Libratus: Superhuman AI for Heads-Up No-Limit Poker

Noam Brown & Tuomas Sandholm
Carnegie Mellon University

July 15, 2025

Motivation

- AI has surpassed humans in perfect-information games: Chess, Go.
- Imperfect-information games are much harder due to hidden knowledge.
- Poker is a key benchmark for real-world strategic decision making.

What is Heads-Up No-Limit Texas Hold'em (HUNL)?

- Two players (Heads-Up), each with 2 private cards.
- 5 shared community cards revealed in 3 stages:
  - Flop: 3 cards
  - Turn: 1 card
  - River: 1 card
- Players make bets with no limit on amount.
- Best 5-card hand at showdown wins the pot.

Game Structure

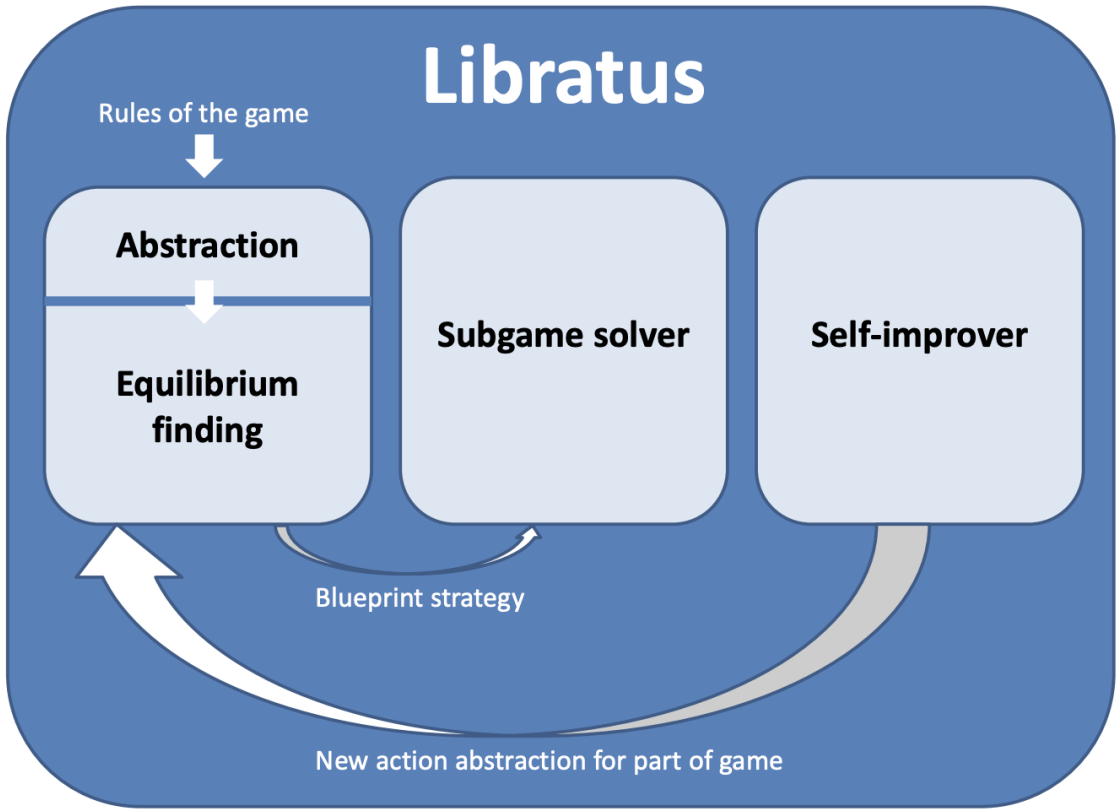
Each hand has four betting rounds:

1. Pre-Flop (after private cards)
2. Flop (3 cards revealed)
3. Turn (4th card)
4. River (5th card)

Actions: Bet, Raise, Call, Fold, Check
It has over 10^{160} decision points.

What is Libratus?

- First AI to defeat top professionals in HUNL poker.
- Domain-independent: no human data or heuristics.
- Composed of three key modules:
  1. Blueprint Strategy
  2. Safe Subgame Solving
  3. Self-Improver



Abstraction

- Same algorithm that used in Tartanian8, but much finer abstraction
  - 1st and 2nd betting round: no abstraction
  - 3rd betting round: 55M card histories $->$ 2.5M buckets
  - 4th betting round: 2.4B card histories $->$ 1.25M buckets
- Abstracting player's actions (bet sizes in poker)
  - Largely based on what top humans and AIs do
  - Added radical bet sizes
  - Optimized some of the bet sizes in the early parts of the tree

Blueprint Strategy

- Uses abstraction to simplify game space:
  - Action abstraction: reduce bet sizes
  - Card abstraction: group similar hands
- Solves abstract game using MCCFR (Monte Carlo Counterfactual Regret Minimization)
- Outputs a strategy used in early rounds

Game-Theoretic Foundations

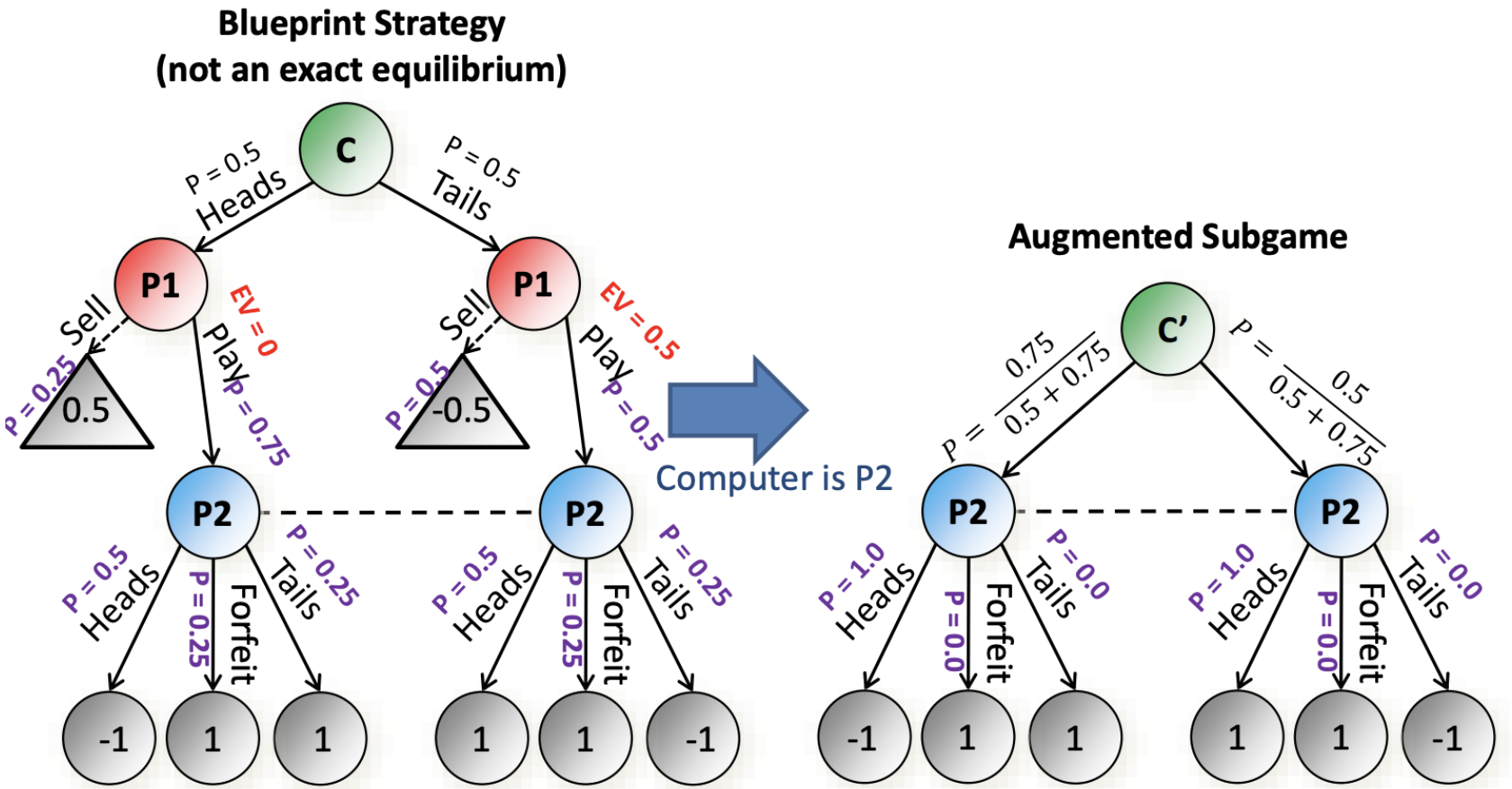
- Poker = 2-player zero-sum extensive-form game
- Objective: Find minimally exploitable strategies (approx. Nash equilibrium)
- Exploitability: how much worse a strategy performs vs. optimal counter-strategy

Safe Subgame Solving

- Real-time solving of subgames with refined abstraction
- Uses an **augmented subgame**: opponent can accept blueprint payoff or play out
- Ensures subgame strategy does not increase exploitability
- The next few slides will illustrate this in more detail

Bayesian subgame solving

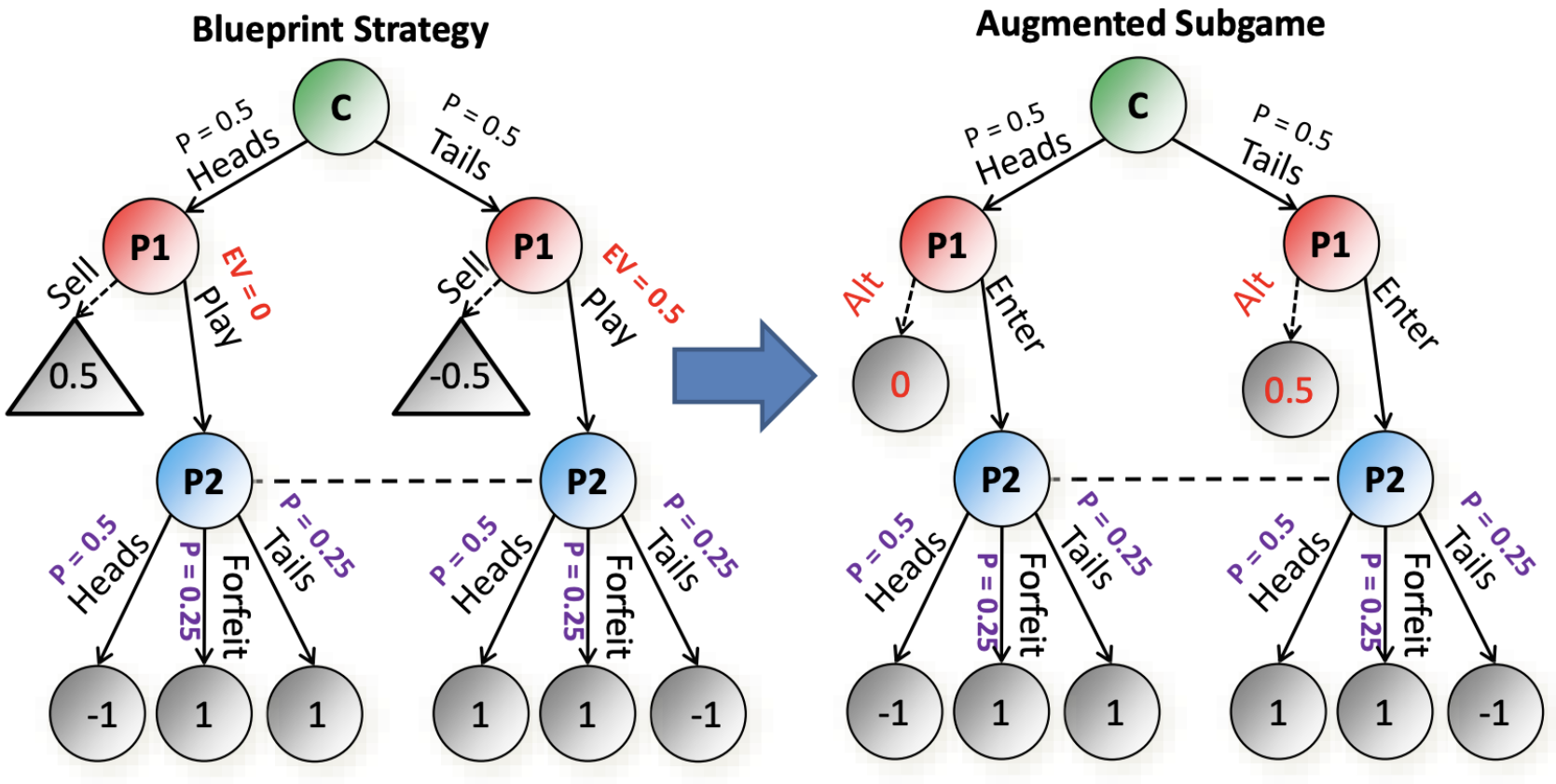
[Gilpin & Sandholm, *AAAI*-06, *AAMAS*-07; Ganzfried & Sandholm, *AAMAS*-15]



- No theoretical guarantees: unsafe
- Does well in practice for some domains

Re-solve refinement [Burch *et al.* AAAI-14]

- P1 can choose between entering the subgame or taking the EV (according to the blueprint) of the subgame
- Makes sure opponent's EV for entering the subgame is no higher than in the blueprint strategy
  => **Safety theorem.** Strategy is no more exploitable than blueprint strategy
- But may miss obvious opportunities for improvement (e.g., not forfeiting)

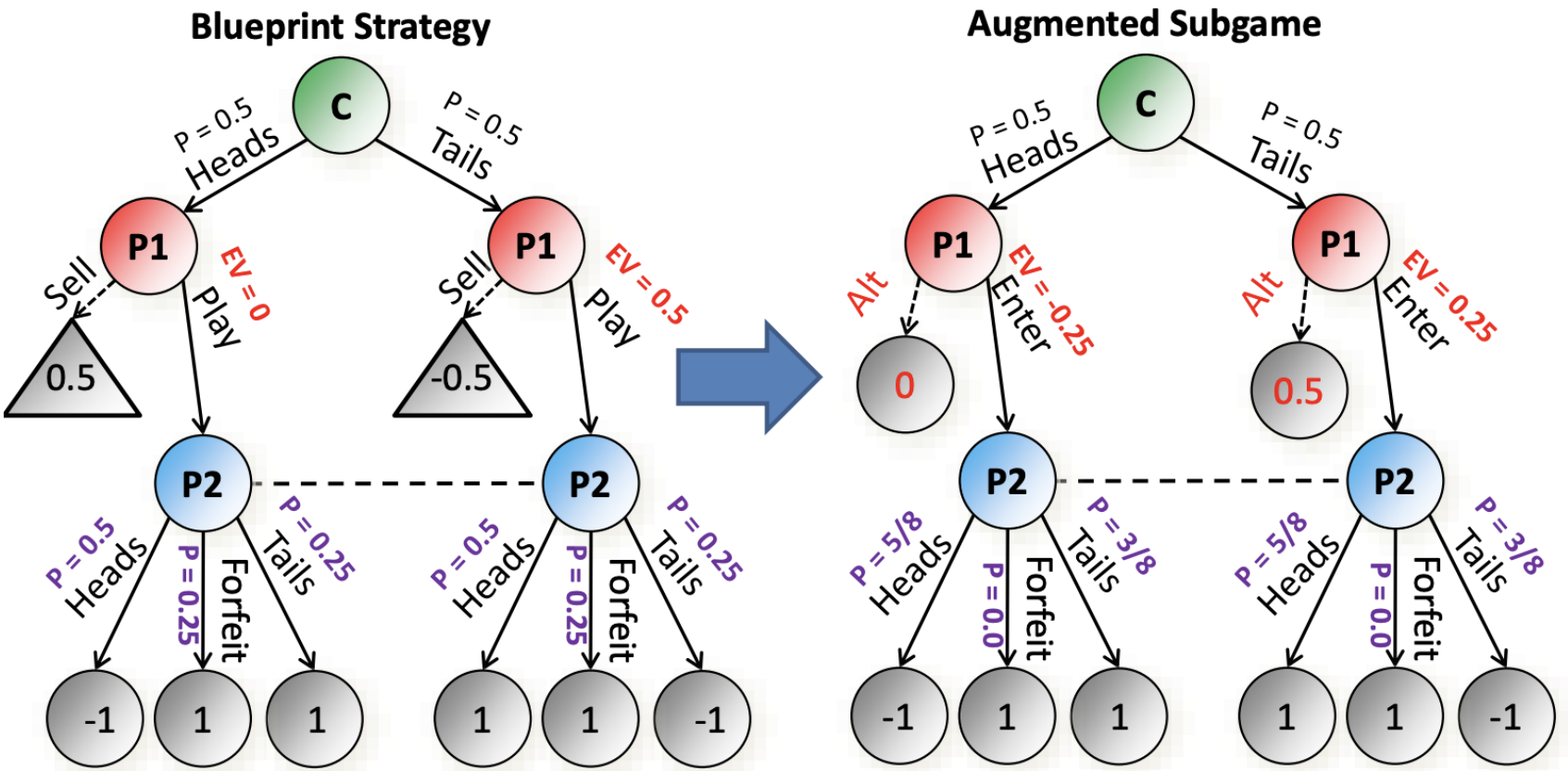


Maxmargin refinement [Moravcik *et al.*, *AAAI*-16]

Similar to Re-solve, but punishes P1 as much as possible for choosing Enter rather than Alt

$Margin_{Heads} = EV[Alt_{Heads}] - EV[Enter_{Heads}]$

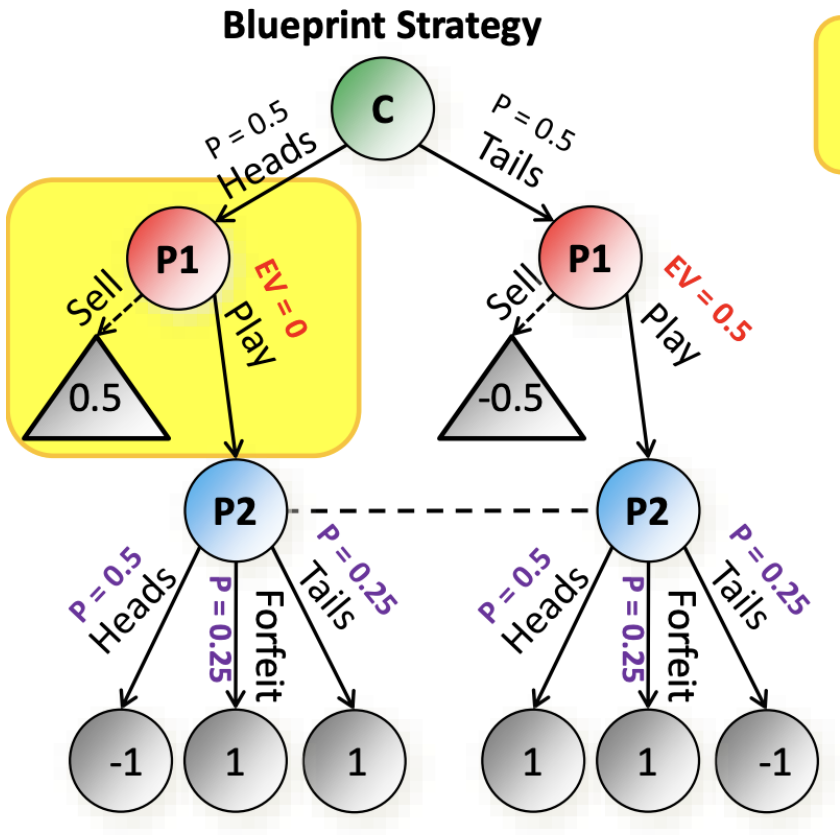
Maximizes the minimum margin (Re-solve simply attempts to make all margins nonnegative)



Problem: While we focus on reducing P1's EV for Heads in the subgame to -0.25, P1 can just Sell for 0.5 in Heads

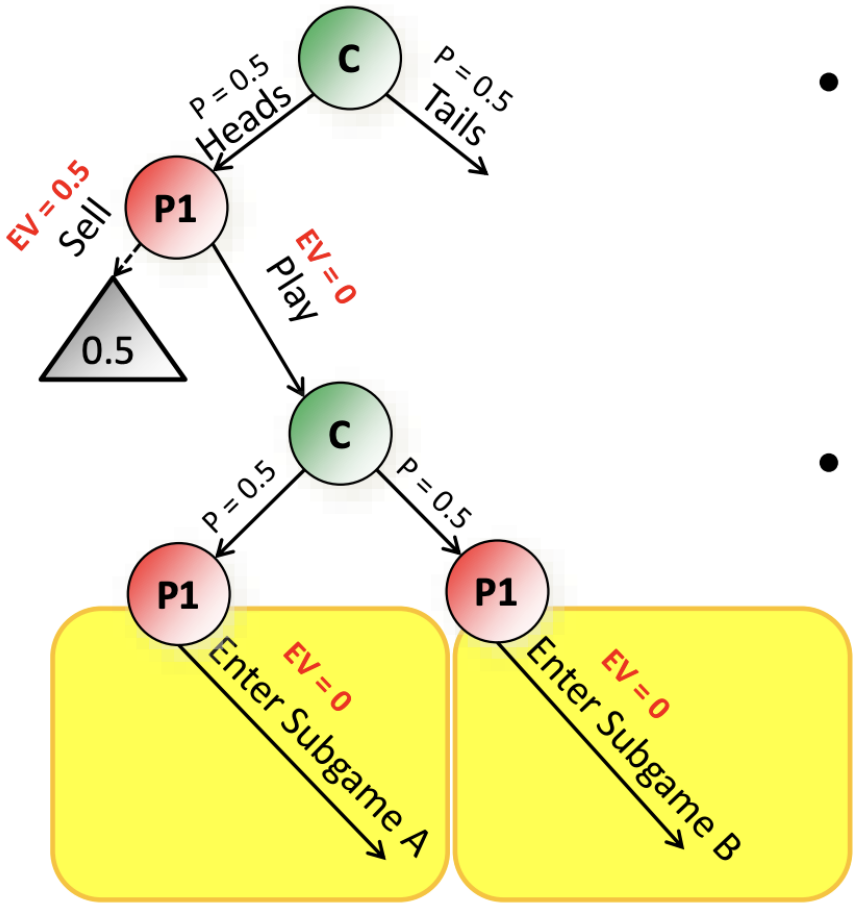
Reach-maxmargin refinement

[Brown & Sandholm, *AAAI*-17 workshop, *NeurIPS*-17, *Science*-18]



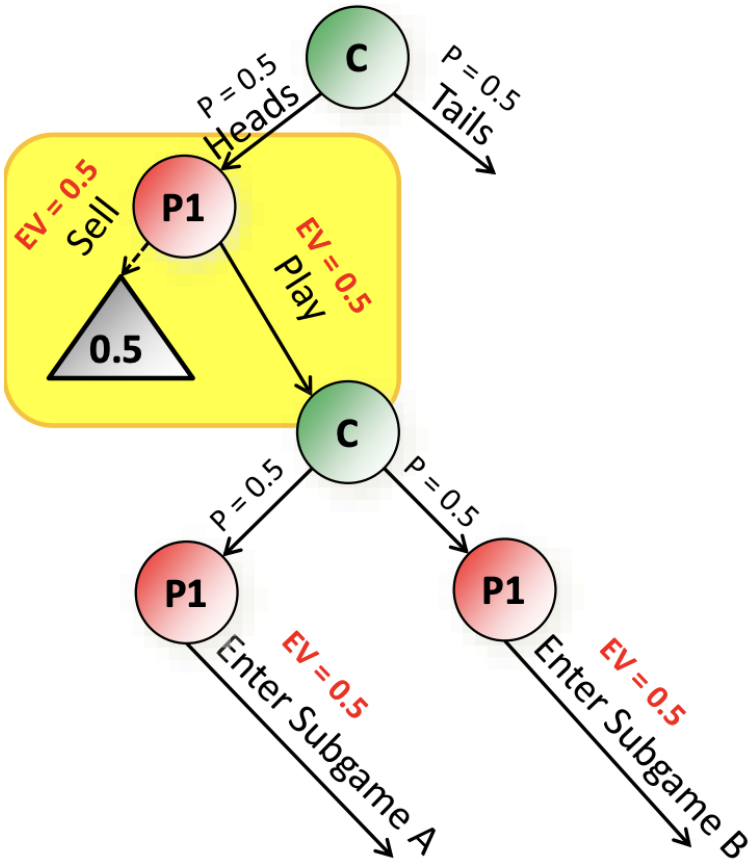
- If P1 chooses Play following Heads, P1 is **gifting** us 0.5
- So, in Augmented Subgame, we can increase the alternative payoff following Heads by 0.5, because choosing Play would still be a mistake for P1 there
- Thus the Gadget Game solver focuses on reducing P1's EV for other types she may have

Reach-maxmargin refinement: multiple subgames



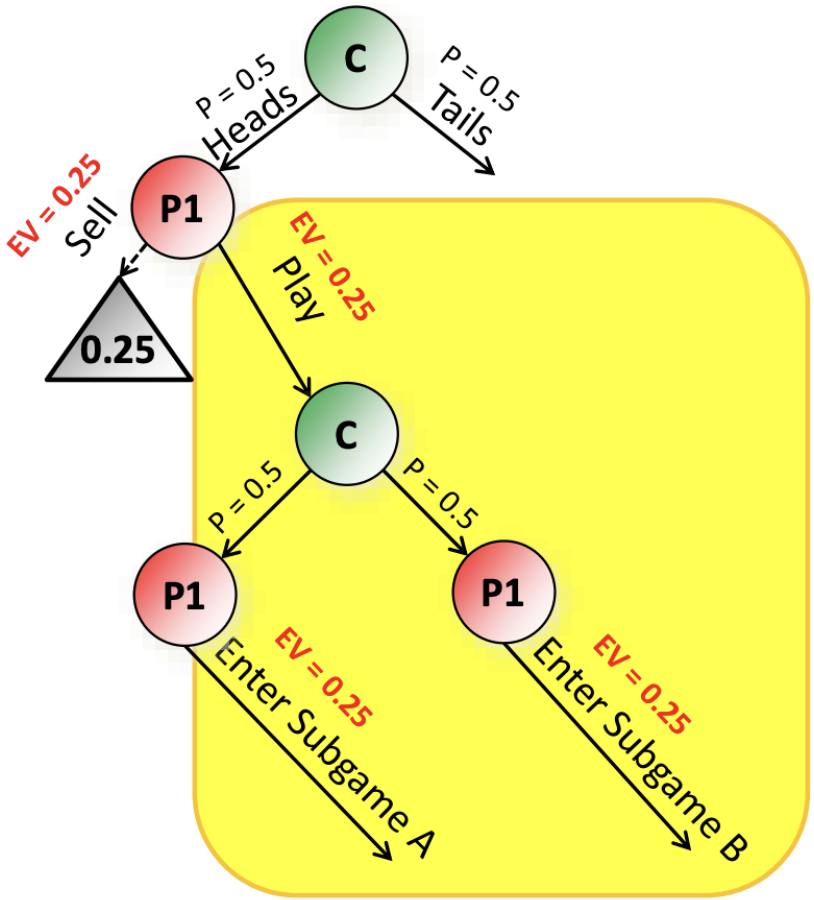
- If multiple subgames are refined, off-path EVs might not remain constant
- Solution: split gifts among subgames by probability subgame is reached

Reach-maxmargin refinement: multiple subgames



- Gifts might not be as large as we thought, because the subgames they come from will be improved
- Solution: substitute a lower bound on the gift

Reach-maxmargin refinement: multiple subgames



- Gifts might not be as large as we thought, because the subgames they come from will be improved
- Solution: substitute a lower bound on the gift

Subgame Solving (Tuomas Sandholm)

Medium-scale experiments on subgame solving within action abstraction

	Small Game Exploitability	Large Game Exploitability
Blueprint Strategy	91.3 mbb / hand	41.4 mbb / hand
Unsafe Subgame Solving	5.51 mbb / hand	397 mbb / hand
Re-solve Refinement	81.2 mbb / hand	36.3 mbb / hand
Maxmargin Refinement	9.36 mbb / hand	6.12 mbb / hand
Reach-Maxmargin Refinement	**8.26 mbb / hand**	**5.50 mbb / hand**

Estimated-Maxmargin Technique Summary

- For each hand h, margin is:

$$\text{Margin}(h) = \text{BlueprintEV}(h) - \text{SubgameEV}(h)$$

- Choose strategy that maximizes the minimum margin
- If EV estimates are $\pm\Delta$ accurate, exploitability increases by at most 2Δ

Nested Subgame Solving

- Off-tree opponent actions (e.g., \$101) trigger new subgame solving
- Unlike rounding used before, it now solves exact response in real time
- Supports recursive nesting: adapt to any bet

Self-Improver Module

- Enhances blueprint strategy during the match
- Detects gaps based on opponents' actions
- Computes game-theoretic fixes overnight
- Non-exploitative: improvements generalize to all opponents

Experimental Results

- Defeated Baby Tartanian8 by +63 mbb/hand
- Beat 4 top professionals in 120,000 hands by +147 mbb/hand
- Win rate was statistically significant (Statistical significance 99.98%, $p = 0.0002$)

Key Contributions

- First superhuman AI in imperfect-information games
- Safe subgame solving with provable bounds
- Practical strategies without relying on opponent modeling