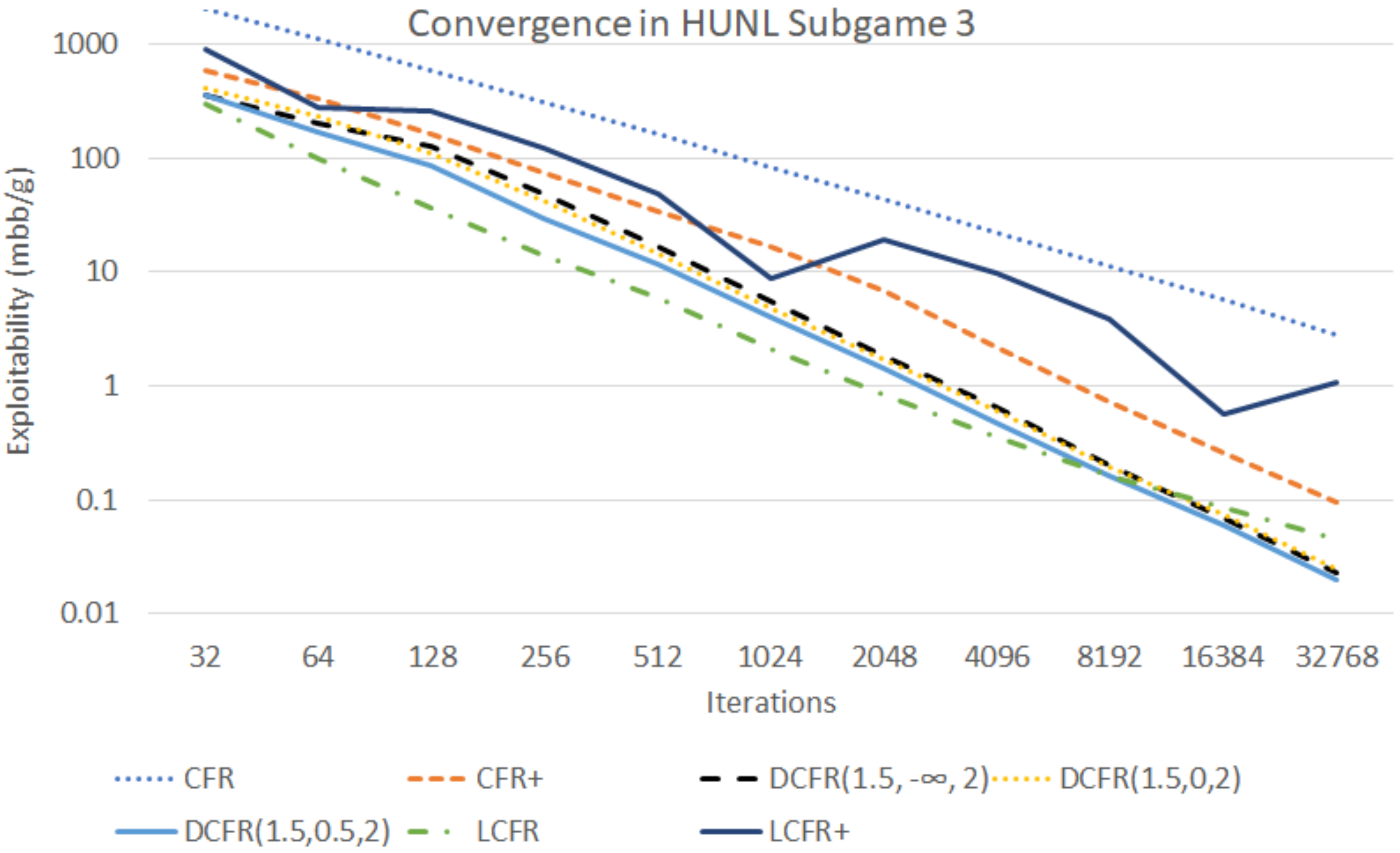
Discounted CFR

- Linear CFR: Weigh iteration t by t
- CFR+: Floor regrets at zero
- Can we combine both into Linear CFR+?
  - Theory: Yes!   Practice: **No!** Does very poorly in practice

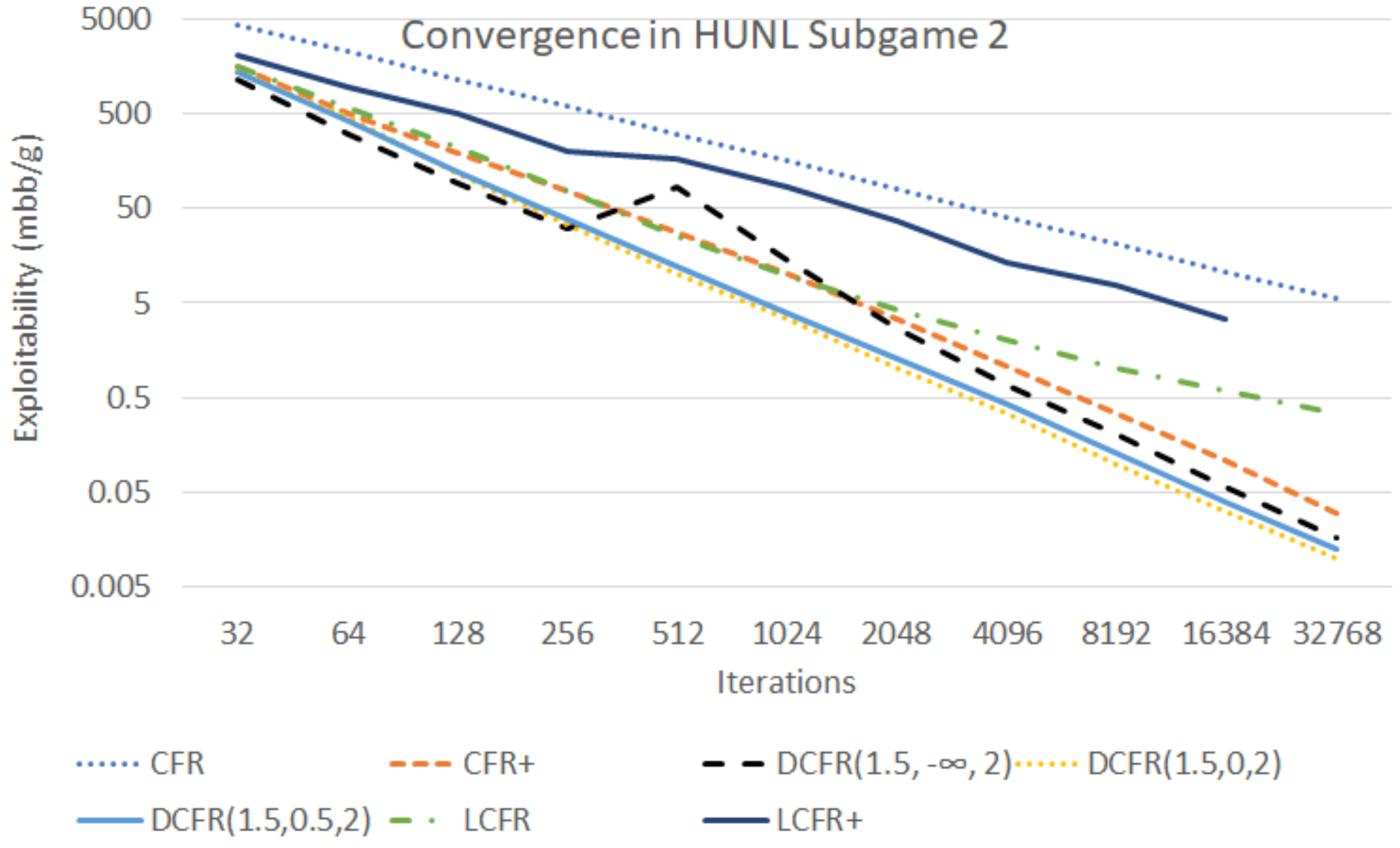
Discounted CFR

- Linear CFR: Weigh iteration t by t
- CFR+: Floor regrets at zero
- Can we combine both into Linear CFR+?
  - Theory: Yes!   Practice: **No!** Does very poorly in practice
- **But** less-aggressive combinations do well: Discounted CFR (DCFR)
  - On each iteration, multiply positive regrets by $\frac{t^\alpha}{t^\alpha+1}$
  - On each iteration, multiply negative regrets by $\frac{t^\beta}{t^\beta+1}$
  - $\alpha = 1.5$, $\beta = 0$ consistently outperforms CFR+

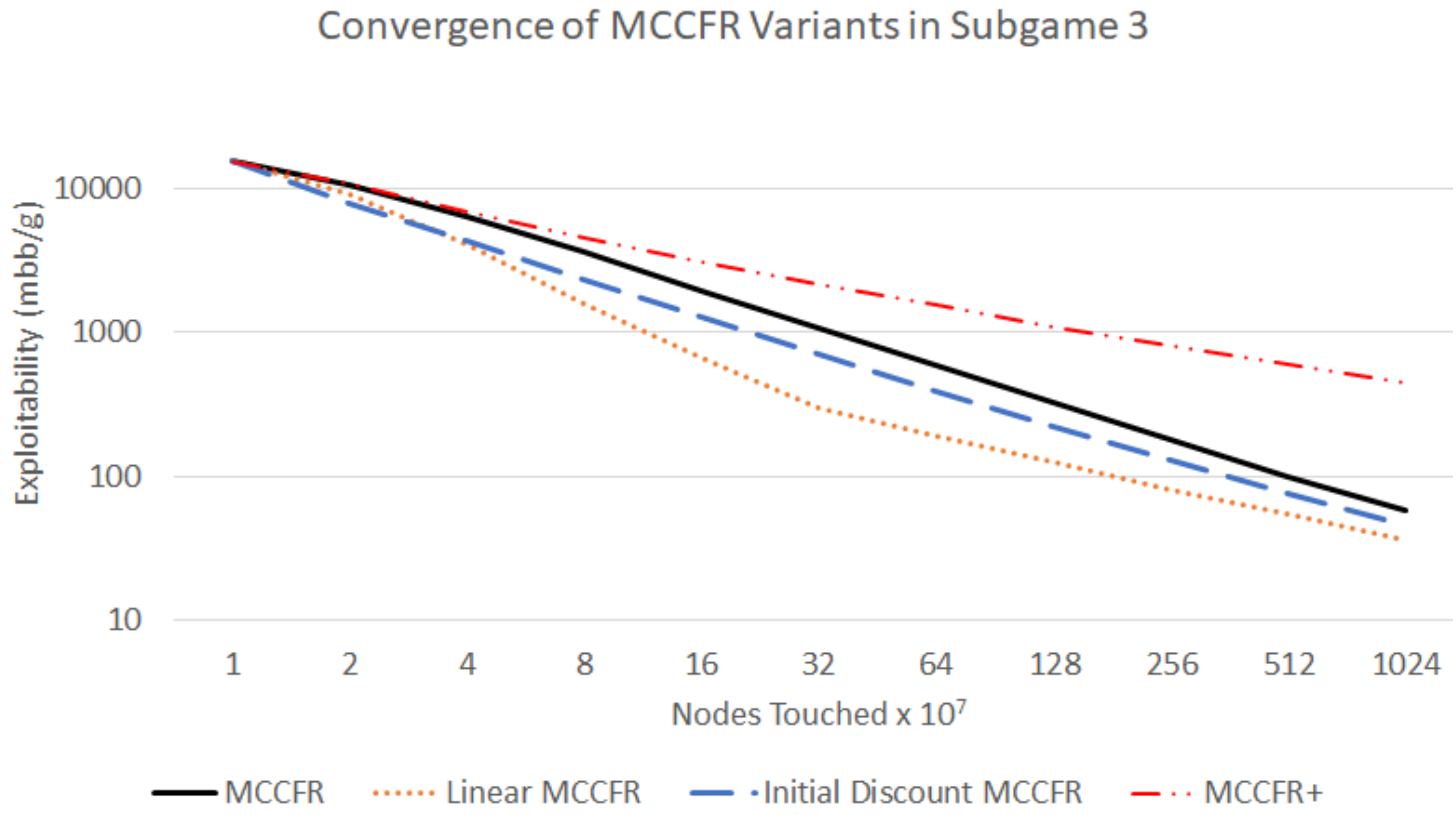
Experimental results on heads-up no-limit Texas hold'em poker endgames used by *Libratus*



Experimental results on heads-up no-limit Texas hold’em poker endgames used by *Libratus*



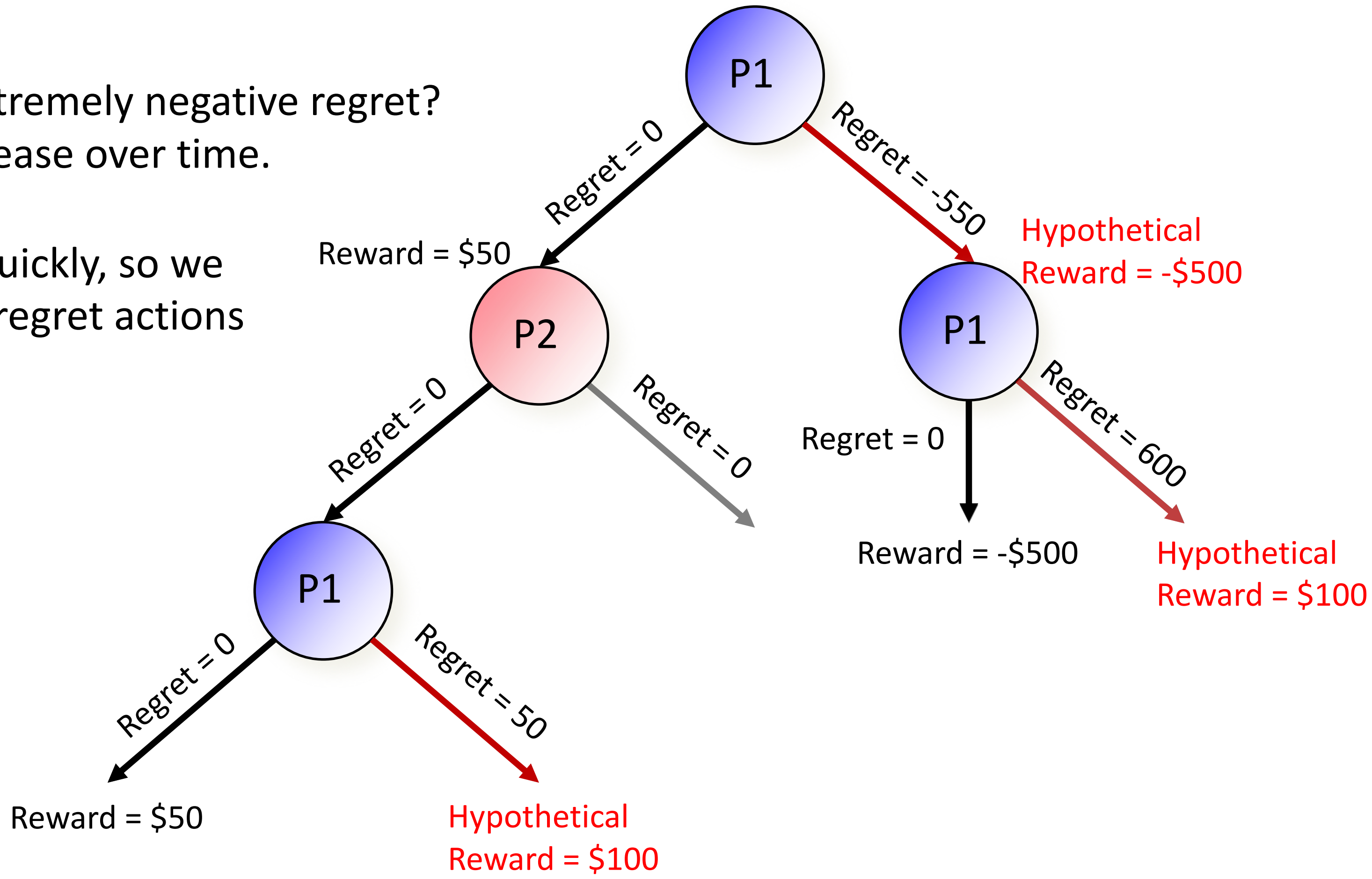
Linear Monte Carlo CFR



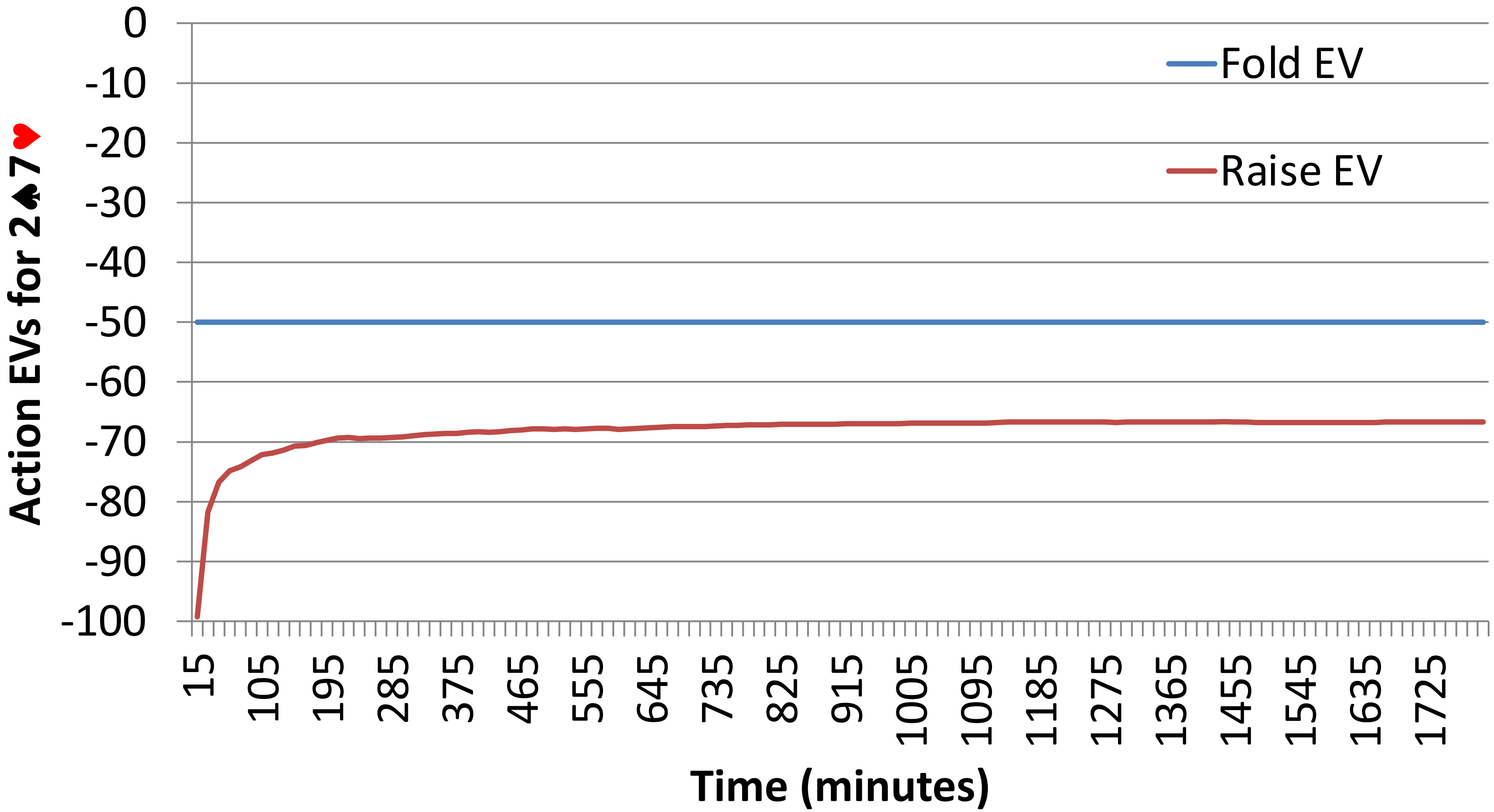
Pruning in CFR

Q: Can we prune actions with extremely negative regret?
A: No, because regret might increase over time.

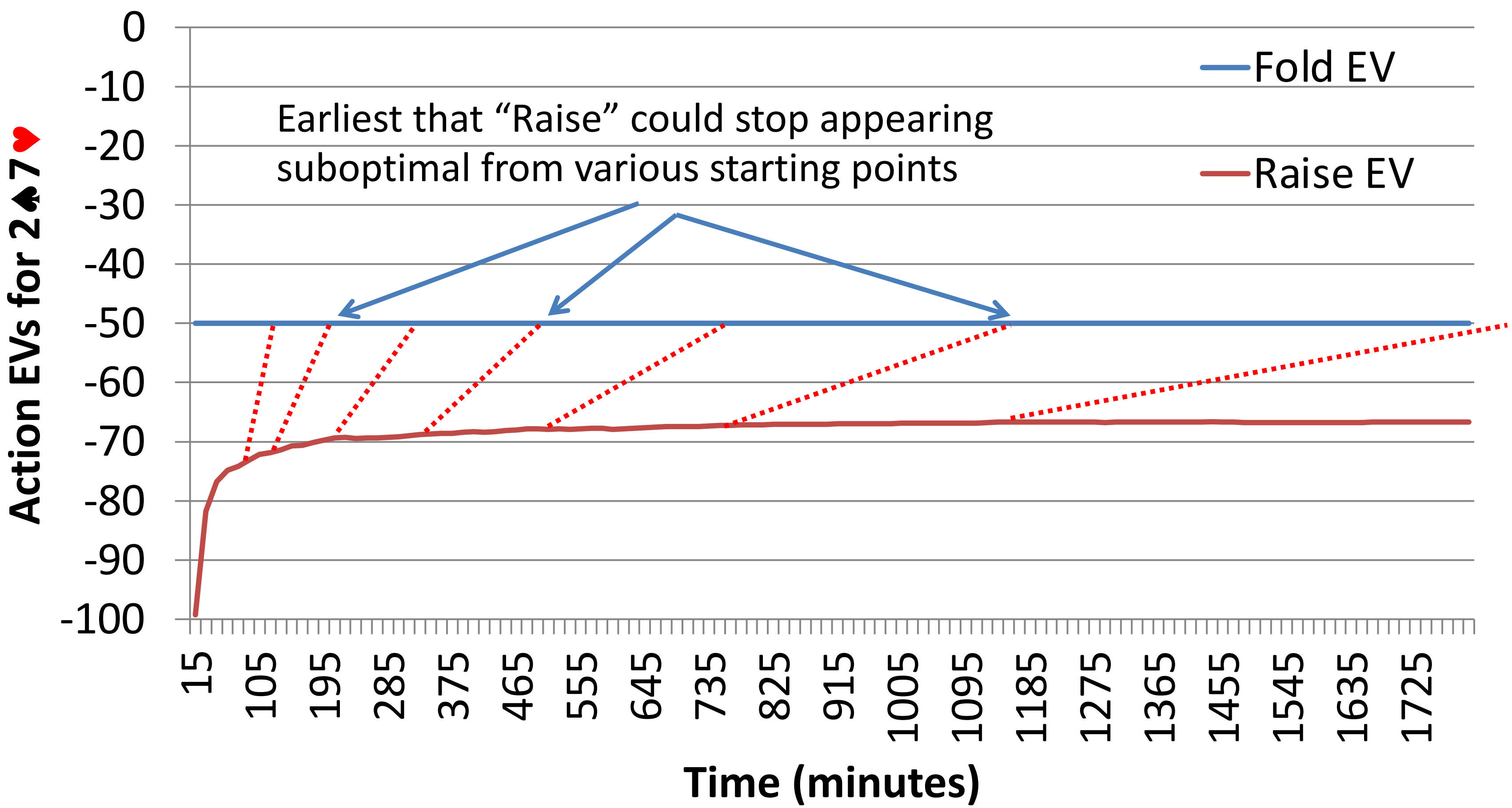
But regret can only increase so quickly, so we can **temporarily** prune negative-regret actions



First Action EV in poker for 2♠7♥



First Action EV in poker for 2♠7♥

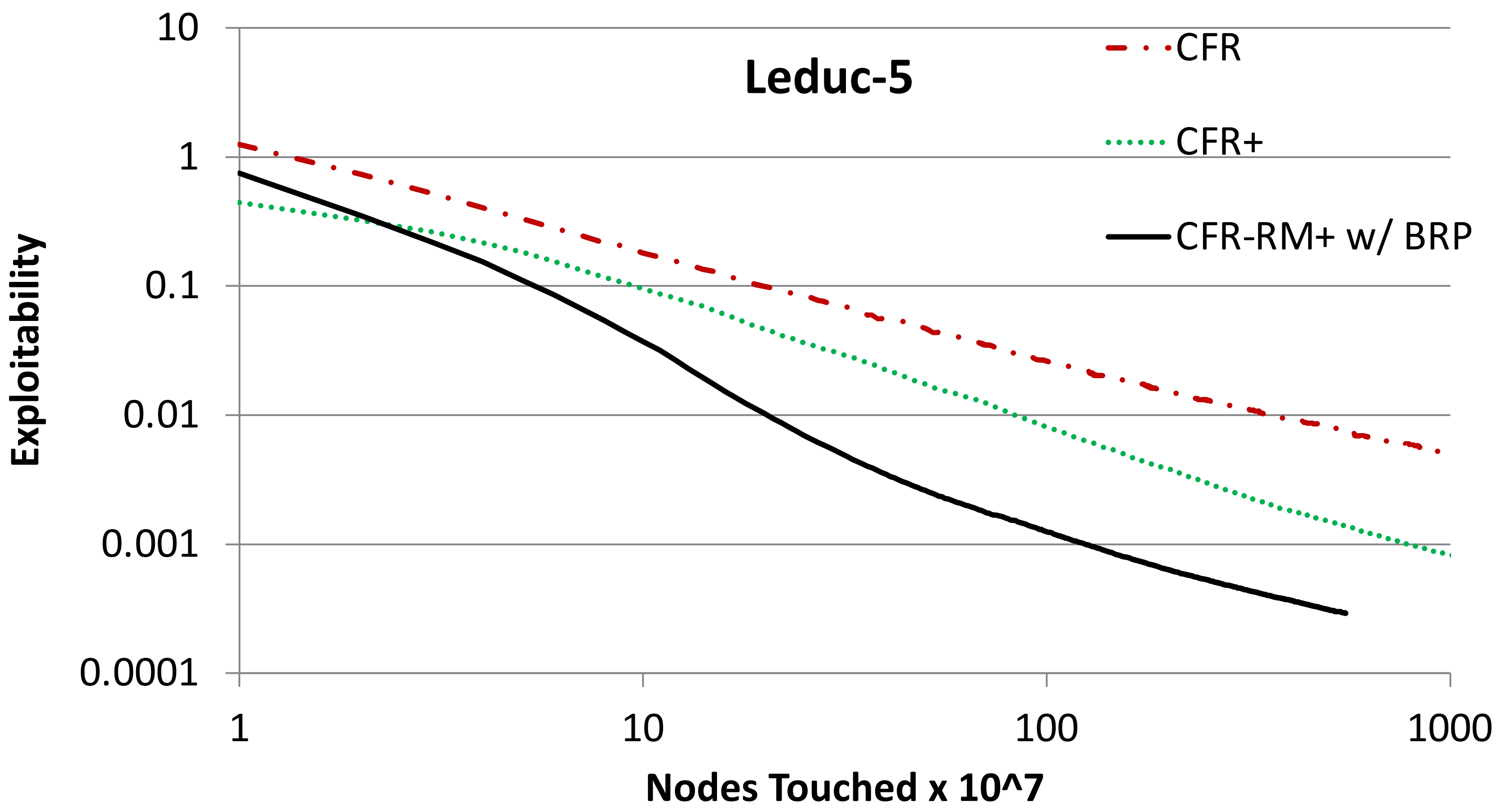


Theoretical Results for Best Response Pruning (BRP)

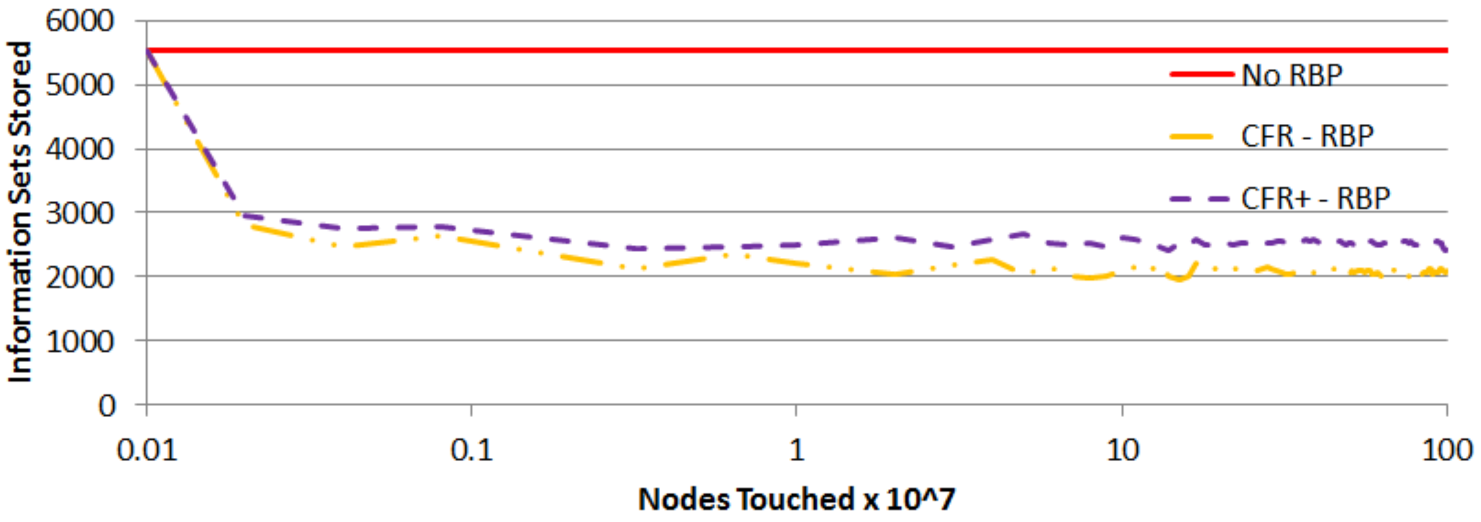
- The asymptotic time and space complexity of solving a game with BRP is not dependent on the **number of actions in the game**, but on the **number of actions that are part of a best response to an equilibrium**

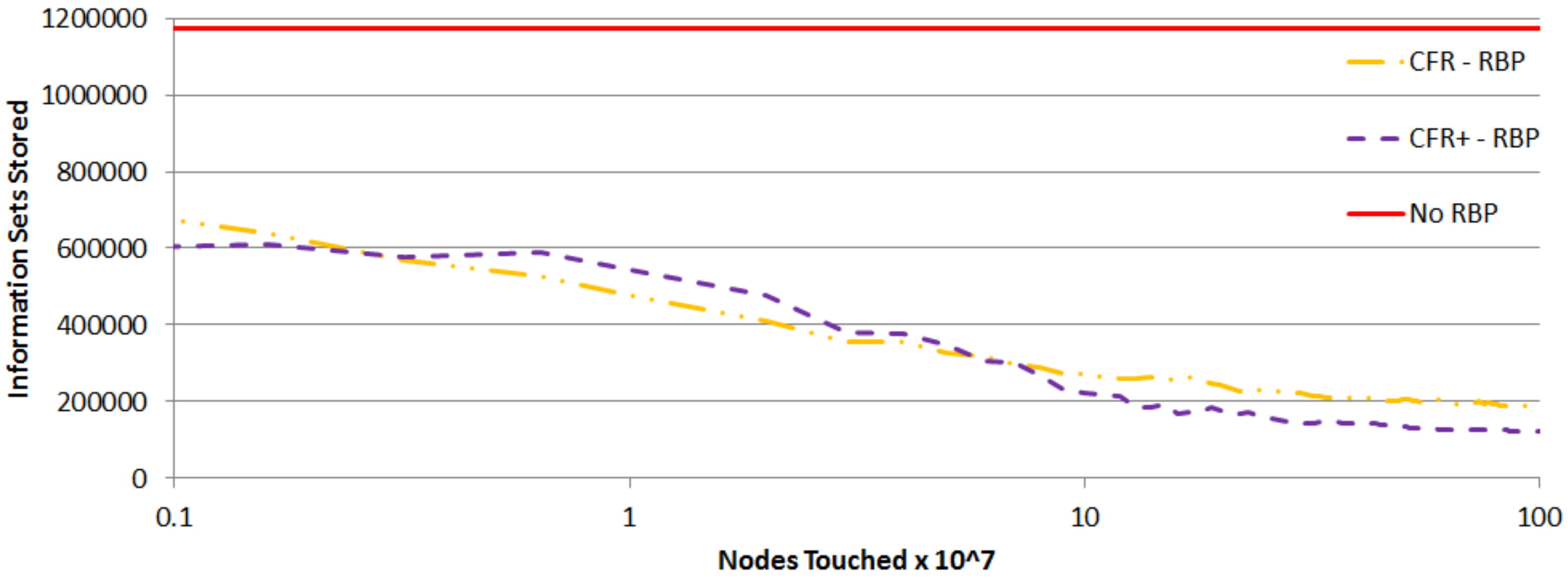
- This can be orders of magnitude smaller

Better Convergence with BRP

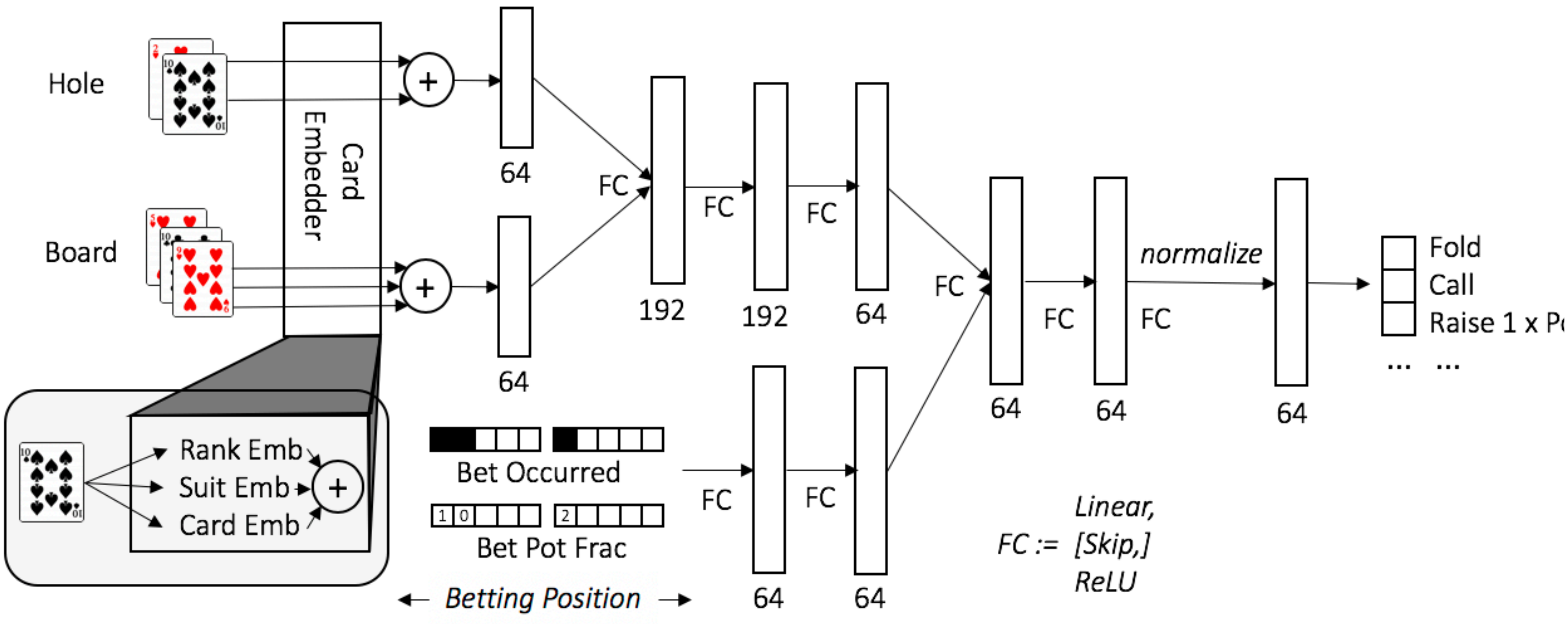


Using Less Memory with BRP

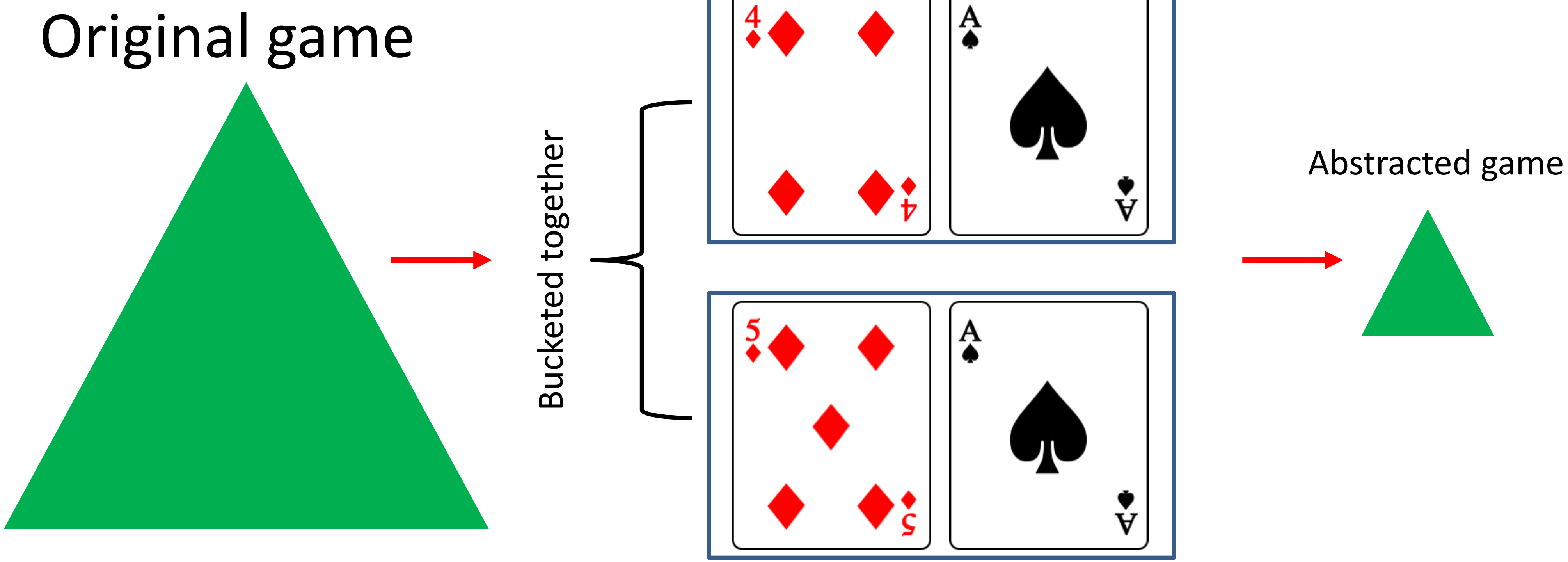




Scaling to Large Games with Deep CFR



Prior Approach: Abstraction in Games

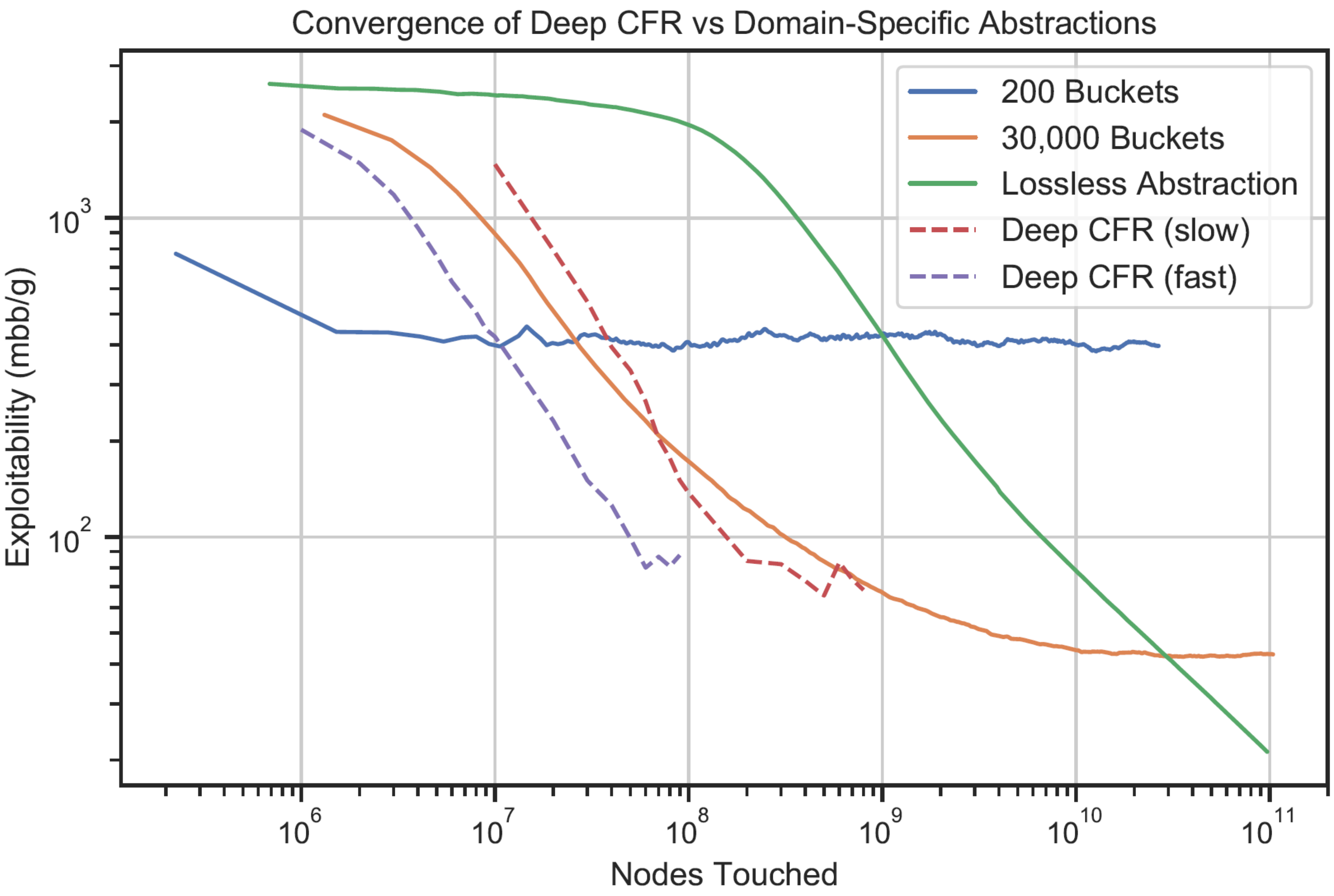


- Requires extensive domain knowledge
  - Several papers written on how to do abstraction just in poker
  - Difficult to extend to other games

Deep CFR

- **Input:** low-level features (visible cards, observed actions)
- **Output:** estimate of action regrets
- On each iteration:
  1. Collect samples of action regrets, add to a buffer
  2. Train a network to predict regrets
  3. Use network's regret estimates to play on next iteration
- **Theorem:** With arbitrarily high probability, Deep CFR converges to an ϵ-Nash equilibrium in two-player zero-sum games, where ϵ is determined by prediction error

Exploitability in Flop Hold'em (10^{11} nodes)



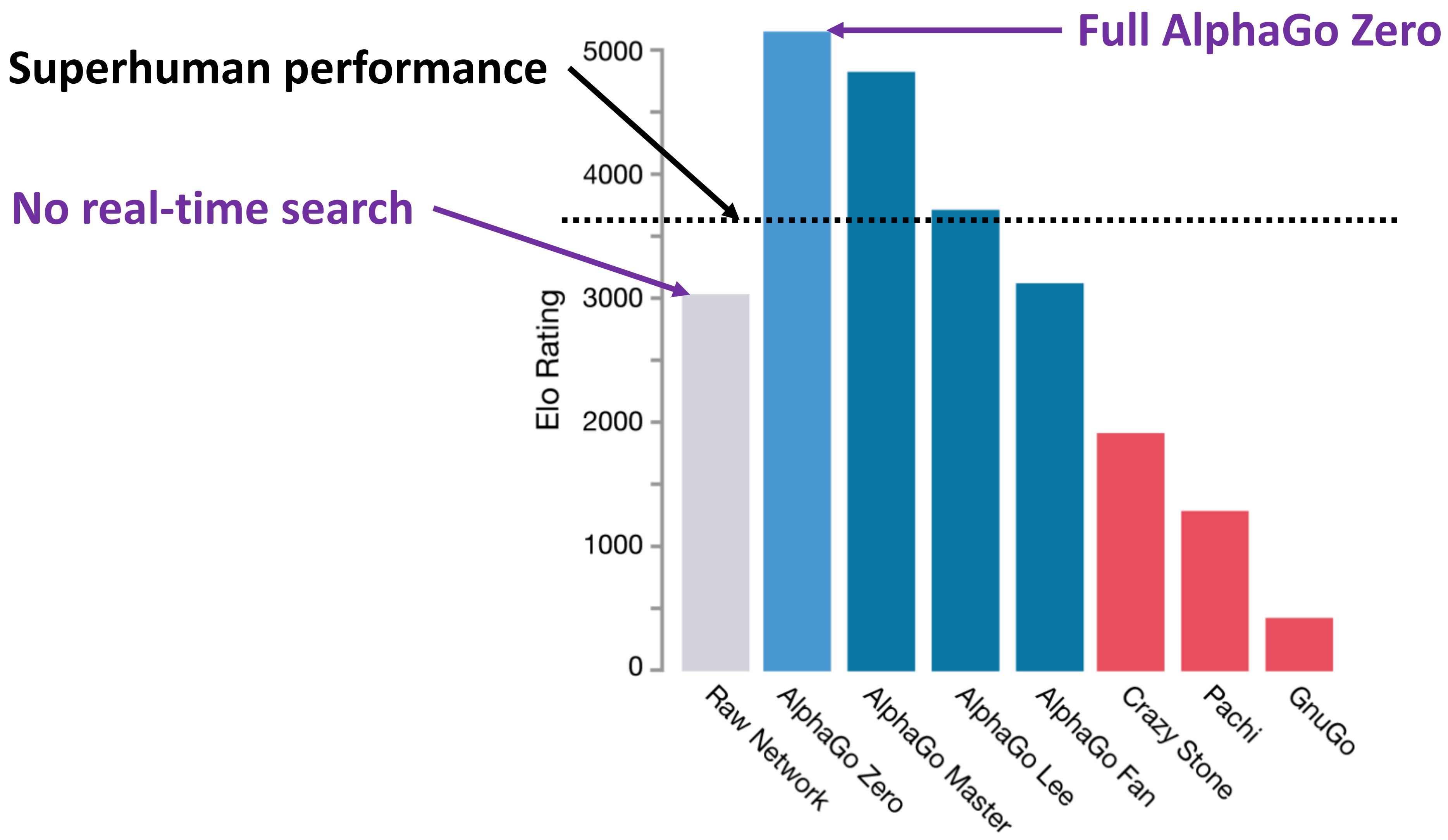
Experimental results in limit Texas hold'em

- Deep CFR produces superhuman performance in heads-up limit Texas hold'em poker
- Deep CFR outperforms Neural Fictitious Self Play (NFSP), the prior best deep RL algorithm for imperfect-info games [Heinrich & Silver arXiv-15]
  - Deep CFR is also much more sample efficient
- Deep CFR is competitive with domain-specific abstraction algorithms

Searching for a better strategy in real time

Image Credit: UC Berkeley CS-188 Lecture 6

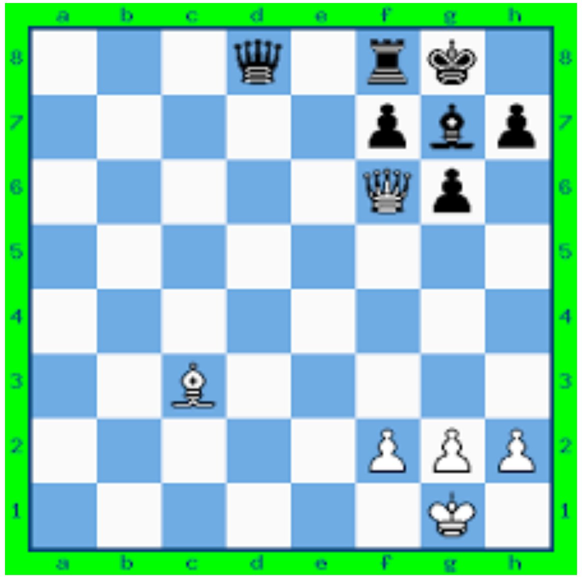
Real-time search is important



Search in Perfect-Information Games

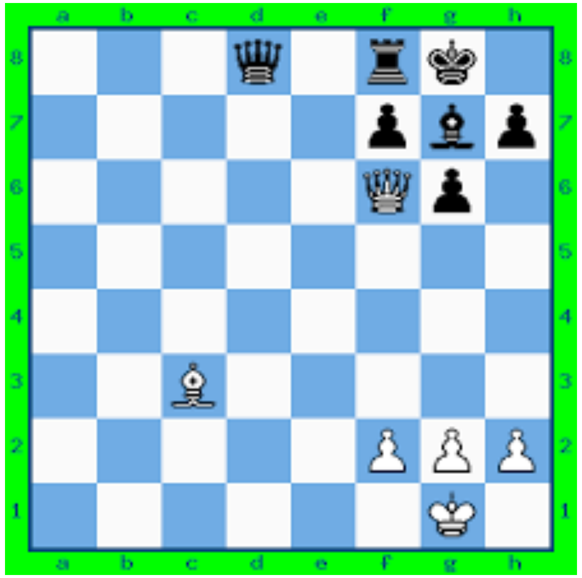
- In perfect-information games, the **value of a state** is the **unique** value resulting from backward induction

- A **value network** takes a state as input and outputs an estimate of the state value

$f_{white}($  $) = 1$

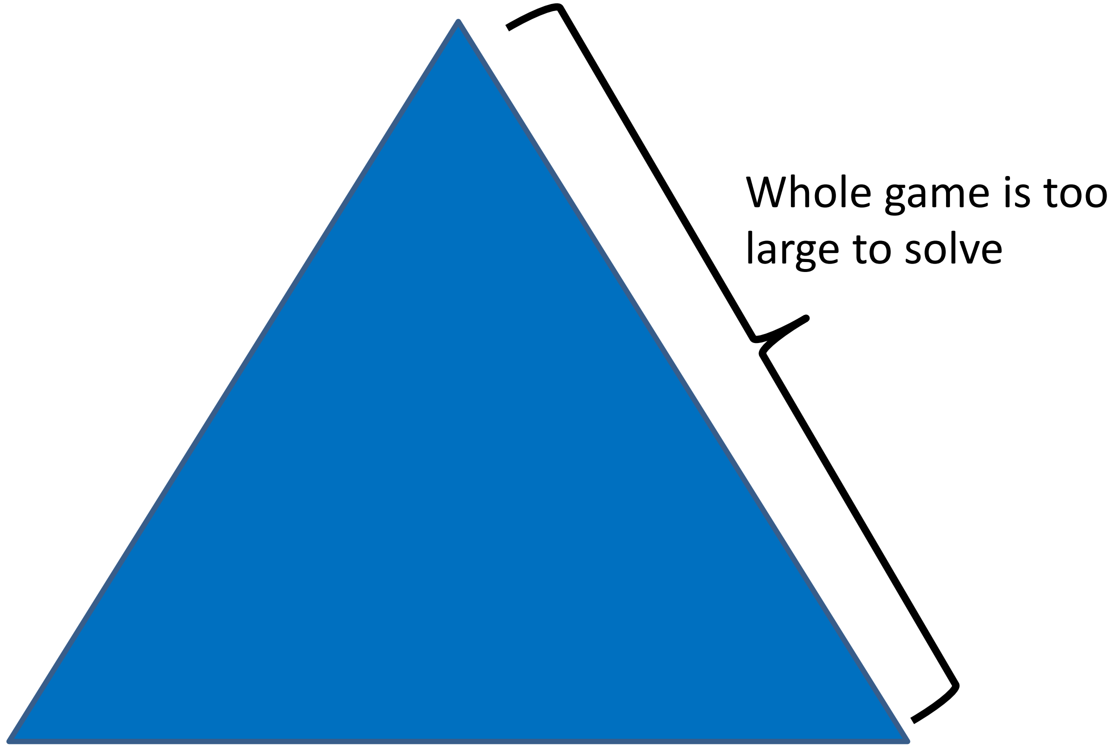
Search in Perfect-Information Games

- Where does the value network come from?
  - It can be a handcrafted heuristic function [early chess AI’s]
  - It can be learned by training on expert human games [AlphaGo]
  - It can be learned through self-play reinforcement learning [AlphaZero]

$f_{white}($  $) = 1$

Search in Perfect-Information Games

- In principle, backward induction alone can solve Chess
- But this would be far too expensive in practice

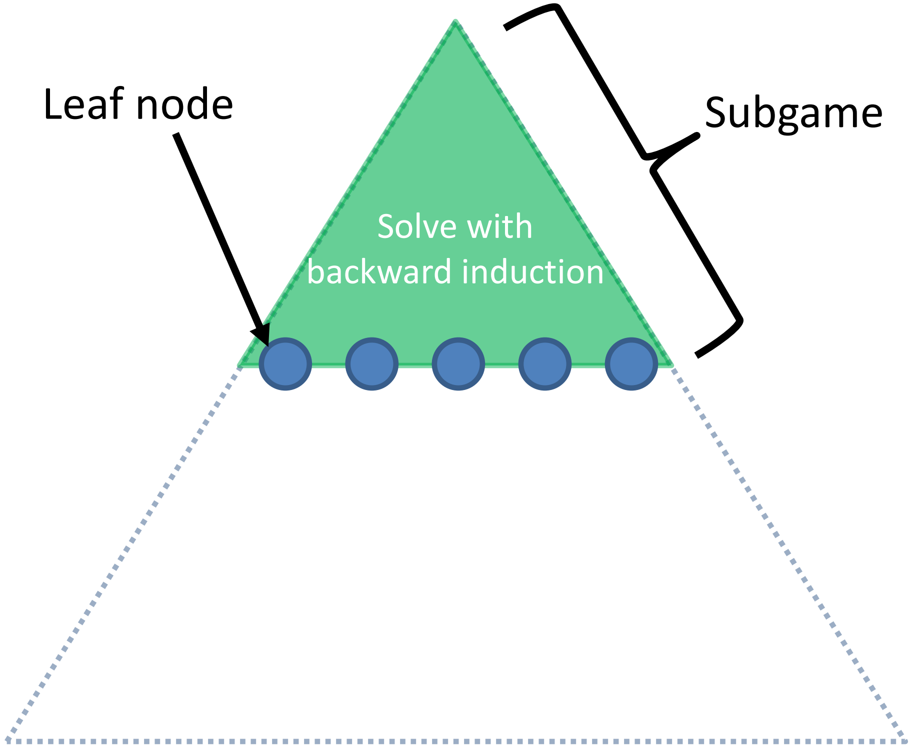


Search in Perfect-Information Games

- Instead, chess AI’s do **search:**
  1. Look ~10 moves ahead
  2. Estimate those state values using the value network
  3. Do backward induction using those state values (ignore the game below those states)

- In other words, solve a **subgame**

- If the value network is perfect, this computes the optimal action

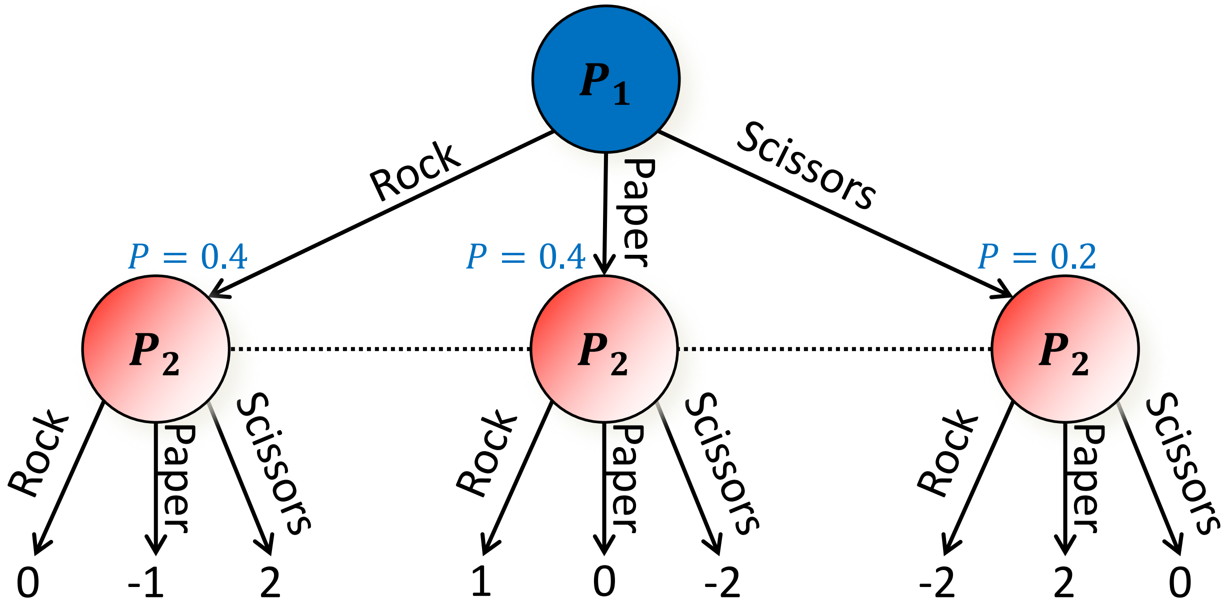


Why is search in imperfect-information games hard?

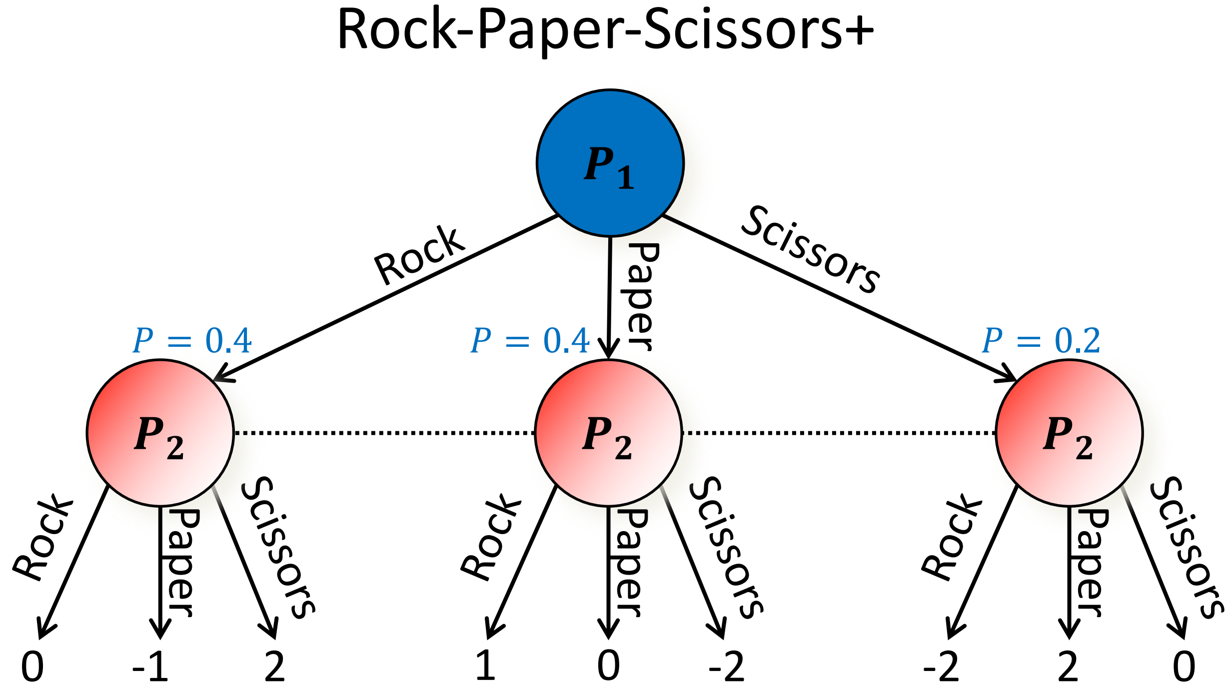
Because “states” don’t have well-defined values

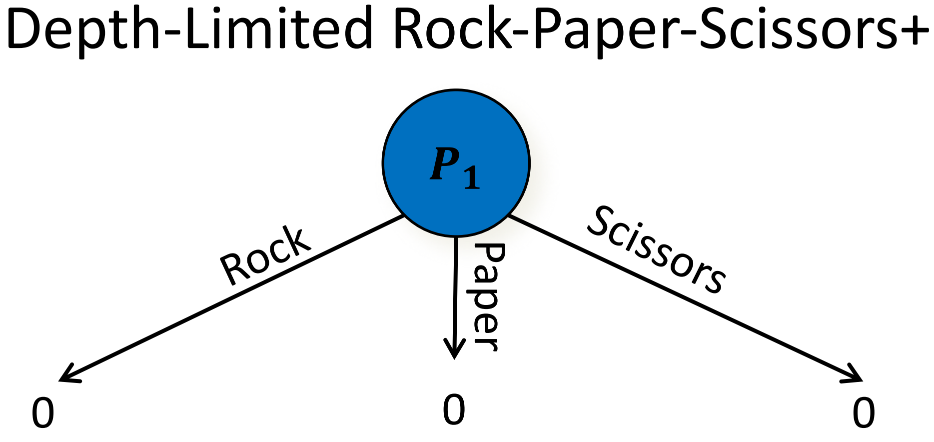
Depth-Limited Search

Rock-Paper-Scissors+



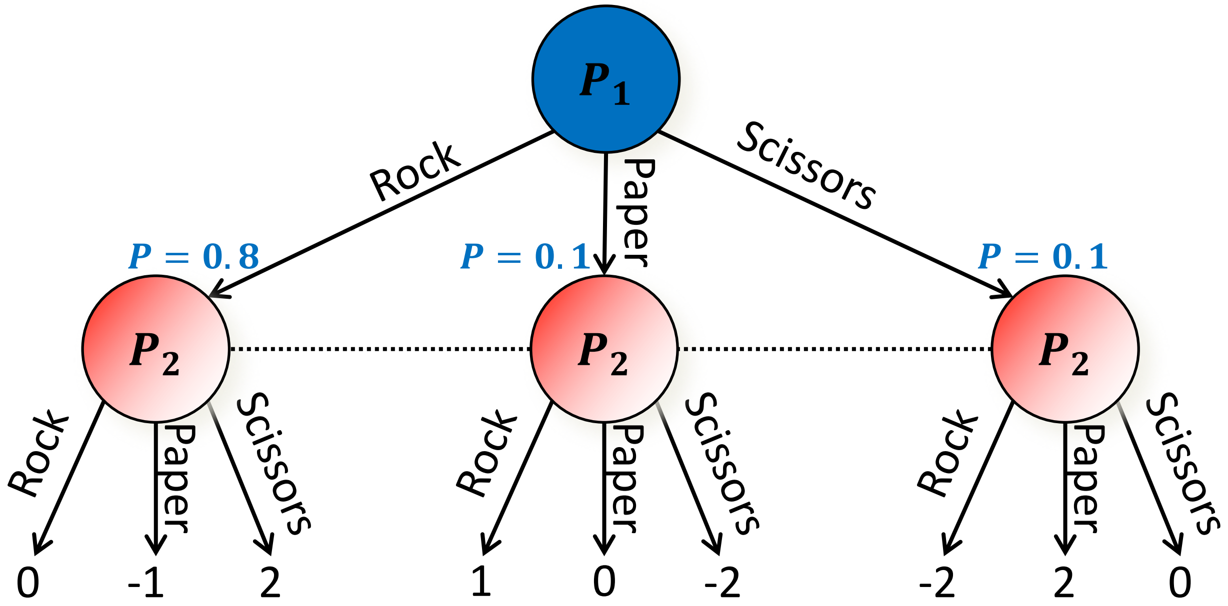
Depth-Limited Search



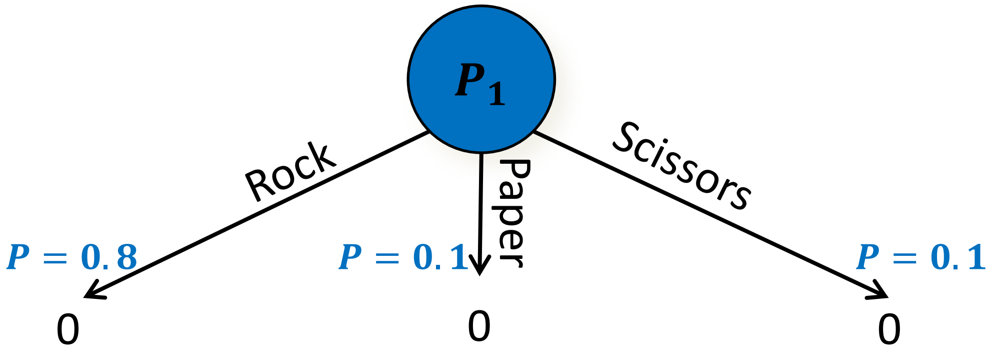


Depth-Limited Search

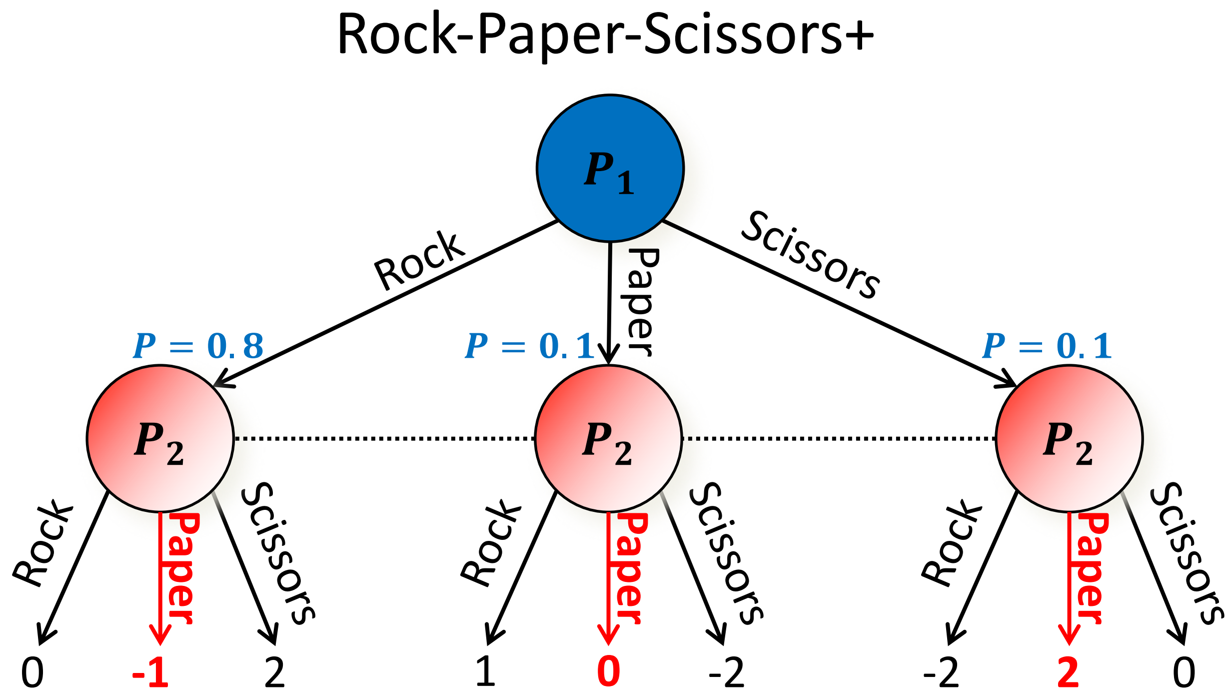
Rock-Paper-Scissors+

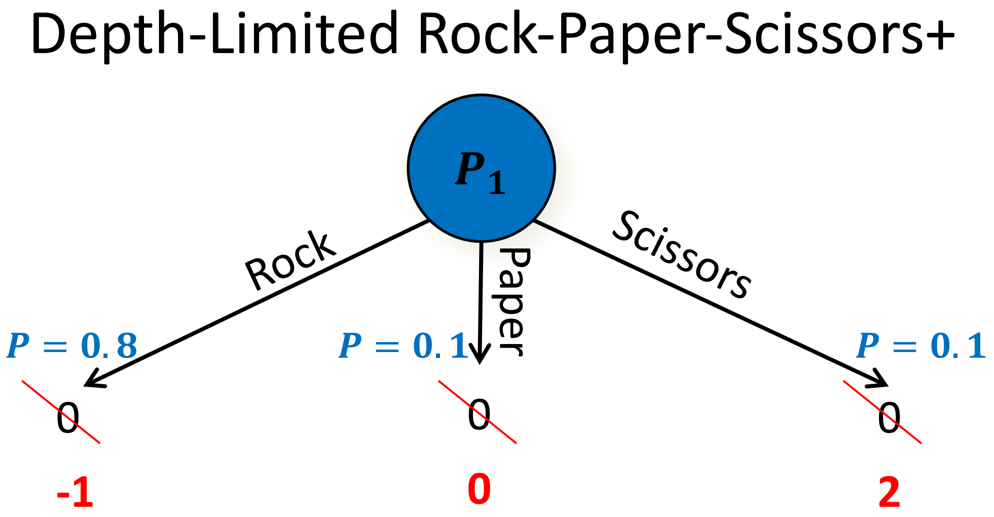


Depth-Limited Rock-Paper-Scissors+

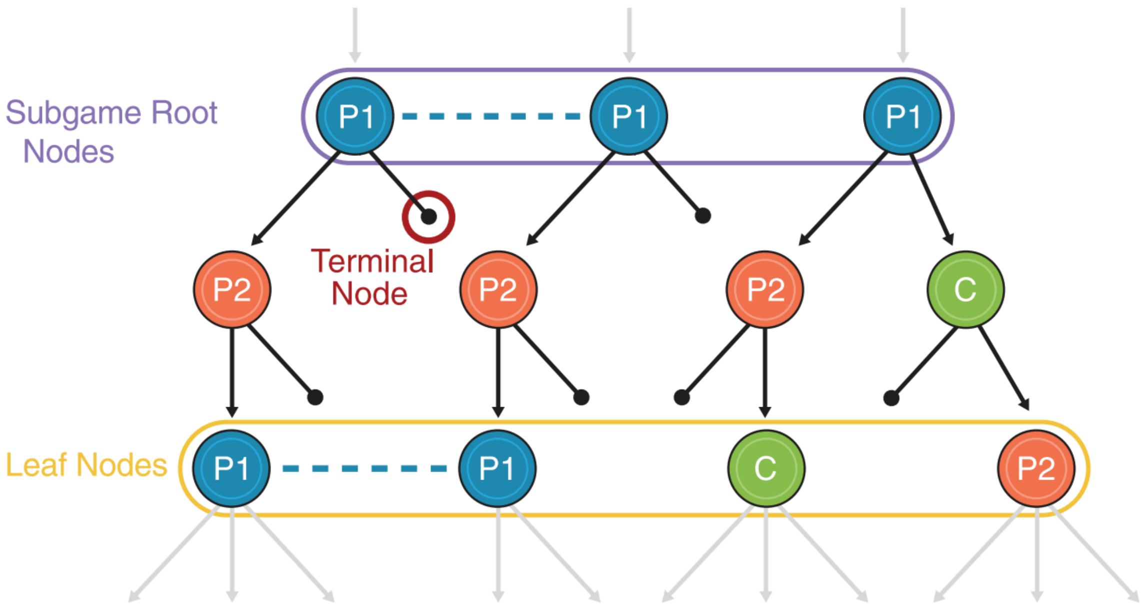


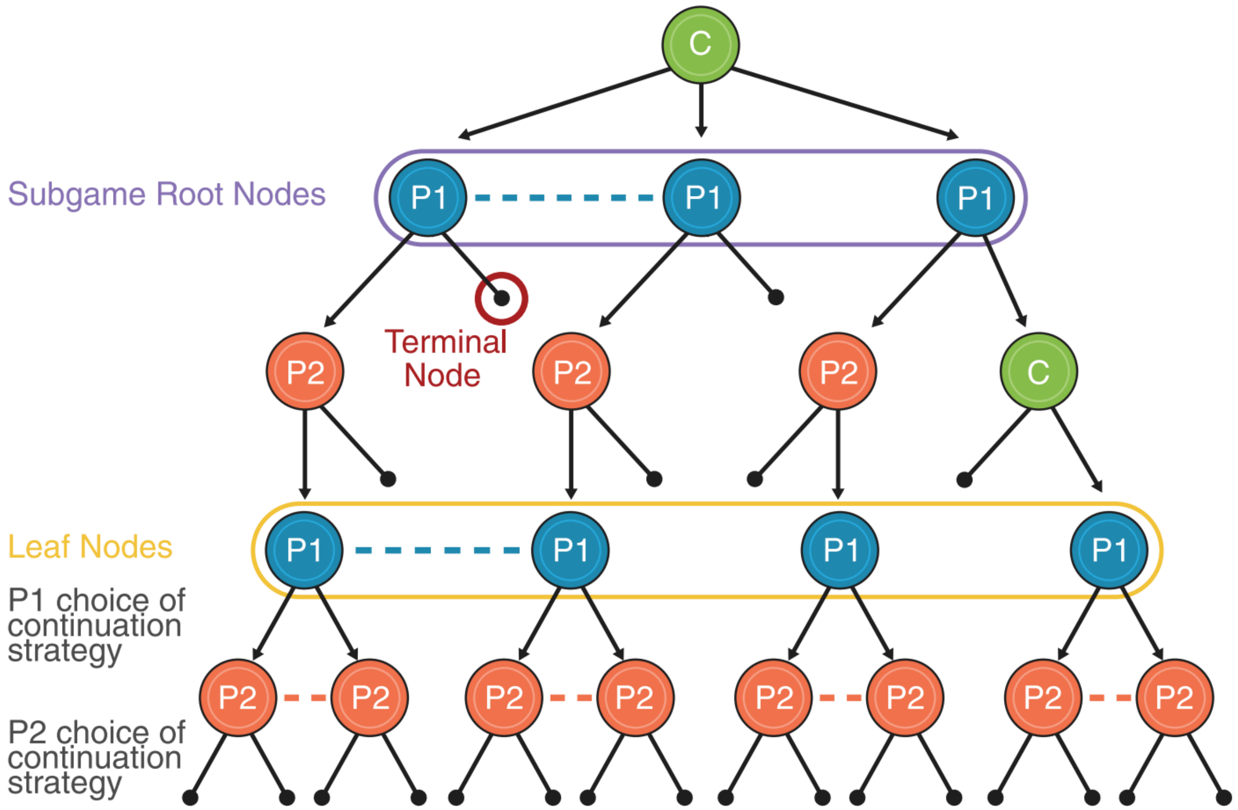
Depth-Limited Search



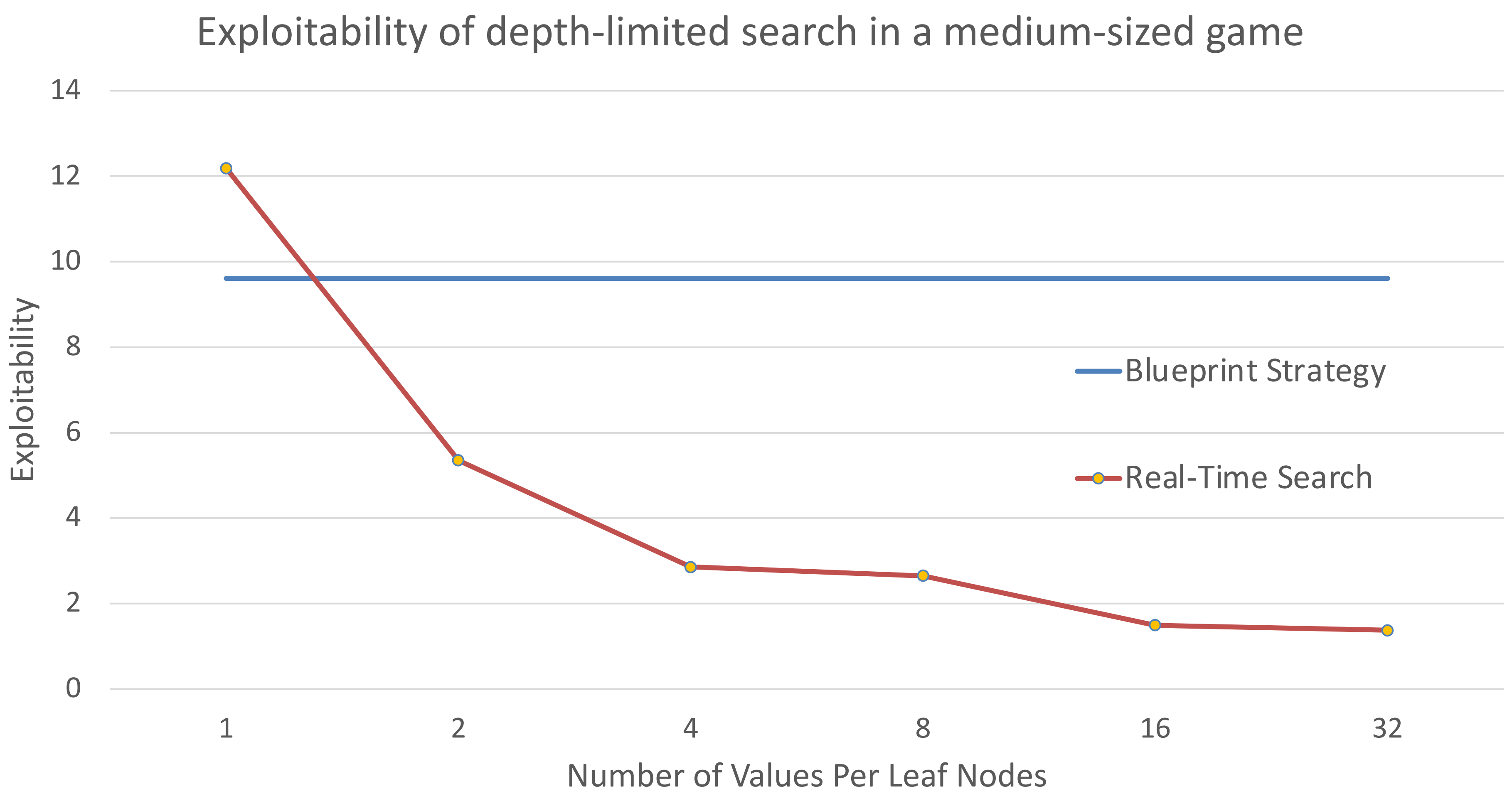


Depth-Limited Search in Pluribus

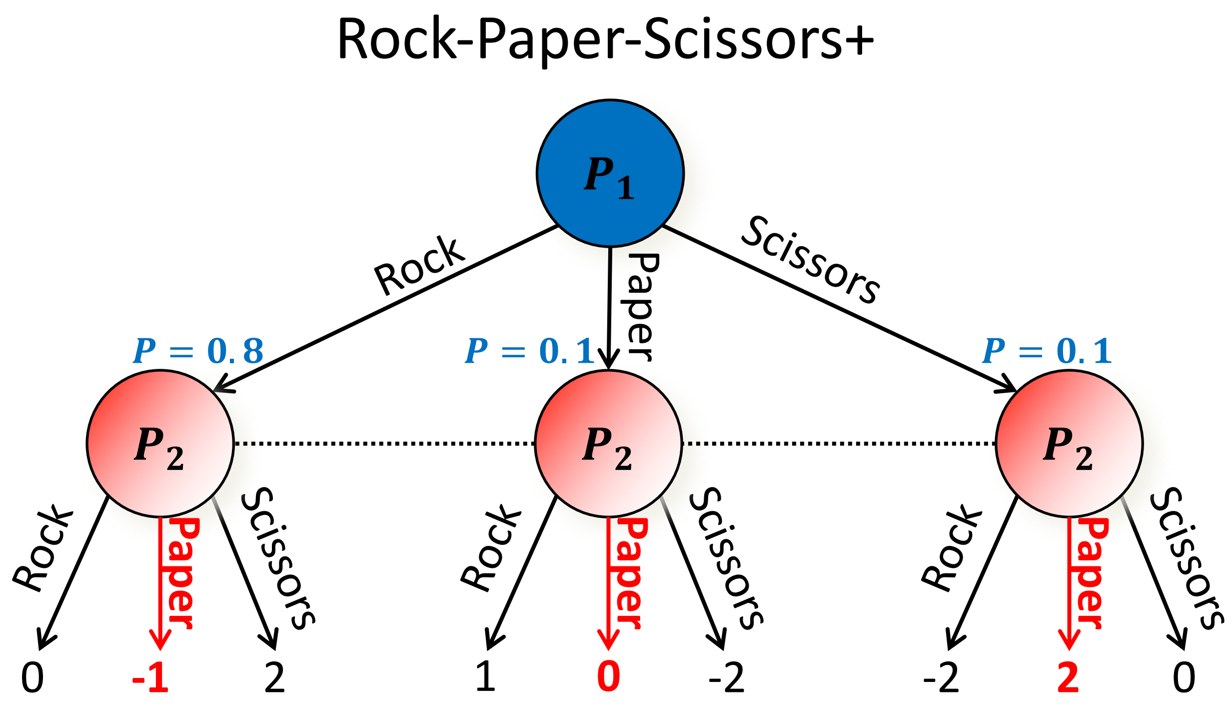


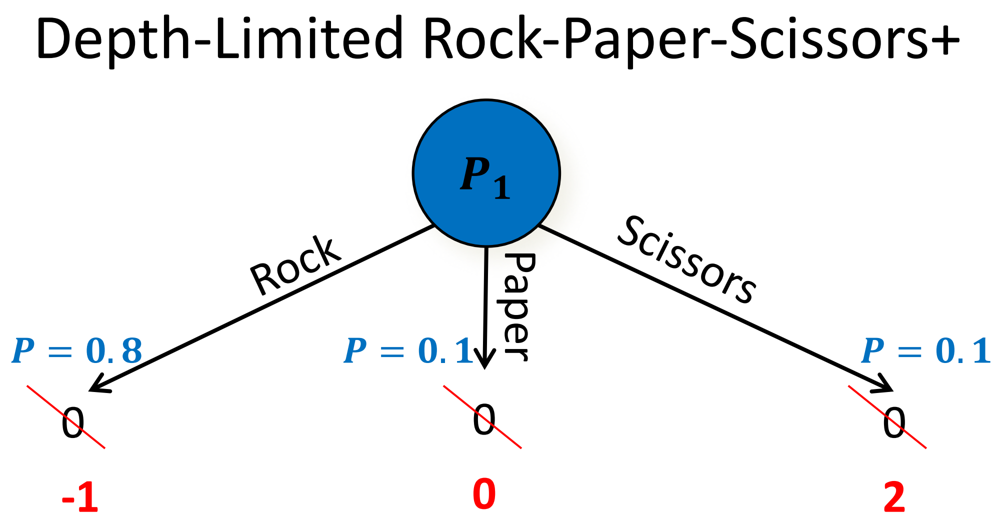


Exploitability Measurements



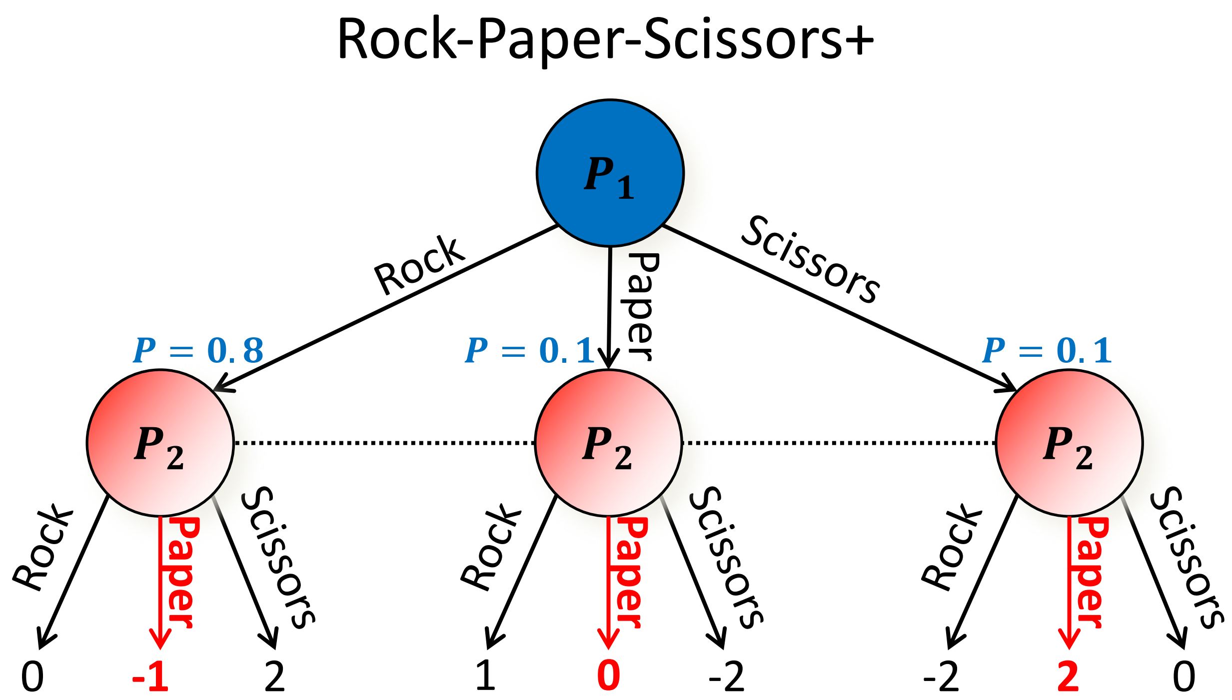
Search in Imperfect-Information Games

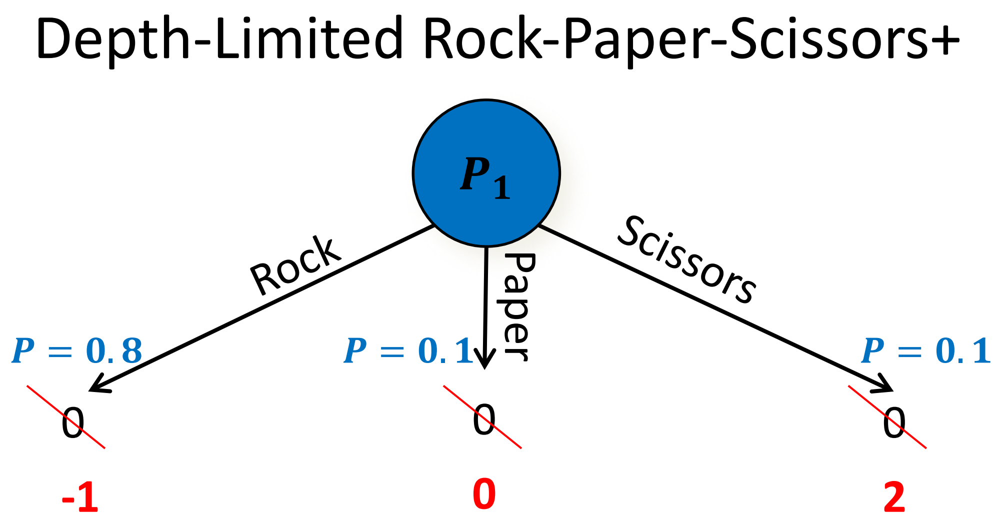




- Another solution: condition value on **probability distribution over possible states** [Nayyar et al. IEEE-13, Moravcik et al. Science-17]
  - $v(Rock)$ is not well-defined
  - $v([0.8\ Rock, 0.1\ Paper, 0.1\ Scissors]) = -0.6$
- Idea originated in Dec-POMDP research, and later used in poker AIs including DeepStack

Search in Imperfect-Information Games

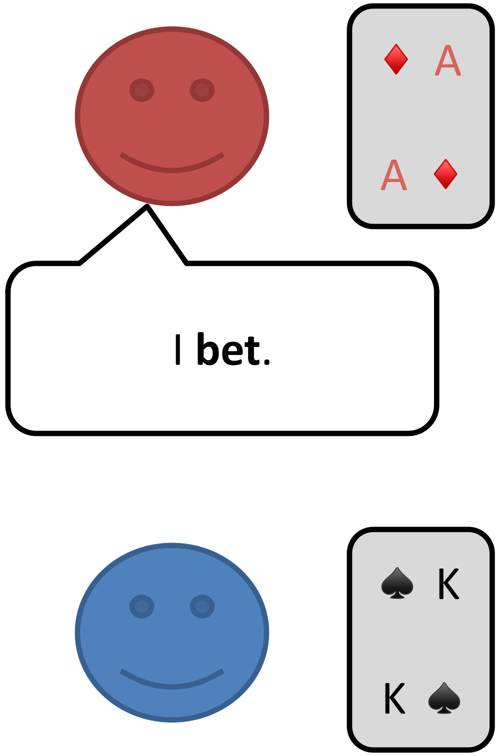


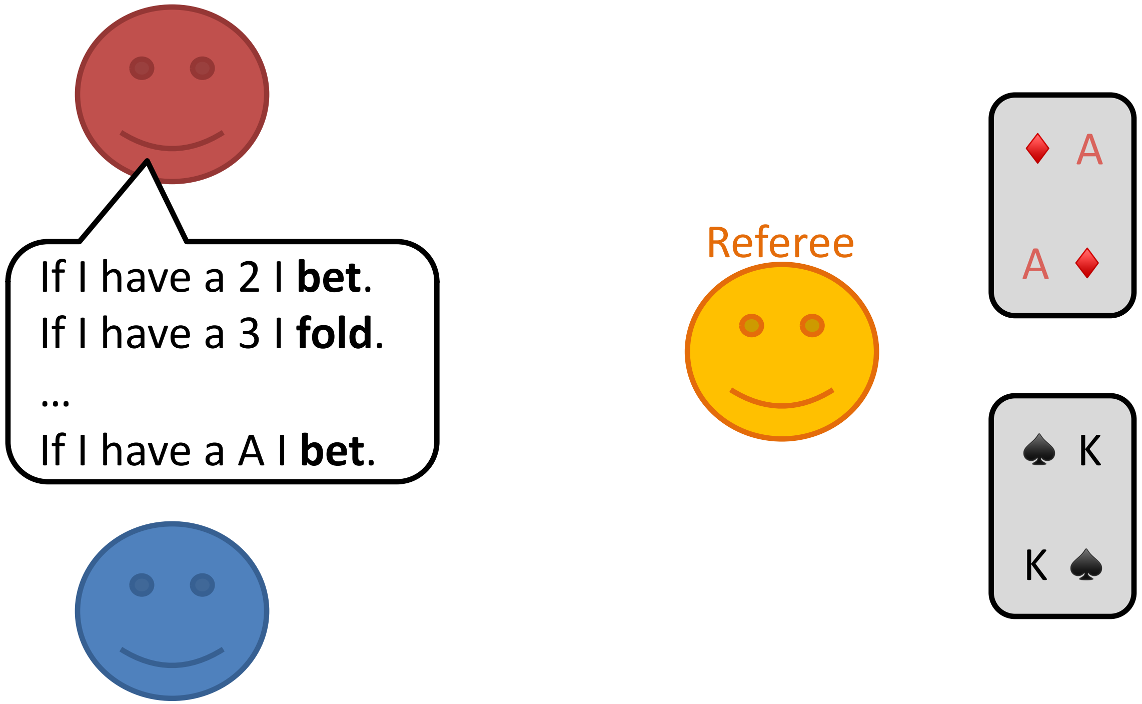


Critical assumption: Our entire policy is **common knowledge**, but the outcomes of random processes are **not** common knowledge

Converting imperfect-information games to continuous-state perfect-information games

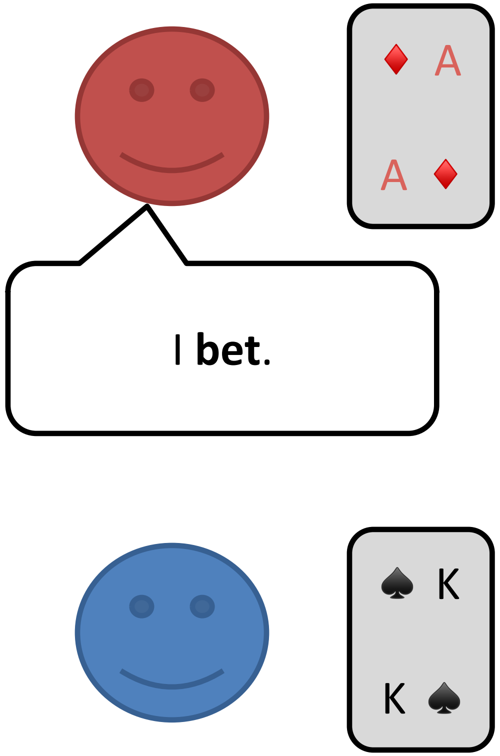
Discrete State Representation

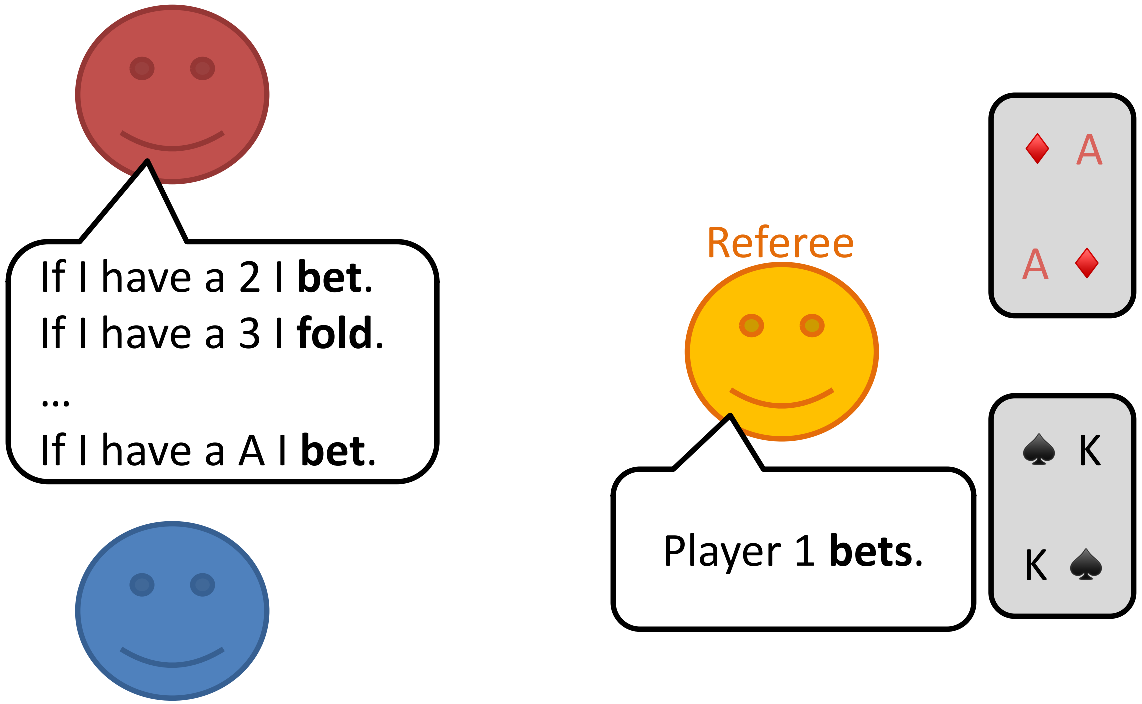




Converting imperfect-information games to continuous-state perfect-information games

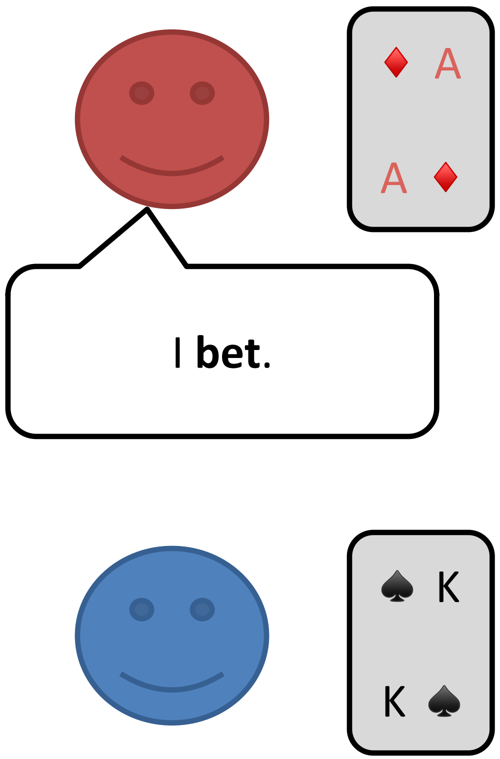
Discrete State Representation

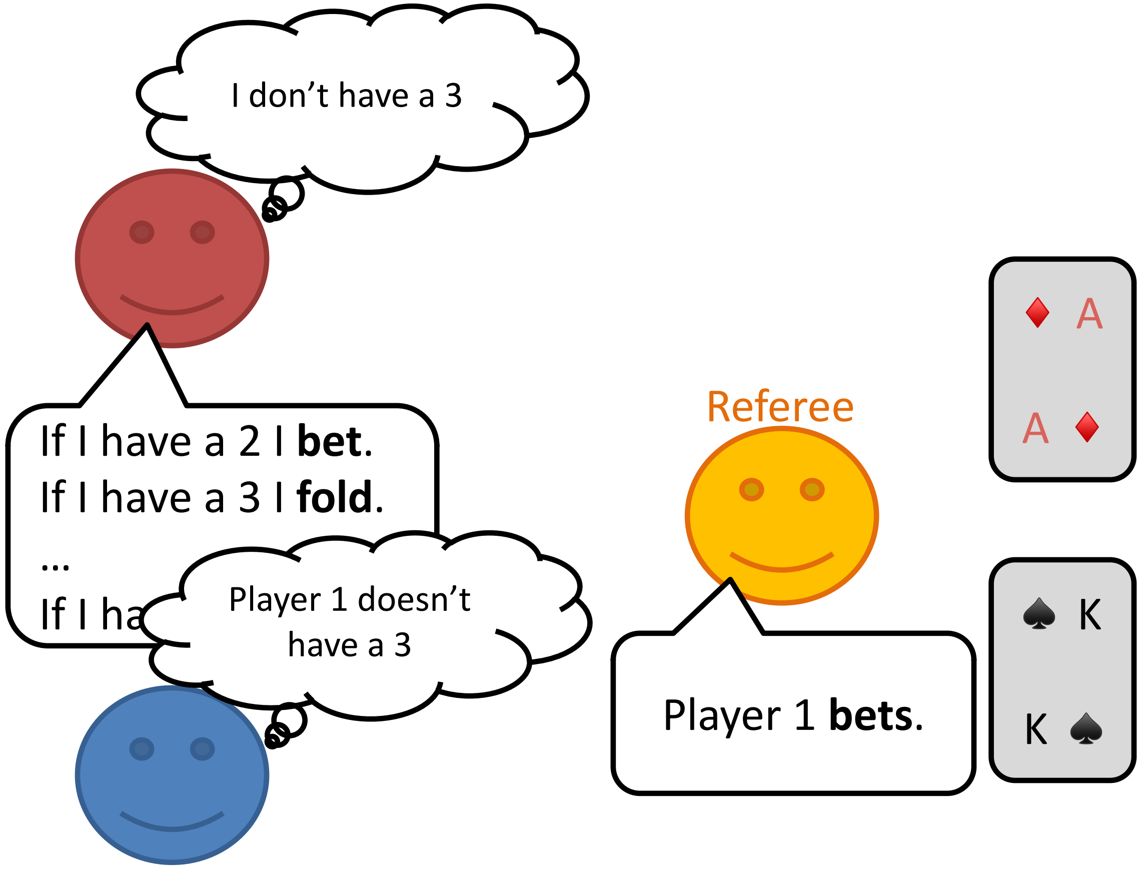




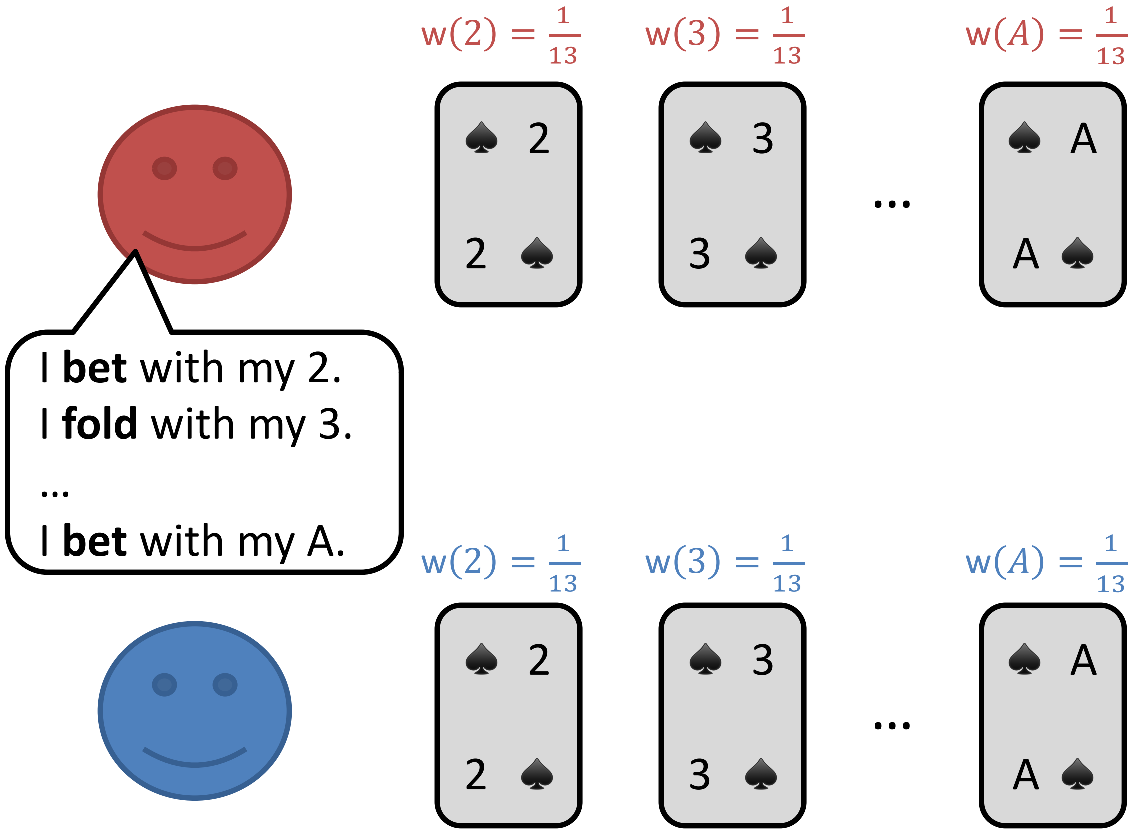
Converting imperfect-information games to continuous-state perfect-information games

Discrete State Representation





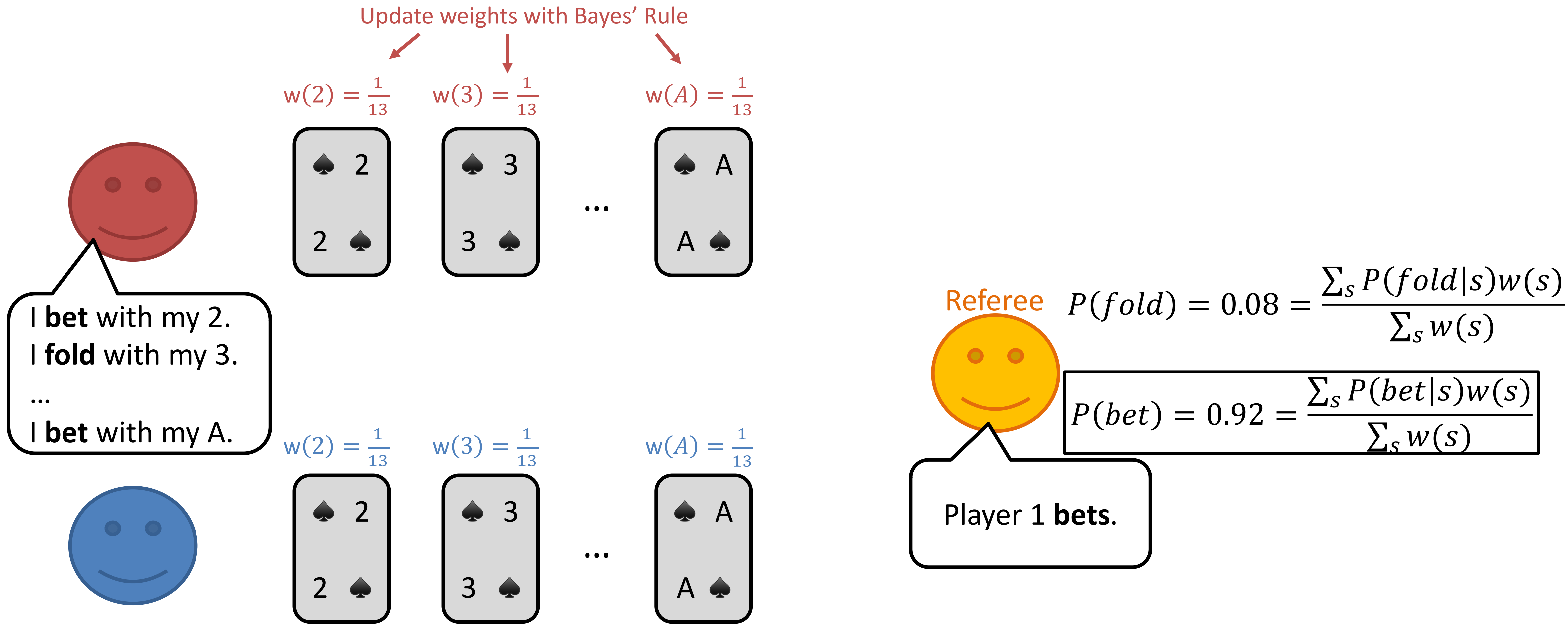
Converting imperfect-information games to continuous-state perfect-information games



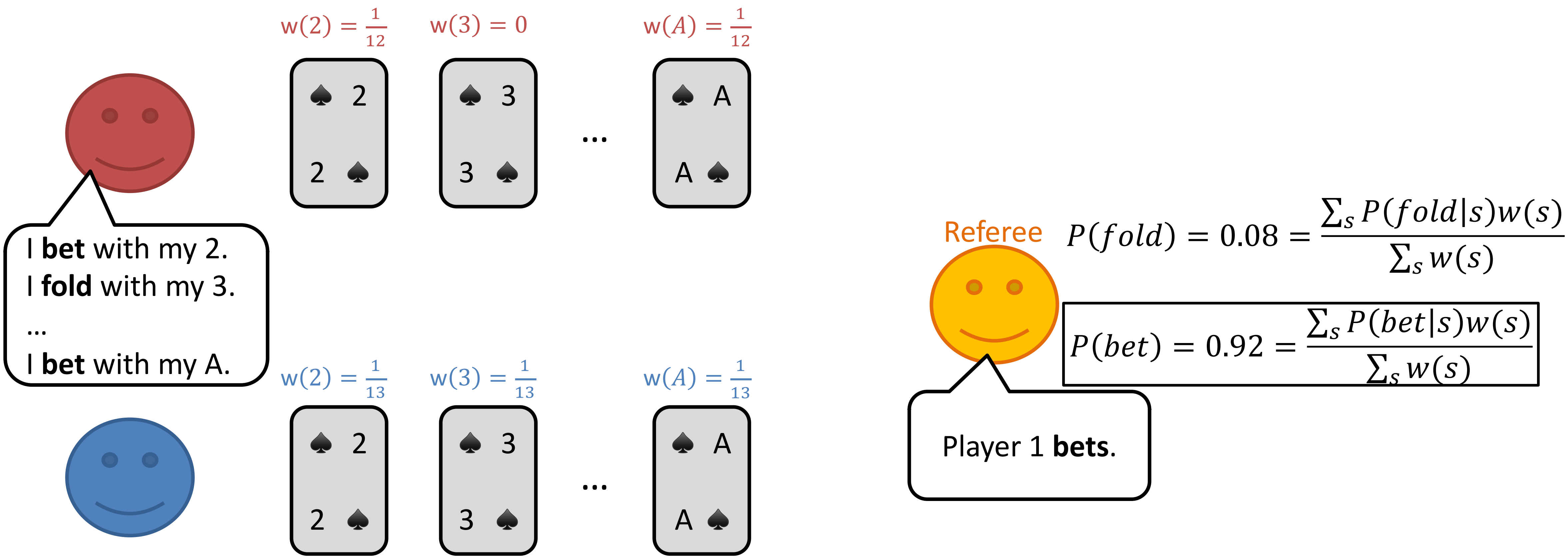
$$P(fold) = 0.08 = \frac{\sum_s P(fold|s)w(s)}{\sum_s w(s)}$$

$$P(bet) = 0.92 = \frac{\sum_s P(bet|s)w(s)}{\sum_s w(s)}$$

Converting imperfect-information games to continuous-state perfect-information games

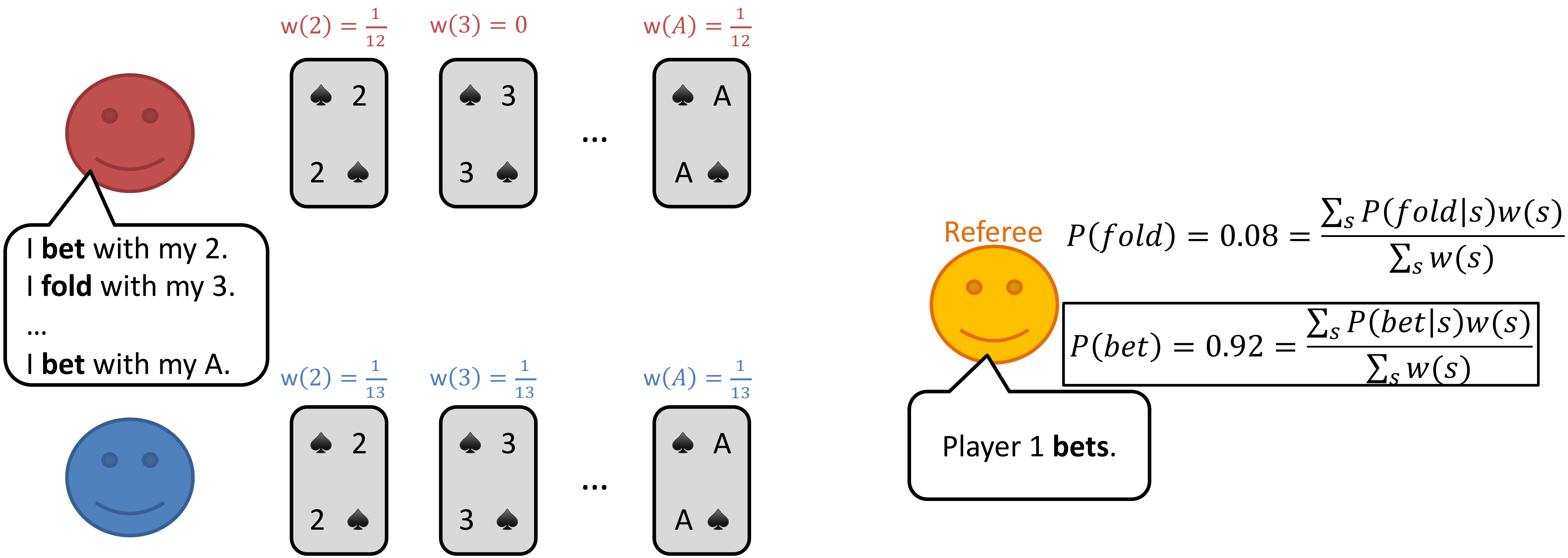


Converting imperfect-information games to continuous-state perfect-information games



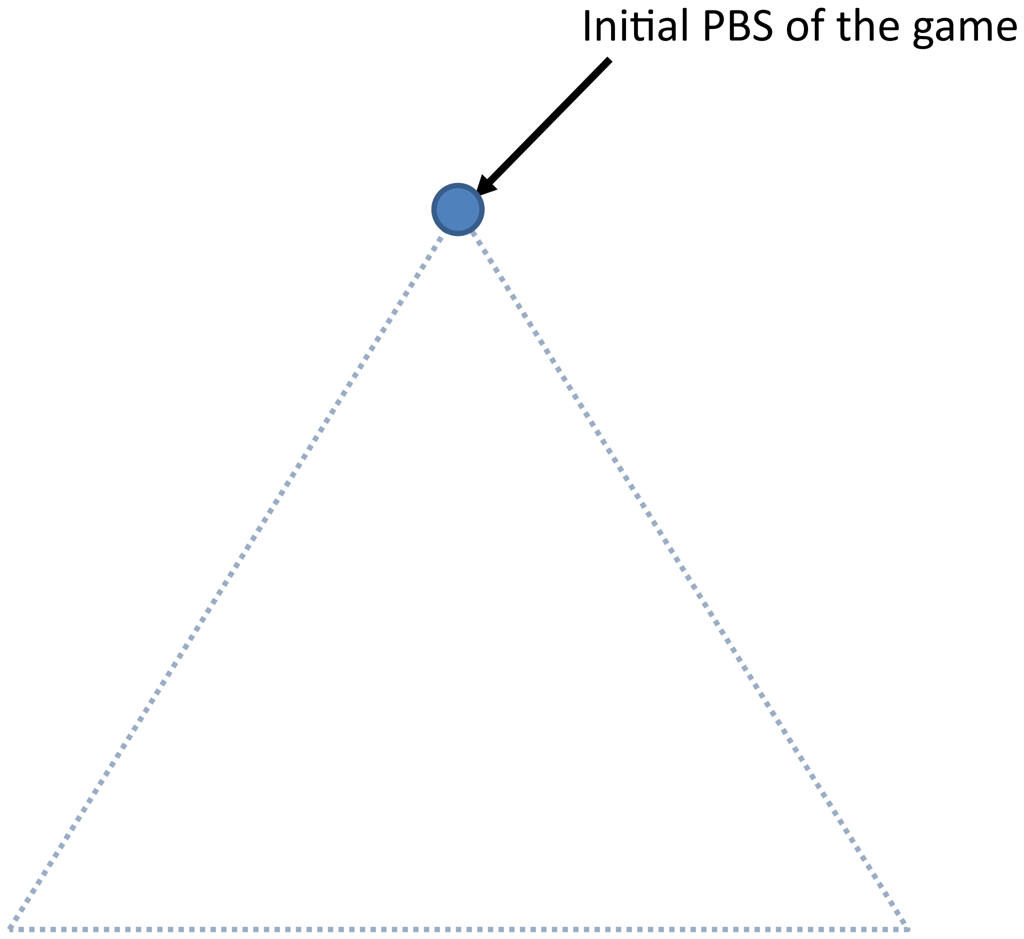
Converting imperfect-information games to continuous-state perfect-information games

Public Belief State (PBS) Representation



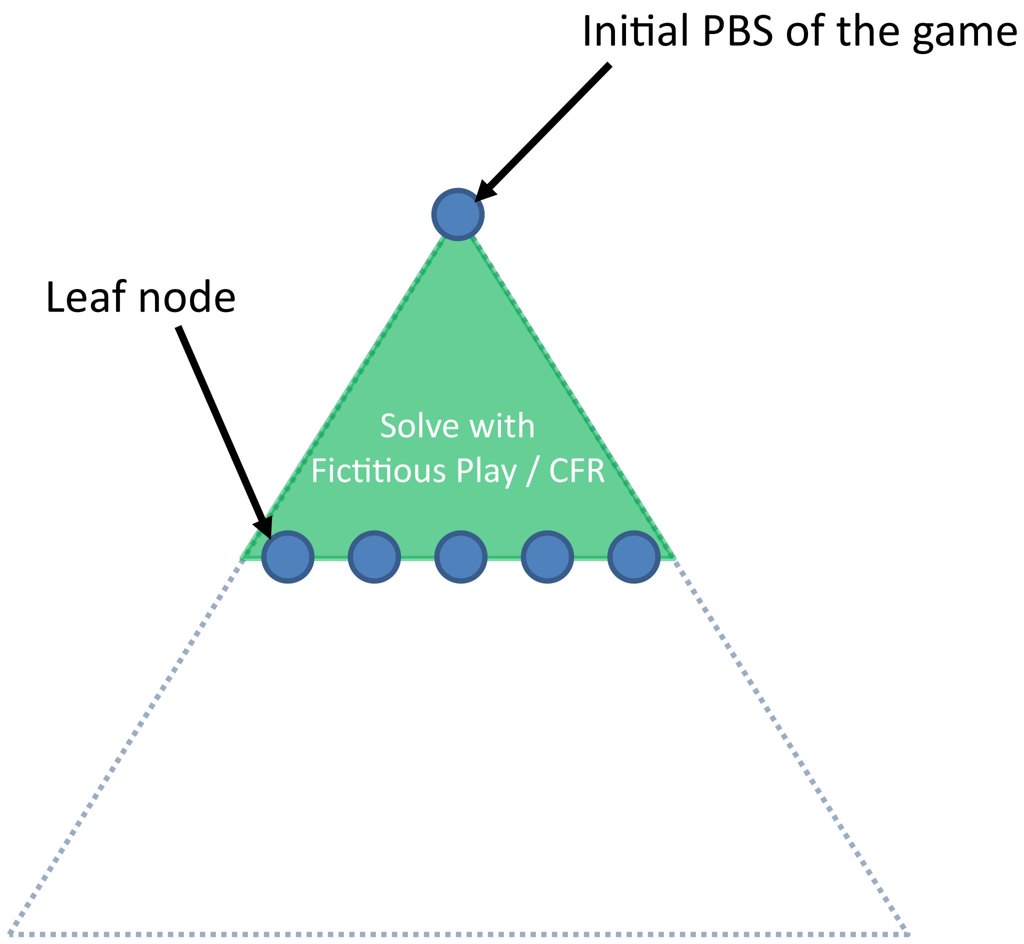
ReBeL

- Whenever an agent acts, generate a **discrete** subgame and solve it



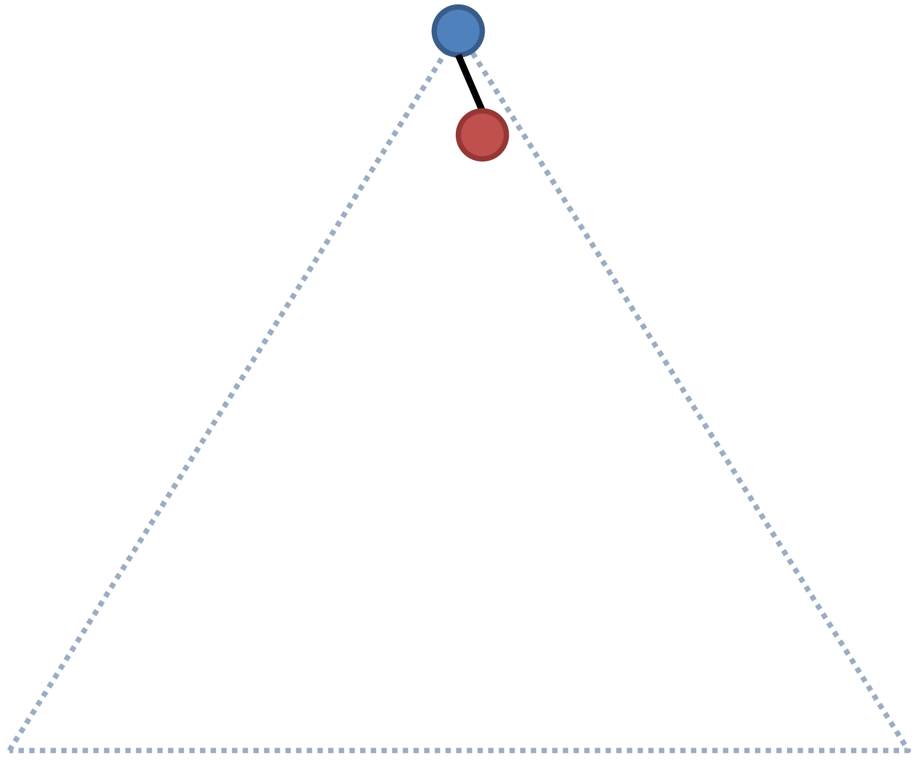
ReBeL

- Whenever an agent acts, generate a **discrete** subgame and solve it
  - Solve using Fictitious Play or CFR
  - Leaf values come from PBS value net
  - Take next action



ReBeL

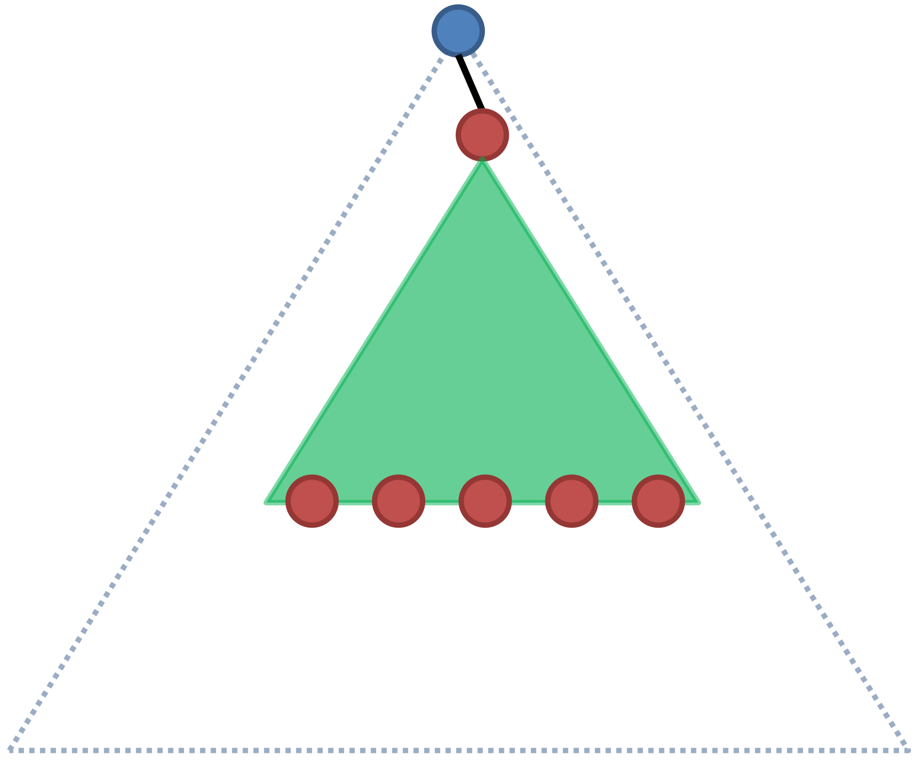
- Whenever an agent acts, generate a **discrete** subgame and solve it
  - Solve using Fictitious Play or CFR
  - Leaf values come from PBS value net
  - Take next action



ReBeL

- Whenever an agent acts, generate a **discrete** subgame and solve it
  - Solve using Fictitious Play or CFR
  - Leaf values come from PBS value net
  - Take next action

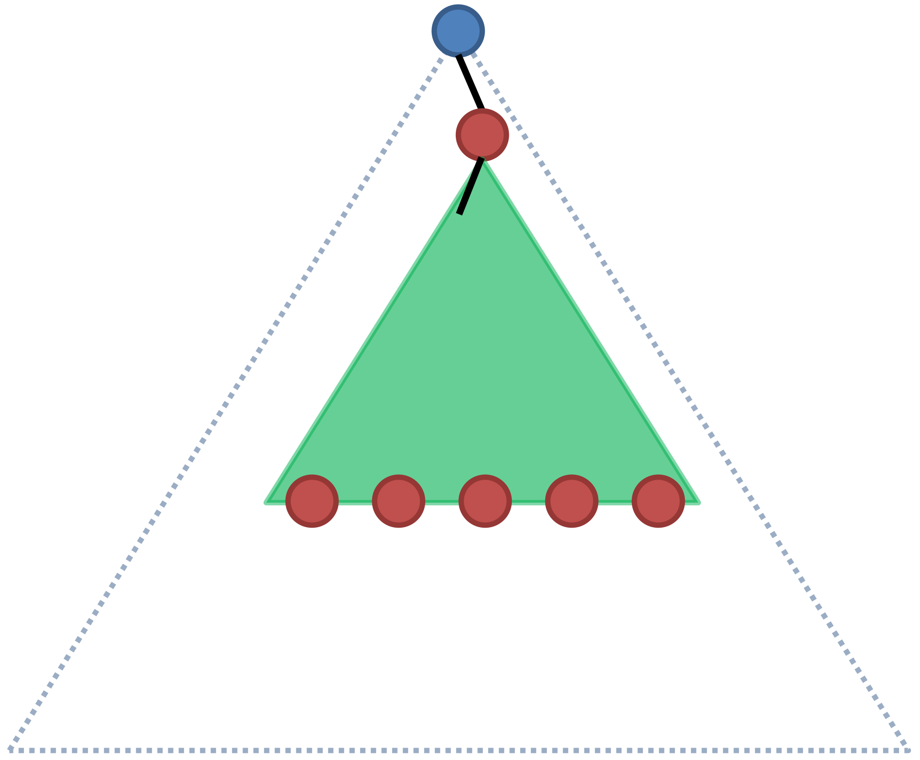
- Repeat until end of game



ReBeL

- Whenever an agent acts, generate a **discrete** subgame and solve it
  - Solve using Fictitious Play or CFR
  - Leaf values come from PBS value net
  - Take next action

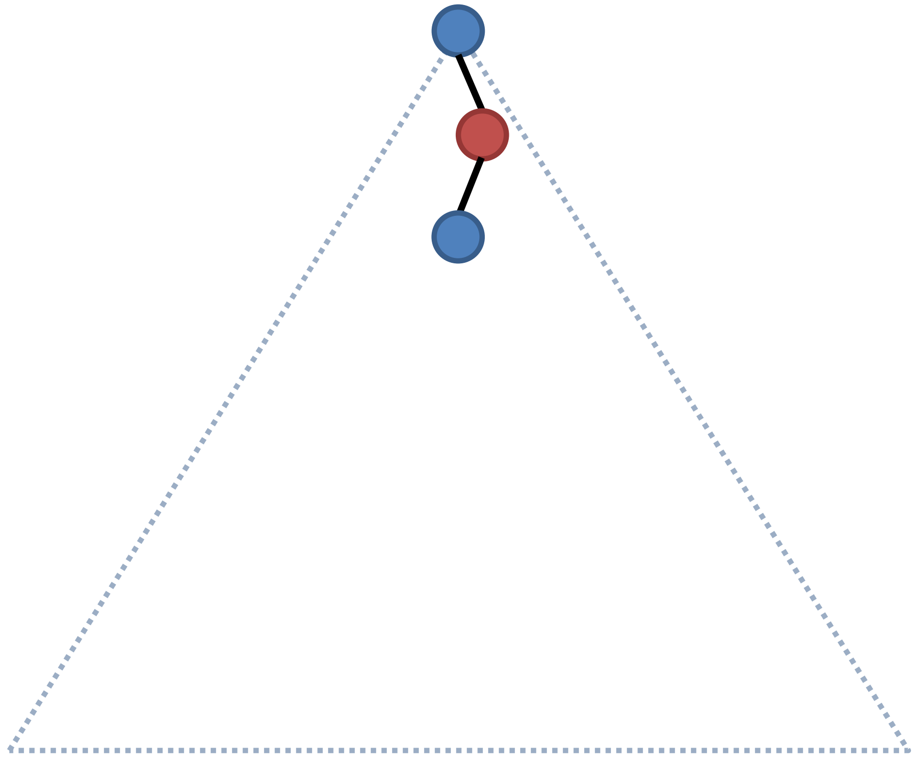
- Repeat until end of game



ReBeL

- Whenever an agent acts, generate a **discrete** subgame and solve it
  - Solve using Fictitious Play or CFR
  - Leaf values come from PBS value net
  - Take next action

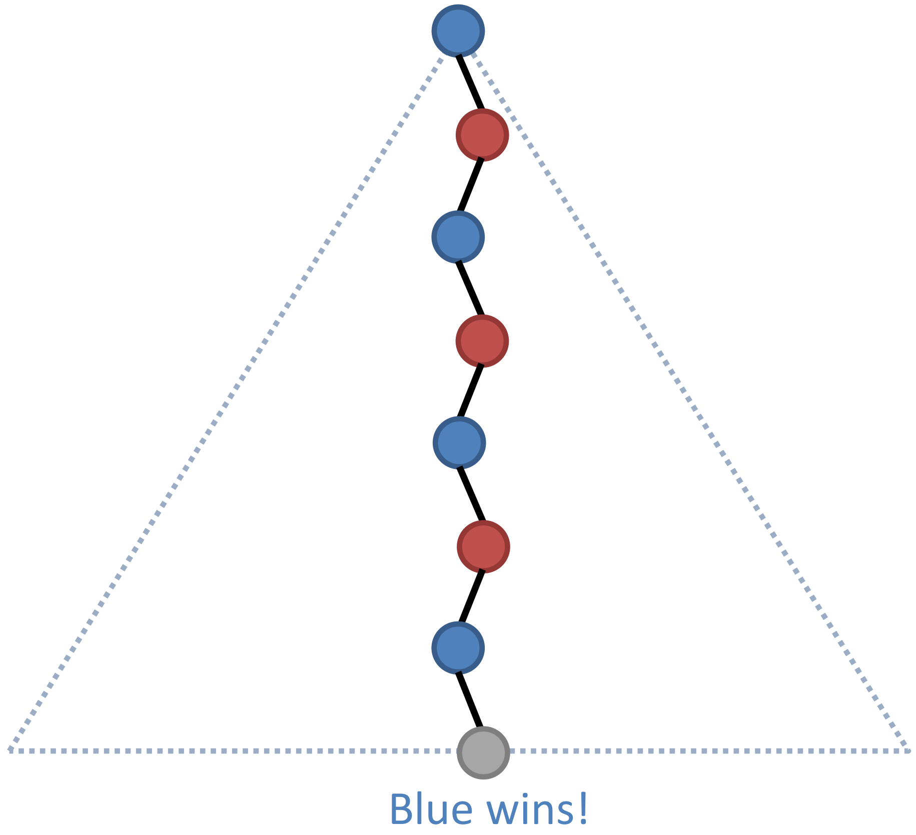
- Repeat until end of game



ReBeL

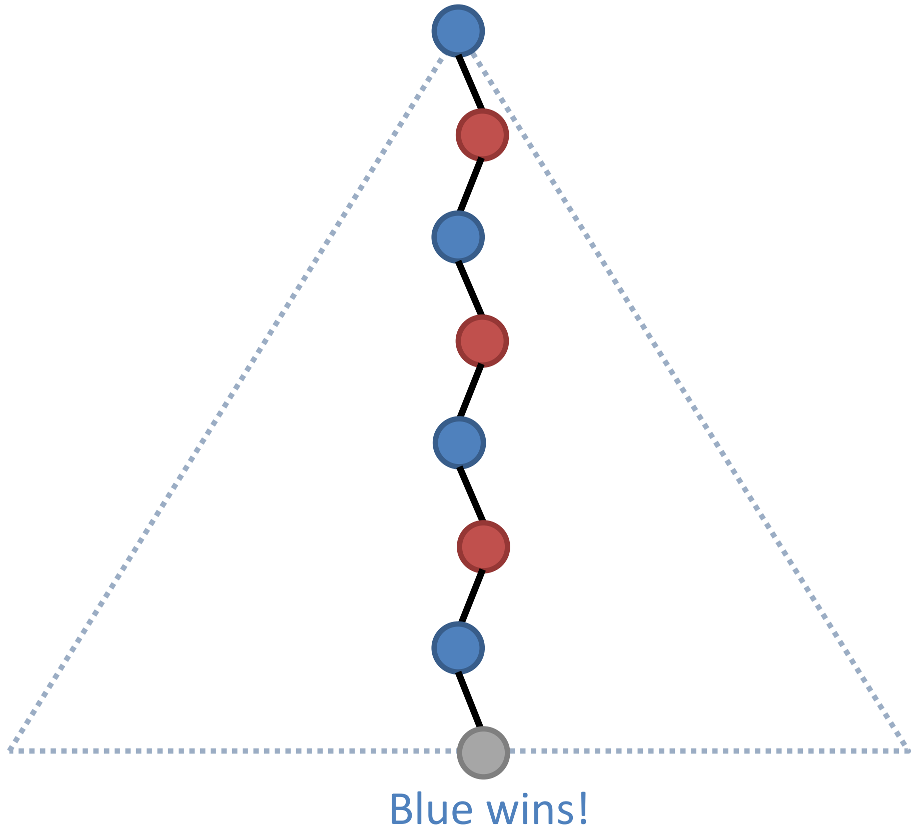
- Whenever an agent acts, generate a **discrete** subgame and solve it
  - Solve using Fictitious Play or CFR
  - Leaf values come from PBS value net
  - Take next action

- Repeat until end of game



ReBeL

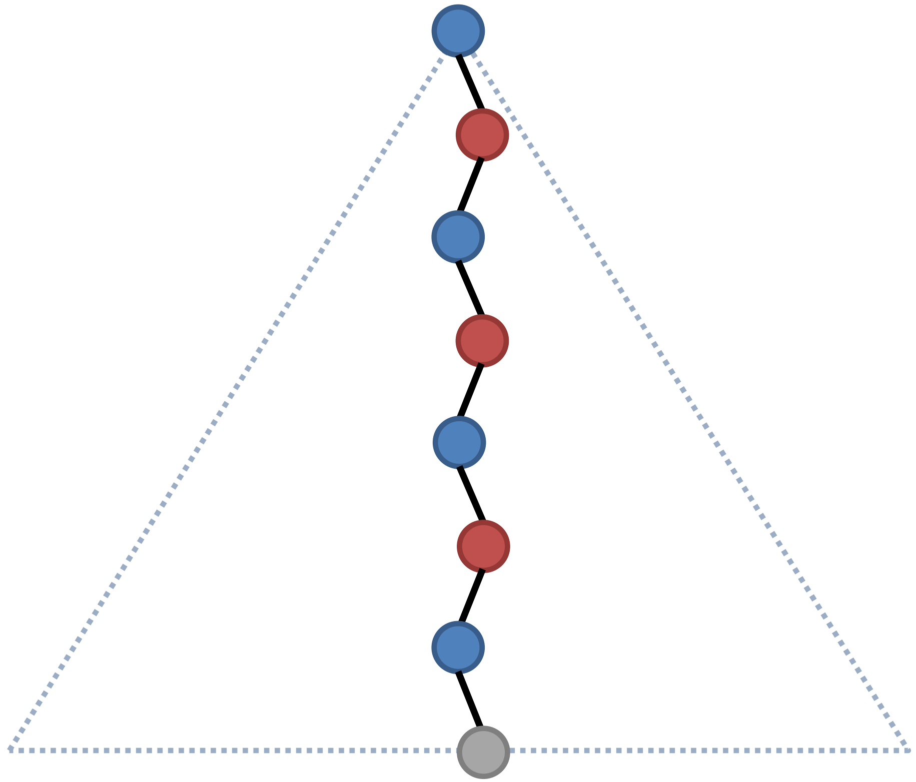
- Whenever an agent acts, generate a **discrete** subgame and solve it
  - Solve using Fictitious Play or CFR
  - Leaf values come from PBS value net
  - Take next action
- Repeat until end of game
- Final value is used as a training example for all encountered PBSs



ReBeL

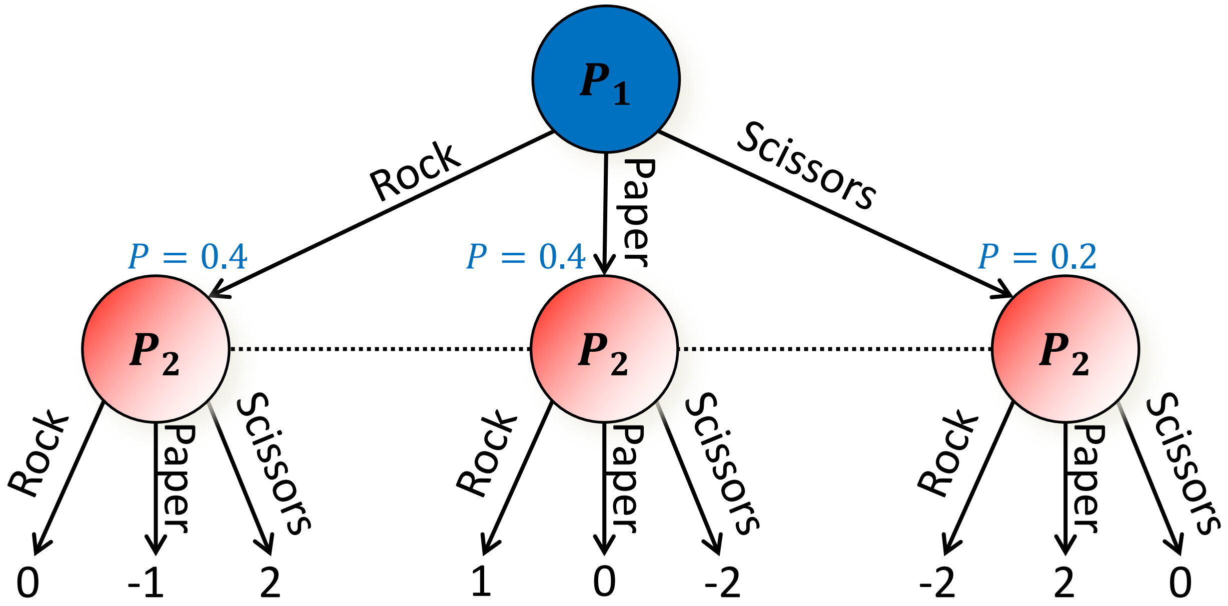
As with AlphaZero, ReBeL chooses a random action with ϵ probability during training to ensure proper exploration

Theorem: With tabular tracking of PBS values, ReBeL will converge to a $\frac{1}{\sqrt{T}}$-Nash equilibrium in finite time, where T is the number of CFR iterations



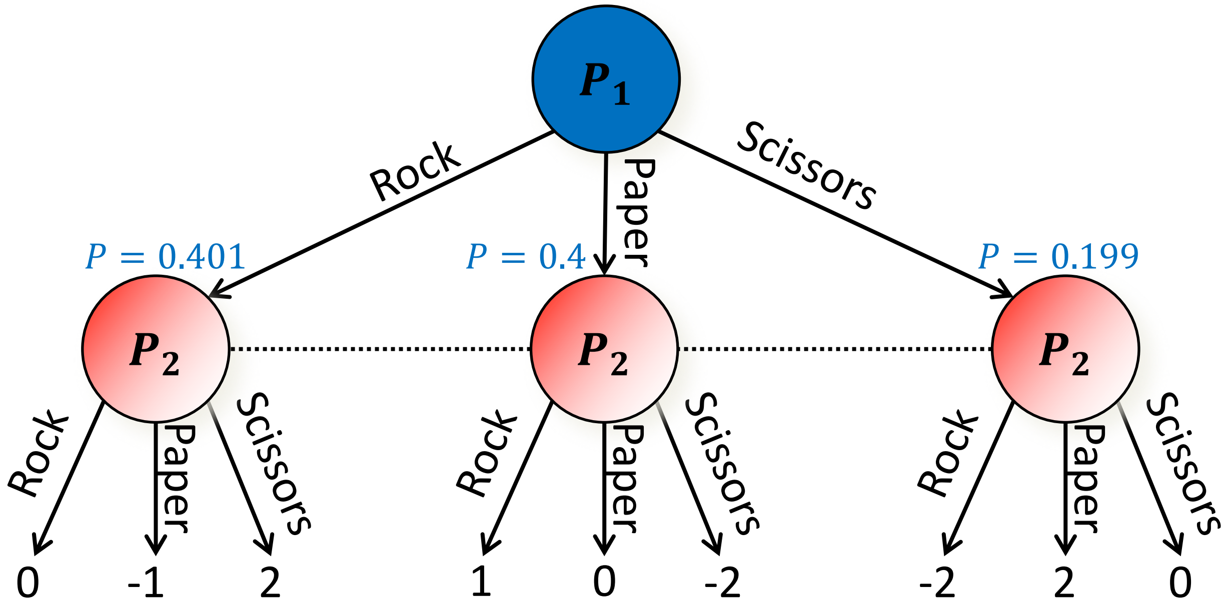
Playing Nash at Test Time

Rock-Paper-Scissors+

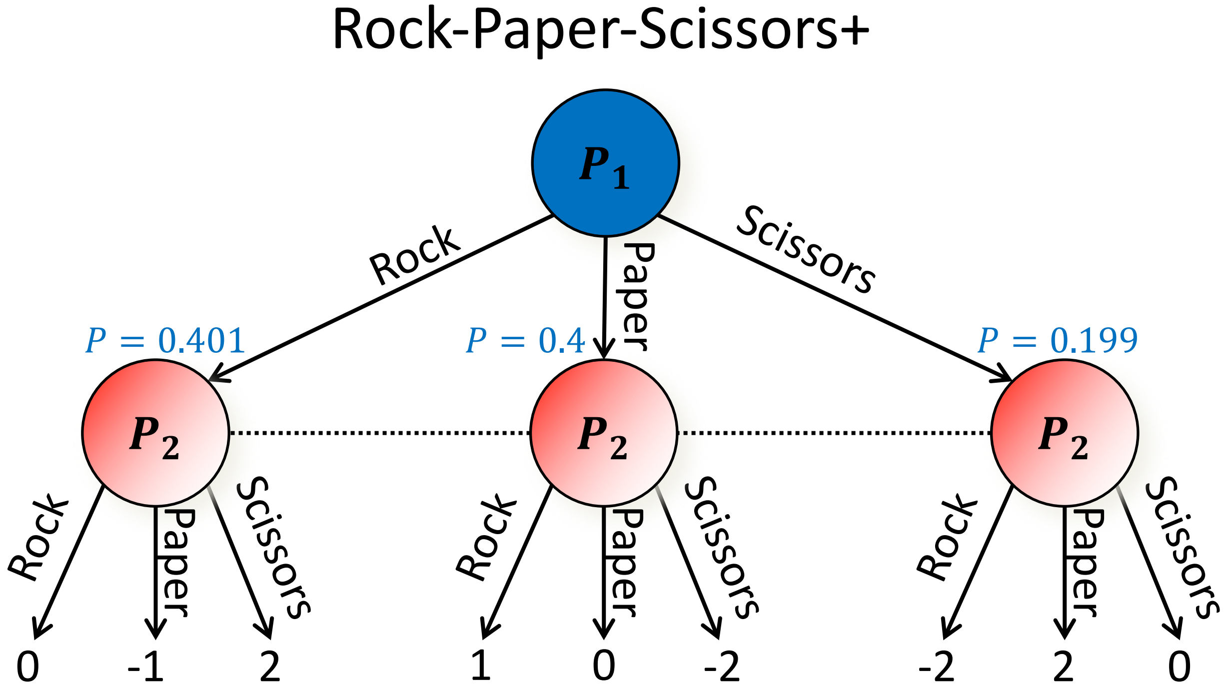


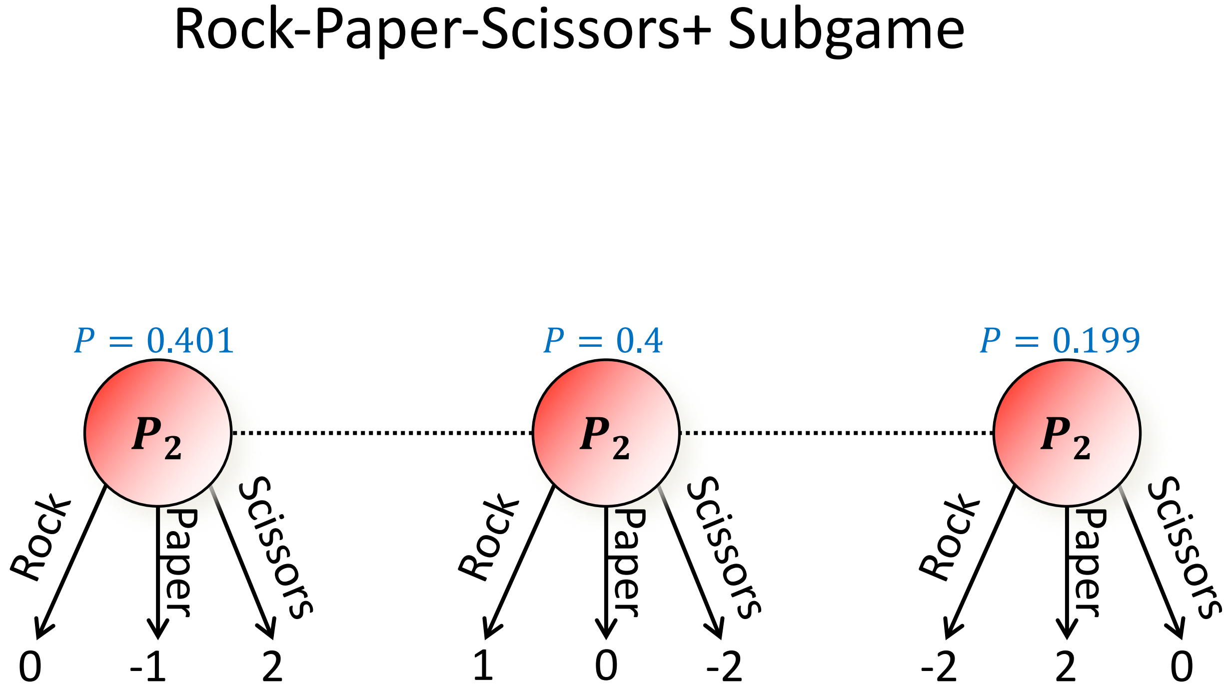
Playing Nash at Test Time

Rock-Paper-Scissors+

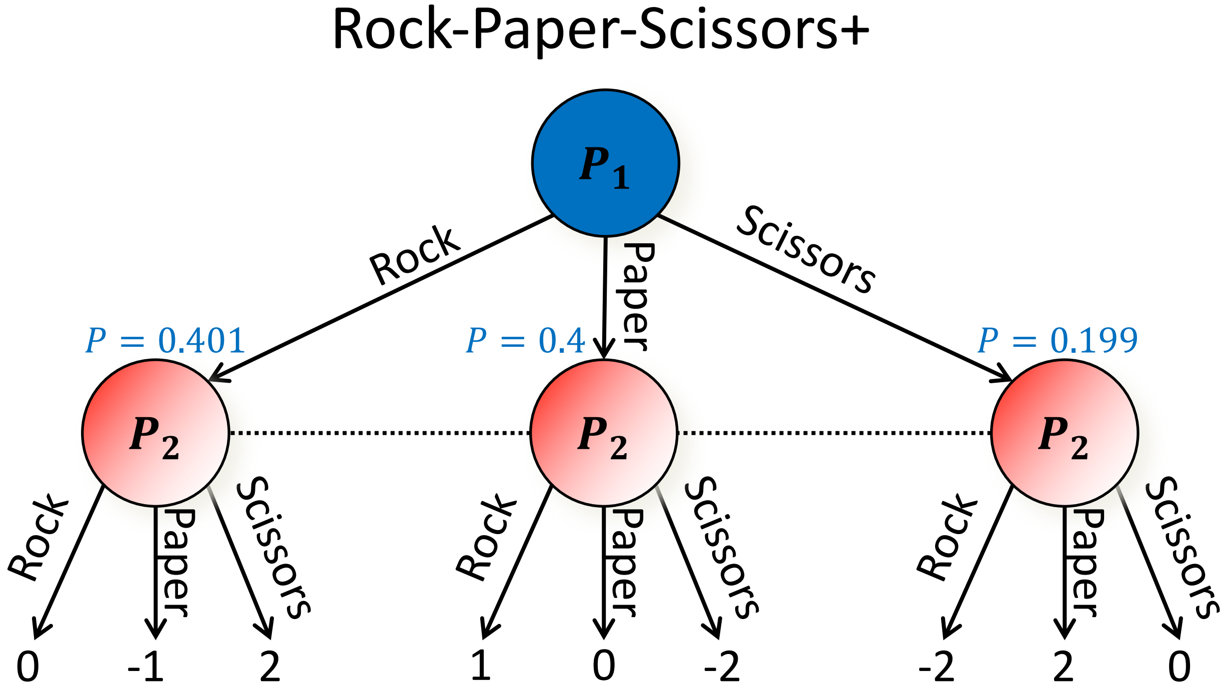


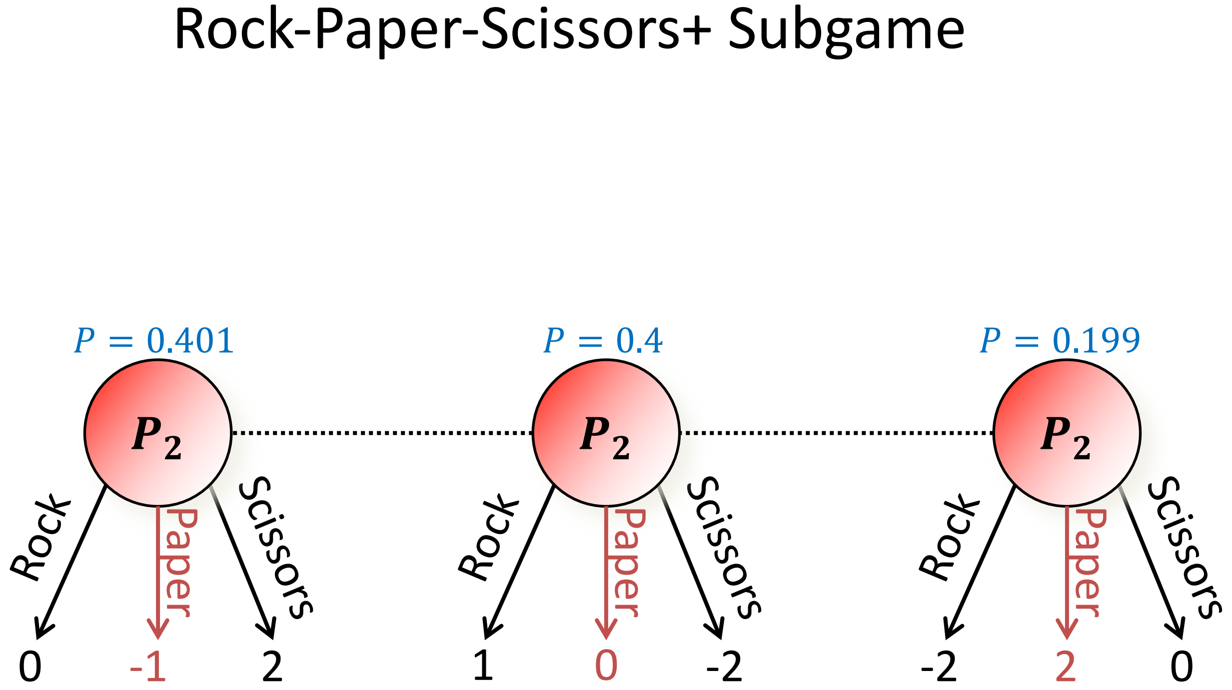
Playing Nash at Test Time



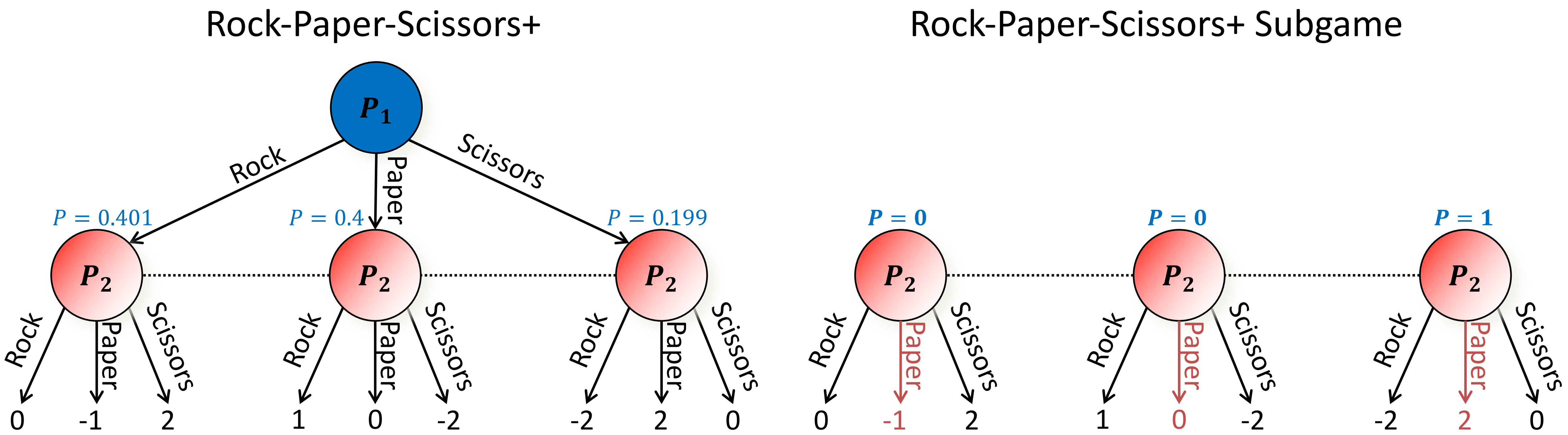


Playing Nash at Test Time





Playing Nash at Test Time



- Our solution: Stop FP / CFR on a **random** iteration and assume beliefs from that iteration
  - Opponent will not know our beliefs, so cannot predict in what way our policy will be pure
  - The subgame policy will be a Nash equilibrium **in expectation**
  - Provably plays according to a Nash equilibrium when using a PBS value function

Results in Two-Player No-Limit Texas Hold'em

	Slumbot	Baby Tartanian8	Local Best Response	Top Humans
DeepStack			383 ± 112	
Libratus		63 ± 14		147 ± 39
Modicum	11 ± 5	6 ± 3		
ReBeL	$\mathbf{45 \pm 5}$	$\mathbf{9 \pm 4}$	$\mathbf{881 \pm 94}$	$\mathbf{165 \pm 69}$

Results in Two-Player Liar's Dice

	1 die, 4 faces	1 die, 5 faces	1 die, 6 faces	2 dice, 3 faces
Tabular Full-Game FP	0.012	0.024	0.039	0.057
Tabular Full-Game CFR	0.001	0.001	0.002	0.002
ReBeL with FP	**0.041**	**0.020**	**0.040**	**0.020**
ReBeL with CFR	**0.017**	**0.015**	**0.024**	**0.017**

Source code available at github.com/facebookresearch/rebel

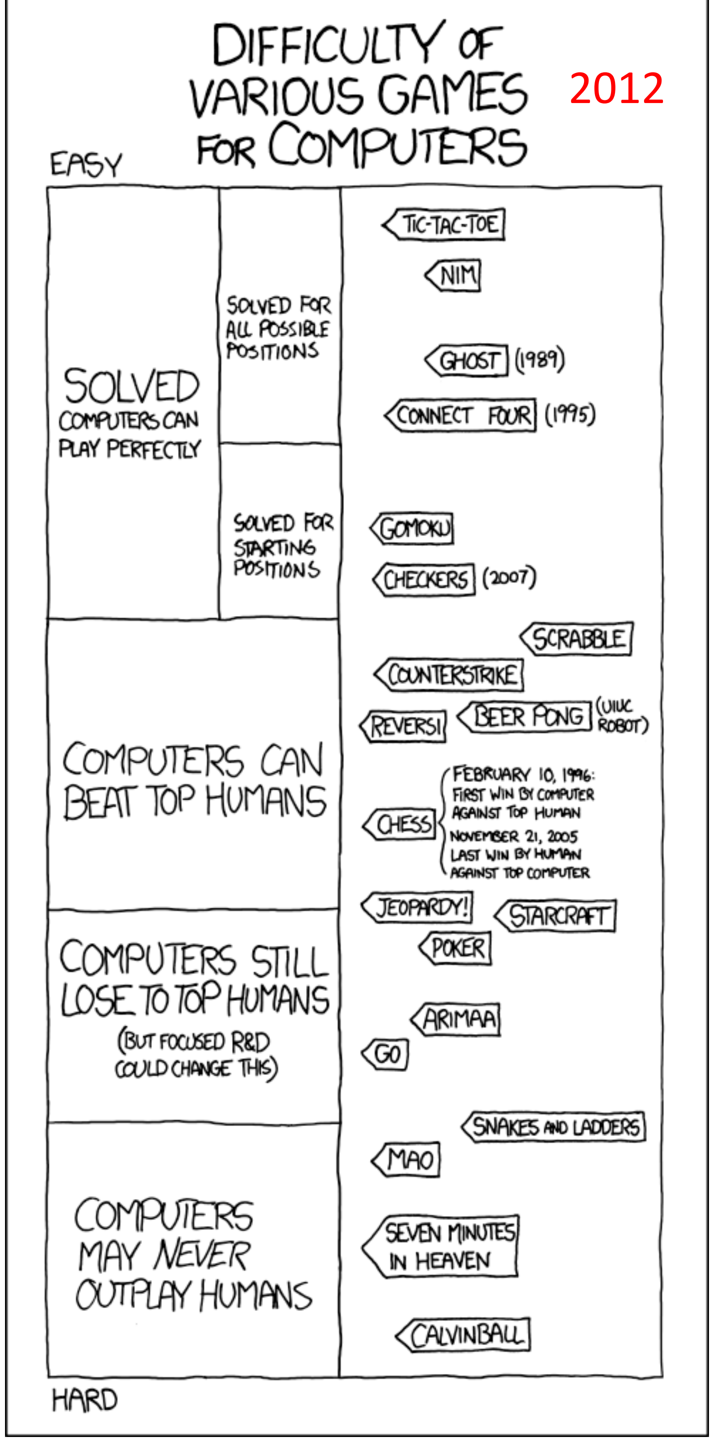
Other thesis topics not covered in this talk

- Improvements to CFR
  - Other forms of pruning
  - Warm starting CFR from arbitrary strategies
- Abstraction Techniques
  - Computing locally optimal discretizations in continuous action spaces
  - Simultaneous abstraction and equilibrium finding
- Search
  - Reach subgame solving and other safe search techniques

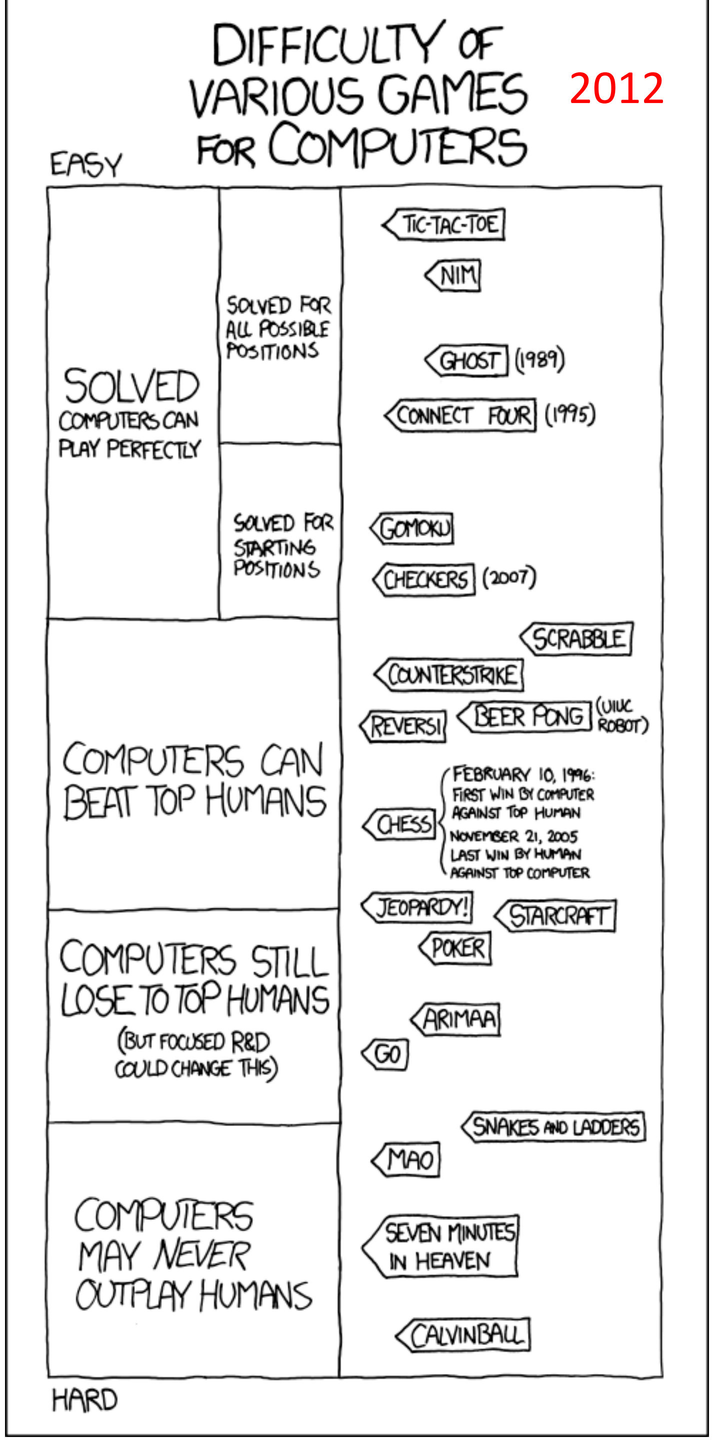
Recap

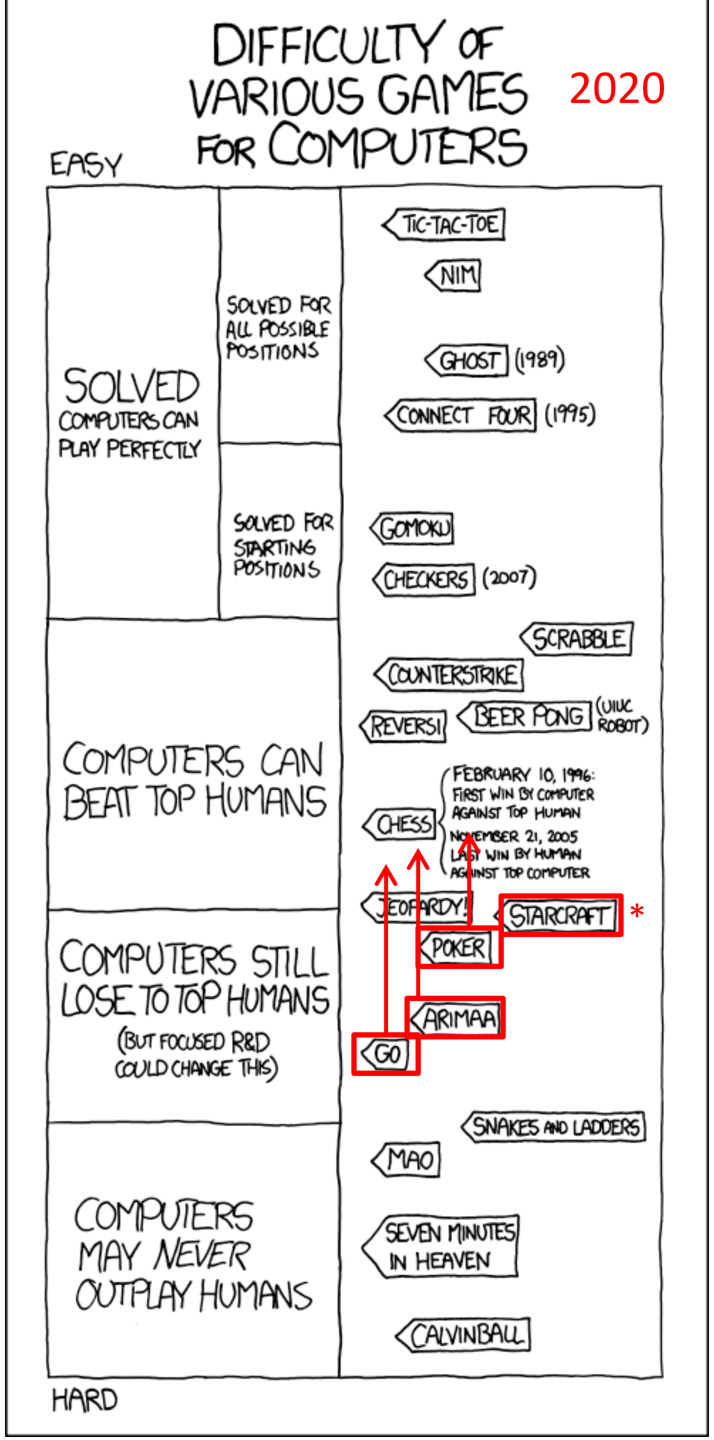
- Developed the state-of-the-art equilibrium-finding algorithm for adversarial imperfect-information games
- Developed the first non-tabular form of CFR to scale to large games
- Developed theoretically sound and scalable search techniques
- Together, these advances enabled an AI to defeat top humans in no-limit poker for the first time

What happens now?


DIFFICULTY OF VARIOUS GAMES FOR COMPUTERS
EASY
SOLVED
COMPUTERS CAN PLAY PERFECTLY
SOLVED FOR ALL POSSIBLE POSITIONS
SOLVED FOR STARTING POSITIONS
COMPUTERS CAN BEAT TOP HUMANS
COMPUTERS STILL LOSE TO TOP HUMANS
(BUT FOCUSED R&D COULD CHANGE THIS)
COMPUTERS MAY NEVER OUTPLAY HUMANS
HARD
TIC-TAC-TOE
NIM
GHOST (1989)
CONNECT FOUR (1995)
GOMOKU
CHECKERS (2007)
SCRABBLE
COUNTERSTRIKE
REVERSI
BEER PONG (UIUC ROBOT)
CHESS
FEBRUARY 10, 1996: FIRST WIN BY COMPUTER AGAINST TOP HUMAN
NOVEMBER 21, 2005 LAST WIN BY HUMAN AGAINST TOP COMPUTER
JEOPARDY!
STARCRAFT
POKER
ARIMAA
GO
SNAKES AND LADDERS
MAO
SEVEN MINUTES IN HEAVEN
CALVINBALL

2012


DIFFICULTY OF VARIOUS GAMES FOR COMPUTERS
2012
EASY
SOLVED
COMPUTERS CAN PLAY PERFECTLY
SOLVED FOR ALL POSSIBLE POSITIONS
TIC-TAC-TOE
NIM
GHOST (1989)
CONNECT FOUR (1995)
SOLVED FOR STARTING POSITIONS
GOMOKU
CHECKERS (2007)
COMPUTERS CAN BEAT TOP HUMANS
SCRABBLE
COUNTERSTRIKE
REVERSI
BEER PONG (UIUC ROBOT)
CHESS
FEBRUARY 10, 1996: FIRST WIN BY COMPUTER AGAINST TOP HUMAN
NOVEMBER 21, 2005 LAST WIN BY HUMAN AGAINST TOP COMPUTER
JEOPARDY!
STARCRAFT
COMPUTERS STILL LOSE TO TOP HUMANS
(BUT FOCUSED R&D COULD CHANGE THIS)
POKER
ARIMAA
GO
COMPUTERS MAY NEVER OUTPLAY HUMANS
SNAKES AND LADDERS
MAO
SEVEN MINUTES IN HEAVEN
CALVINBALL
HARD

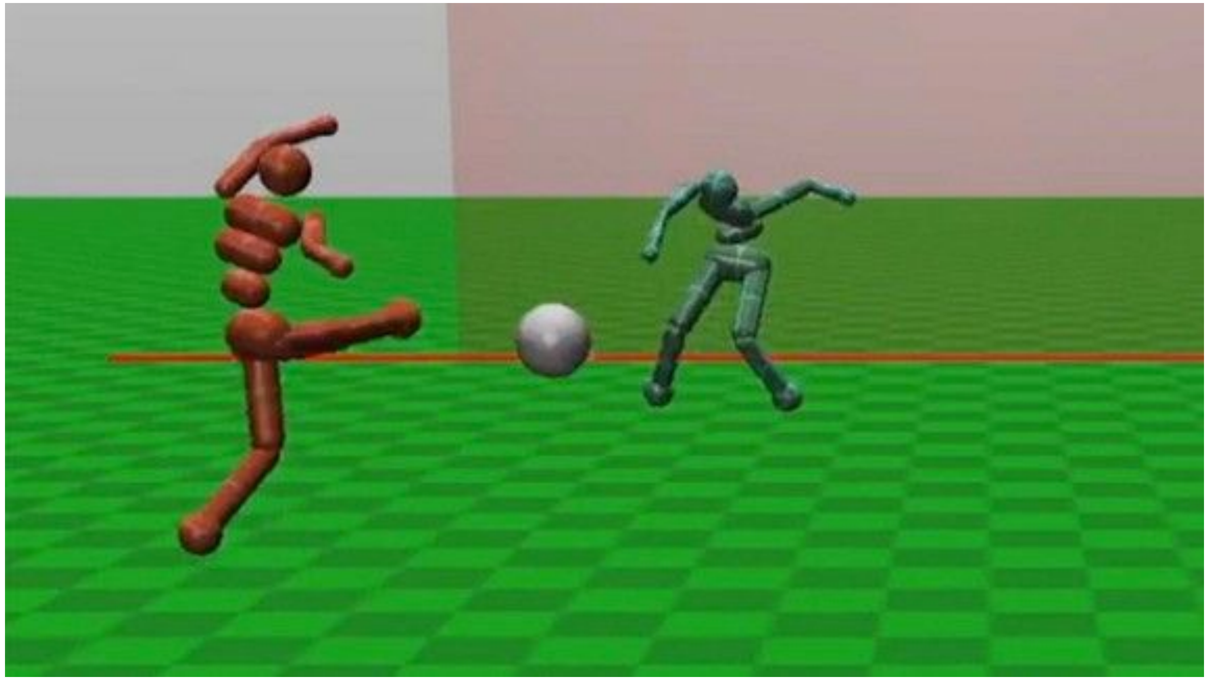

DIFFICULTY OF VARIOUS GAMES FOR COMPUTERS
2020
EASY
SOLVED
COMPUTERS CAN PLAY PERFECTLY
SOLVED FOR ALL POSSIBLE POSITIONS
TIC-TAC-TOE
NIM
GHOST (1989)
CONNECT FOUR (1995)
SOLVED FOR STARTING POSITIONS
GOMOKU
CHECKERS (2007)
COMPUTERS CAN BEAT TOP HUMANS
SCRABBLE
COUNTERSTRIKE
REVERSI
BEER PONG (UIUC ROBOT)
CHESS
FEBRUARY 10, 1996: FIRST WIN BY COMPUTER AGAINST TOP HUMAN
NOVEMBER 21, 2005 LAST WIN BY HUMAN AGAINST TOP COMPUTER
JEOPARDY!
STARCRAFT *
COMPUTERS STILL LOSE TO TOP HUMANS
(BUT FOCUSED R&D COULD CHANGE THIS)
POKER
ARIMAA
GO
COMPUTERS MAY NEVER OUTPLAY HUMANS
SNAKES AND LADDERS
MAO
SEVEN MINUTES IN HEAVEN
CALVINBALL
HARD

Scaling CFR to larger games

- Modern neural network CFR algorithms still discretize action spaces

- Remains to be seen whether CFR scales to 3D environments

- DREAM [Steinberger, Lerer, Brown arXiv-20] is a step in this direction

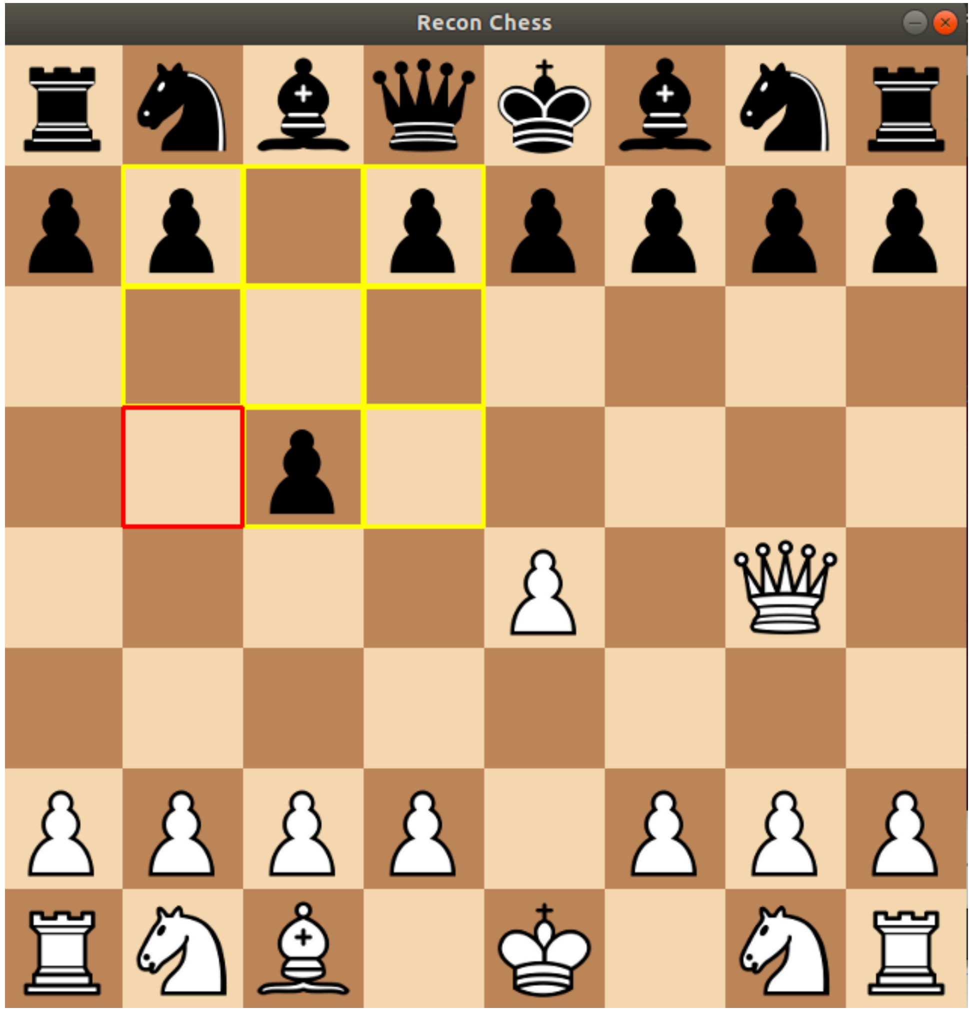


Lack of Common Knowledge

- All of the described search techniques rely on **common knowledge**

- What if there is none?

RECONNAISSANCE
BLIND CHESS



Beyond Two-Player Zero-Sum

- **Life isn't zero sum:** AIs are still bad at cooperation, negotiation, and coalition formation

- Pluribus showed some of these techniques extend beyond two-player zero-sum, but there is more to do

Thank You!

Website: www.noambrown.com

Thesis: http://www.cs.cmu.edu/~noamb/NoamBrownThesis.pdf