

Computing Strong Game-Theoretic Strategies and Exploiting Suboptimal Opponents in Large Games

Sam Ganzfried

Carnegie Mellon University
Computer Science Department

**Thesis Defense
April 14, 2015**

Scope and applicability of game theory

- Strategic multiagent interactions occur in all fields
 - Economics and business: bidding in auctions, offers in negotiations
 - Political science/law: fair division of resources, e.g., divorce settlements
 - Biology/medicine: robust diabetes management (robustness against “adversarial” selection of parameters in MDP)
 - Computer science: theory, AI, PL, systems; national security (e.g., deploying officers to protect ports), cybersecurity (e.g., determining optimal thresholds against phishing attacks), internet phenomena (e.g., ad auctions)

Game theory background

	rock	paper	scissors
Rock	0,0	-1, 1	1, -1
Paper	1,-1	0, 0	-1,1
Scissors	-1,1	1,-1	0,0

- Players
- Actions (aka pure strategies)
- Strategy profile: e.g., (R,p)
- Utility function: e.g., $u_1(\text{R},\text{p}) = -1$, $u_2(\text{R},\text{p}) = 1$

Zero-sum game

	rock	paper	scissors
Rock	0,0	-1, 1	1, -1
Paper	1,-1	0, 0	-1,1
Scissors	-1,1	1,-1	0,0

- Sum of payoffs is zero at each strategy profile:
e.g., $u_1(\text{R},\text{p}) + u_2(\text{R},\text{p}) = 0$
- Models purely adversarial settings

Mixed strategies

- Probability distributions over pure strategies
- E.g., R with prob. 0.6, P with prob. 0.3, S with prob. 0.1

Best response (aka nemesis)

- Any strategy that maximizes payoff against opponent's strategy
- If P2 plays (0.6, 0.3, 0.1) for r,p,s, then a best response for P1 is to play P with probability 1

Nash equilibrium

- Strategy profile where all players simultaneously play a best response
- Standard solution concept in game theory
 - Guaranteed to always exist in finite games [Nash 1950]
- In Rock-Paper-Scissors, the unique equilibrium is for both players to select each pure strategy with probability $1/3$

Minimax Theorem

- Minimax theorem: For every two-player zero-sum game, there exists a value v^* and a mixed strategy profile σ^* such that:
 - a. P1 guarantees a payoff of at least v^* in the worst case by playing σ^*_1
 - b. P2 guarantees a payoff of at least $-v^*$ in the worst case by playing σ^*_2
- v^* ($= v_1$) is the *value* of the game
- All equilibrium strategies for player i guarantee at least v_i in the worst case
- For RPS, $v^* = 0$

Exploitability

- Exploitability of a strategy is difference between value of the game and performance against a best response
 - Every equilibrium has zero exploitability
- Always playing rock has exploitability 1
 - Best response is to play paper with probability 1

Nash equilibria in two-player zero-sum games

- Zero exploitability – “unbeatable”
- Exchangeable
 - If (a,b) and (c,d) are NE, then (a,d) and (c,b) are too
- Can be computed in polynomial time by a linear programming (LP) formulation

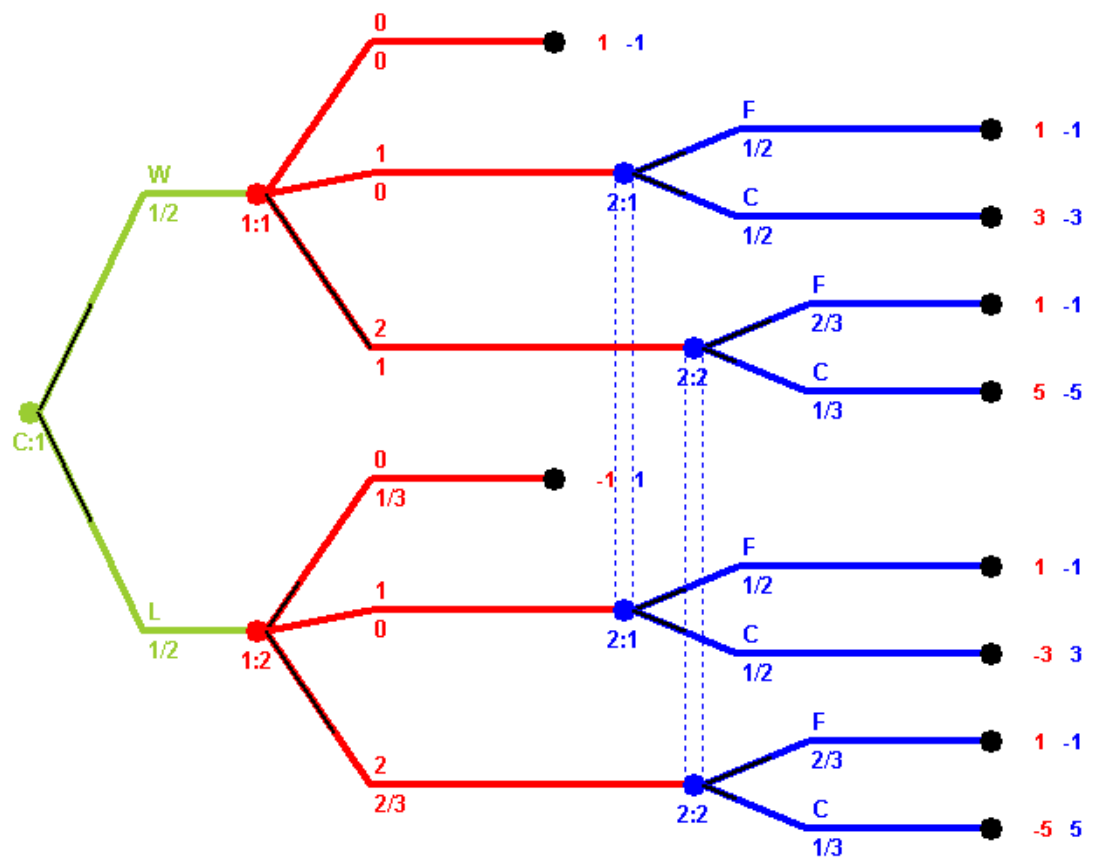
Nash equilibria in multiplayer and non-zero-sum games

- None of the two-player zero-sum results hold
- There can exist multiple equilibria, each with different payoffs to the players
- If one player follows one equilibrium while other players follow a different equilibrium, overall profile is not guaranteed to be an equilibrium
- If one player plays an equilibrium, he could do worse if the opponents deviate from that equilibrium
- Computing an equilibrium is PPAD-hard

Imperfect information

- In many important games, there is information that is private to only some agents and not available to other agents
 - In auctions, each bidder may know his own valuation and only know the distribution from which other agents' valuations are drawn
 - In poker, players may not know private cards held by other players

Extensive-form representation



Extensive-form games

- Two-player zero-sum EFGs can be solved in polynomial time by linear programming
 - Scales to games with up to 10^8 states
- Iterative algorithms (CFR and EGT) have been developed for computing an ϵ -equilibrium that scale to games with 10^{14} states
 - CFR also applies to multiplayer and general sum games, though no significant guarantees in those classes
 - (MC)CFR is self-play algorithm that samples actions down tree and updates regrets and average strategies stored at every information set

Leading paradigm for solving large imperfect-information games

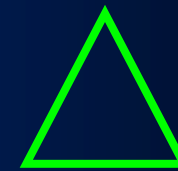
Original game



Automated abstraction



Abstracted game



Custom
equilibrium-finding
algorithm



Nash equilibrium

Reverse mapping



Nash equilibrium

Texas hold 'em poker

- Huge game of imperfect information
 - Most studied imp-info game in AI community since 2006 due to AAAI computer poker competition
 - Most attention on 2-player variants (2-player zero-sum)
 - Multi-billion dollar industry
- Limit Texas hold 'em – fixed betting size
 - $\sim 10^{17}$ nodes in game tree
- No limit Texas hold 'em – unlimited bet size
 - $\sim 10^{165}$ nodes in game tree
 - Most active domain in last several years
 - Most popular variant for humans

No-limit Texas hold 'em poker

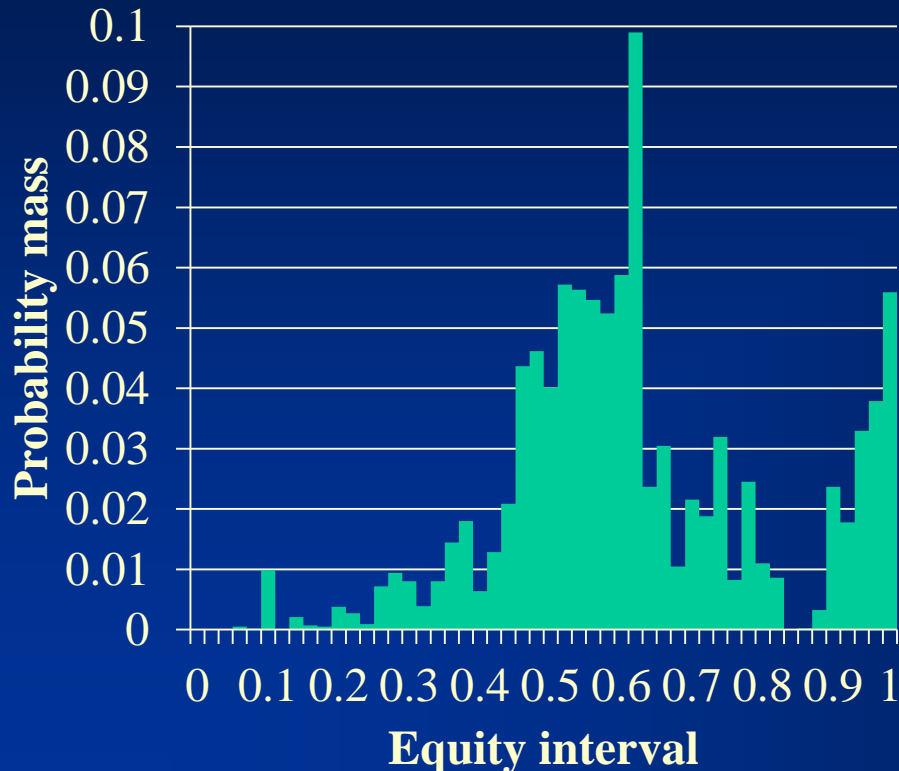
- Two players have stack and pay blinds (ante)
- Each player dealt two private cards
- Round of betting (preflop)
 - Players can fold, call, bet (any amount up to stack)
- Three public cards dealt (flop) and a second round of betting
- One more public card and round of betting (turn)
- Final card and round of betting (river)
- Showdown

Game abstraction

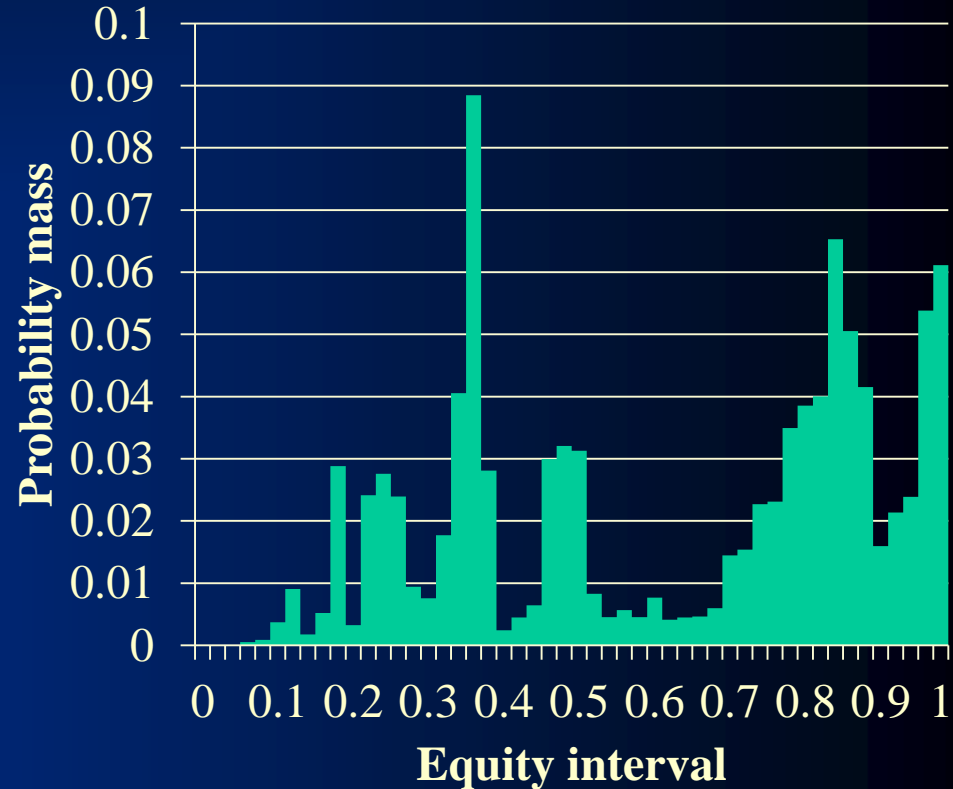
- Necessary for solving large games
 - 2-player no-limit Texas hold 'em has 10^{165} game states, while best solvers “only” scale to games with 10^{14} states
- Information abstraction: grouping information sets together
- Action abstraction: discretizing action space
 - E.g., limit bids to be multiples of \$10 or \$100

Information abstraction

Equity distribution for
6c6d. EHS: 0.634



Equity distribution for
KcQc. EHS: 0.633



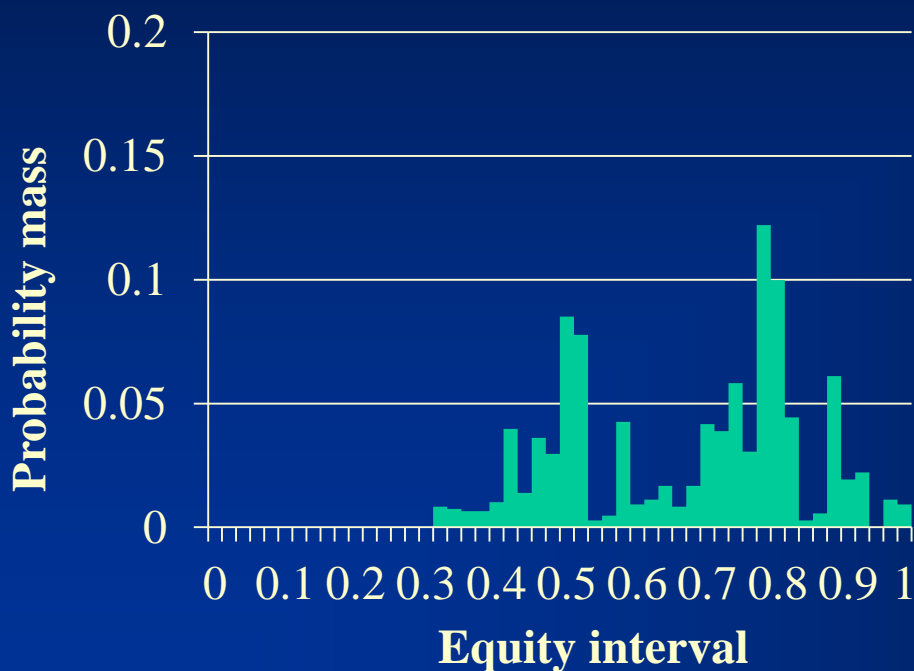
Human abstraction



Potential-aware abstraction with EMD

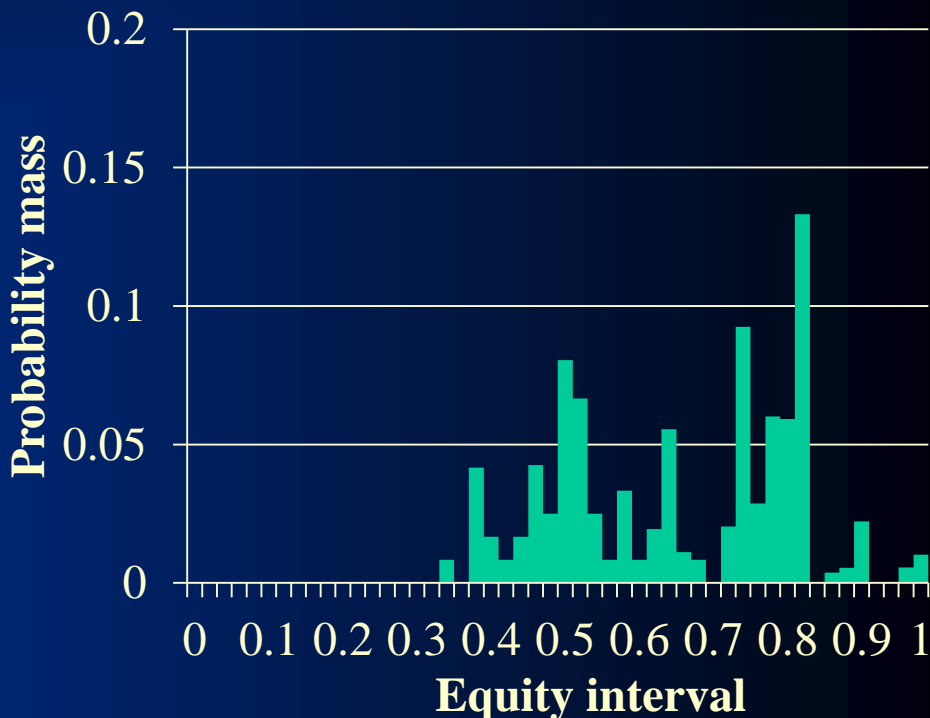
Equity distribution for
TcQd-7h9hQh on river
(final round)

EHS: 0.683



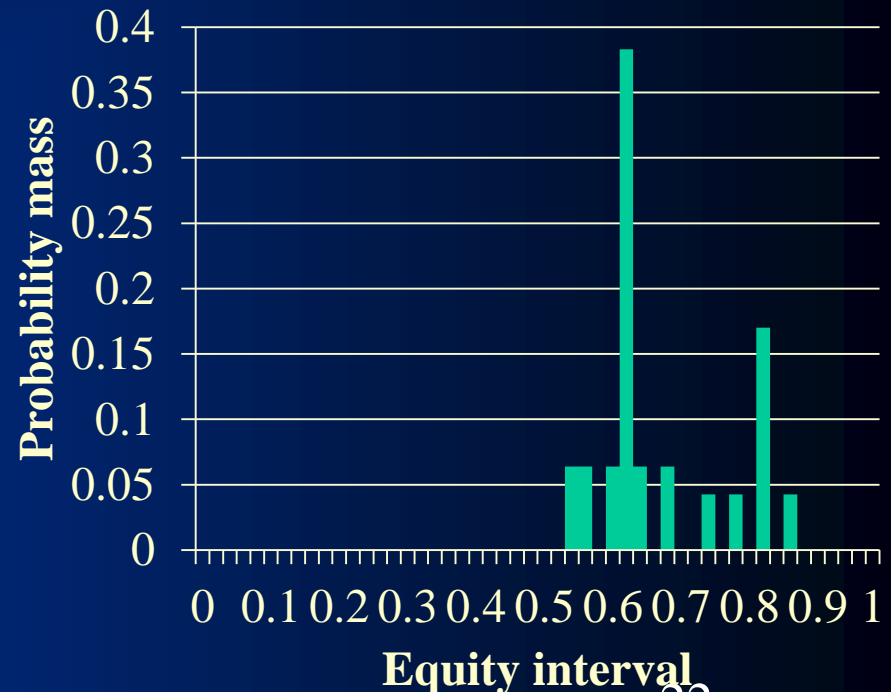
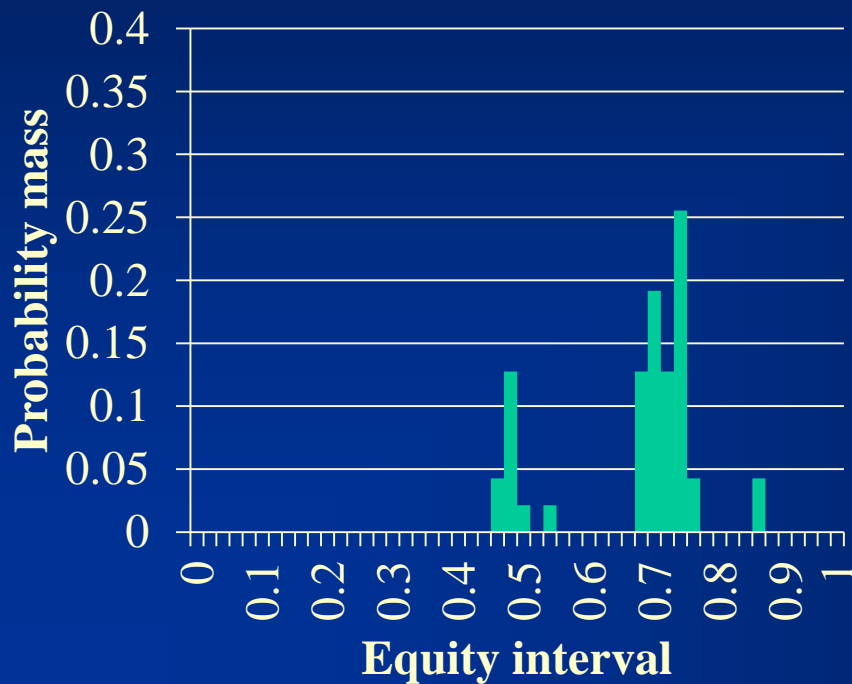
Equity distribution for
5c9d-3d5d7d on river
(final round)

EHS: 0.679



Potential-aware abstraction with EMD

- Equity distributions on the turn. Each point is EHS for given turn card assuming uniform random river and opponent hand
- EMD is 4.519 (vs. 0.559 using comparable units to river EMD)



Algorithm for potential-aware imperfect-recall abstraction with EMD

- Bottom-up pass of the information tree (assume an abstraction for final rounds has already been computed using arbitrary approach)
- For each round n
 - Let m_i^{n+1} denote mean of cluster i in A^{n+1}
 - For each pair of round $n+1$ clusters (i,j) , compute distance $d_{i,j}^n$ between m_i^{n+1} and m_j^{n+1} using d^{n+1}
 - For each point x^n , create histogram over clusters from A^{n+1}
 - Compute abstraction A^n using EMD with $d_{i,j}^n$ as ground distance function
 - Developed fast custom heuristic for approximating EMD in our multidimensional setting
 - Best commercially-available algorithm was far too slow to compute abstractions in poker

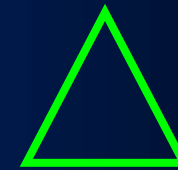
Leading paradigm for solving large extensive-form games

Original game

Abstracted game



Automated abstraction



Custom
equilibrium-finding
algorithm



Nash equilibrium

Reverse mapping



Nash equilibrium

Hierarchical abstraction to enable distributed equilibrium computation

- On distributed architectures and supercomputers with high inter-blade memory access latency, straightforward MCCFR parallelization approaches lead to impractically slow runtimes
 - When a core does an update at an information set it needs to read and write memory with high latency
 - Different cores working on same information set may need to lock memory, wait for each other, possibly over-write each others' parallel work, and work on out-of-sync inputs
- Our approach solves the former problem and also helps mitigate the latter issue

High-level approach

- To obtain these benefits, our algorithm creates an information abstraction that allows us to assign disjoint components of the game tree to different blades so the trajectory of each sample only accesses information sets located on the same blade.
 - First cluster public information at some early point in the game (public flop cards in poker), then cluster private information separately for each public cluster.
- Run modified version of external-sampling MCCFR
 - Samples one pair of preflop hands per iteration. For the later betting rounds, each blade samples public cards from its public cluster and performs MCCFR within each cluster.

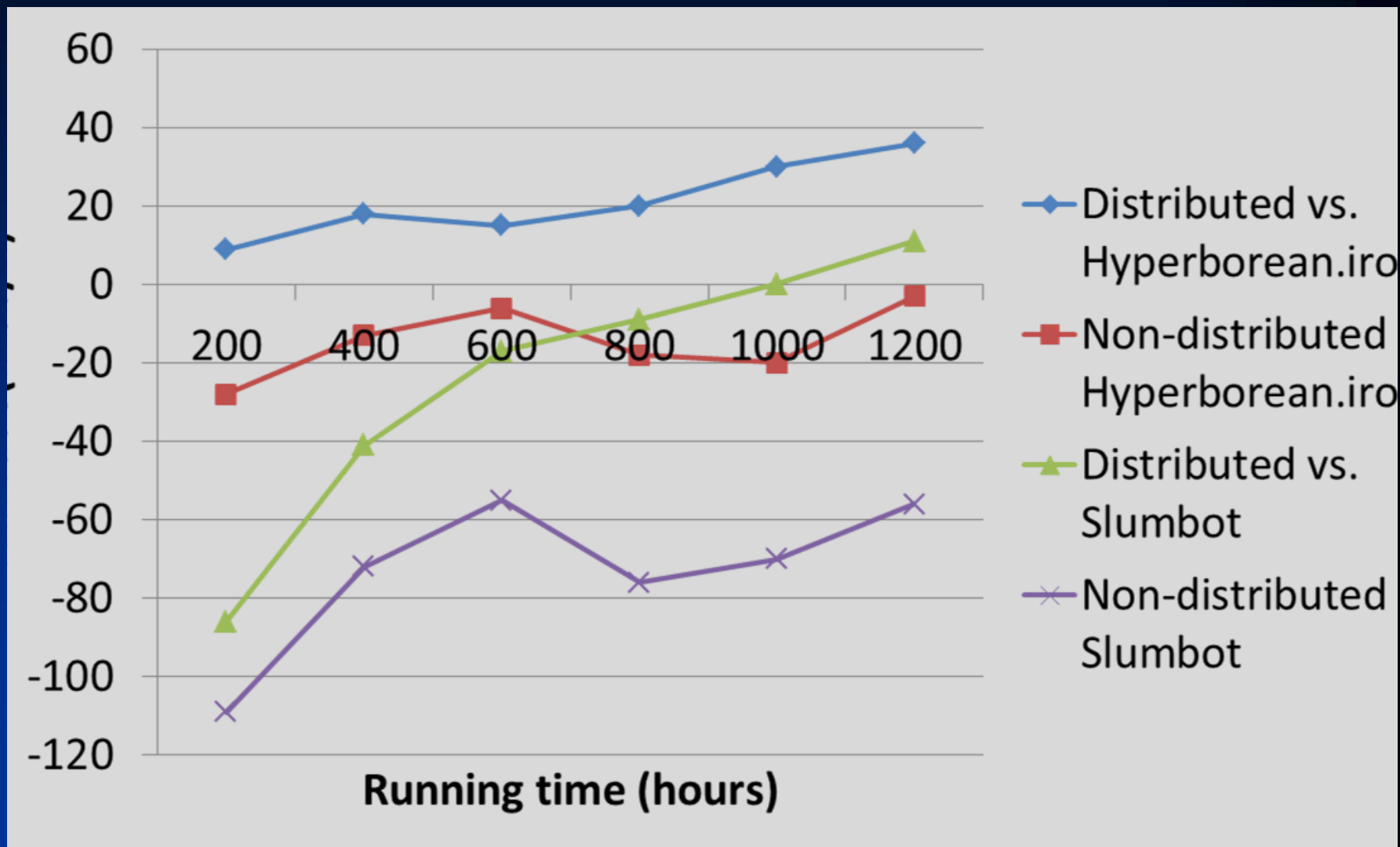
Hierarchical abstraction algorithm for distributed equilibrium computation

- For $r = 1$ to r^*-1 , cluster states at round r using A_r
 - A_r is arbitrary abstraction algorithm
 - E.g., for preflop round in poker
- Cluster public states at round r^* into C buckets
 - E.g., flop round in poker
- For $r = r^*$ to R , $c = 1$ to C , cluster states at round r that have public information states in public bucket c into B_r buckets using abstraction algorithm A_r

Algorithm for computing public information abstraction

- Construct transition table T
 - $T[p][b]$ stores how often public state p will lead to bucket b of the *base abstraction* A , aggregated over all possible states of private information.
- for $i = 1$ to $M-1$, $j = i+1$ to M (M is # of public states)
 - $s_{i,j} := 0$
 - for $b = 1$ to B
 - $s_{i,j} += \min(T[i][b], T[j][b])$
 - $d_{i,j} = (V - s_{i,j})/V$
- Cluster public states into C clusters using (custom) clustering algorithm L with distance function d
 - $d_{i,j}$ corresponds to fraction of private states not mapped to same bucket of A when paired with public info i and j

Comparison to non-distributed approach



Tartanian7: champion two-player no-limit Texas hold 'em agent

- Beat every opponent with statistical significance in 2014 AAAI computer poker competition

SartreNLExp	Nyx	Hyperborean.iro	Slumbot	Prelude	HibiscusBiscuit	PijaiBot	Feste.iro	LittleRock	KEmpfer	Rembrant3	HTSZ_CS_14	Lucifer
261 ± 47	121 ± 38	21 ± 16	33 ± 16	20 ± 16	125 ± 44	499 ± 68	141 ± 45	214 ± 57	516 ± 61	980 ± 34	1474 ± 180	1819 ± 111

Table 1: Win rate (in mbb/h) of our agent in the 2014 AAAI Annual Computer Poker Competition against opposing agents.

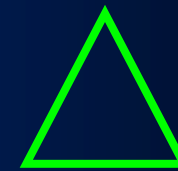
Leading paradigm for solving large extensive-form games

Original game

Abstracted game



Automated abstraction



Custom equilibrium-finding algorithm



Nash equilibrium

Reverse mapping



Nash equilibrium

Reverse mapping

- **Action translation** mapping interprets opponents' actions that have been omitted from action abstraction
 - Natural approaches perform very poorly
 - Developed new approach that has theoretical justification, outperforms prior approaches on several domains, satisfies natural axioms, adopted by most strong poker agents
- Further **post-processing** approaches
 - Also important even if we do not perform any action abstraction

Purification and thresholding

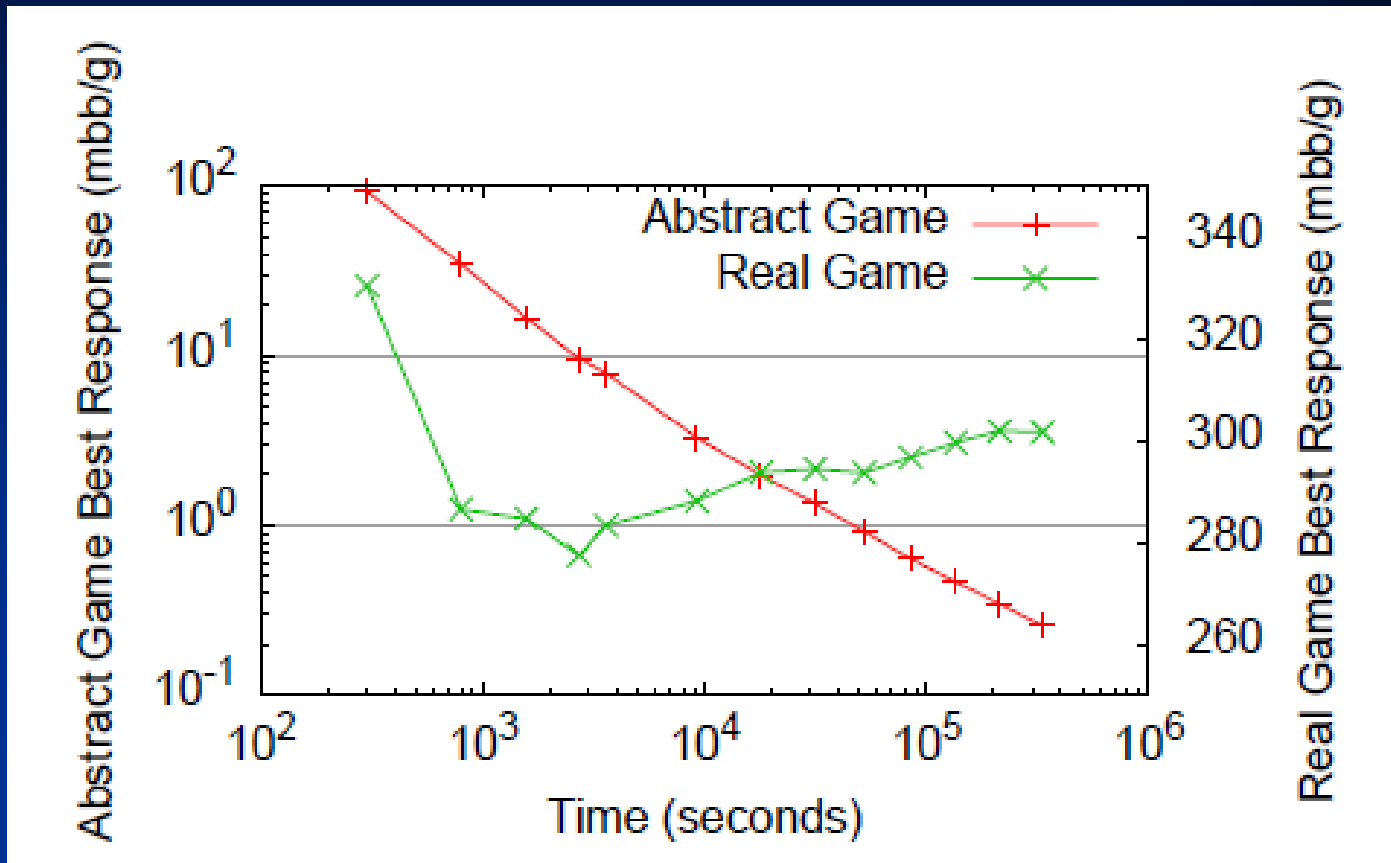
- *Thresholding*: round action probabilities below c down to 0 (then renormalize)
- *Purification* is extreme case where we play maximal-probability action with probability 1

Benefits of post-processing techniques

- 1) Failure of equilibrium-finding algorithm to fully converge
 - Tartanian4 had exploitability of 800 mbb/hand even within its abstraction (always folding has exploitability of 750 mbb/hand!)

Benefits of post-processing techniques

- 2) Combat overfitting of equilibrium to the abstraction



Experiments on no-limit Texas hold 'em

- Purification outperforms using a threshold of 0.15
 - Does better than it against all but one 2010 competitor, beats it head-to-head, and won bankroll competition

Purification and thresholding

- 4x4 two-player zero-sum matrix games with payoffs uniformly at random from $[-1,1]$
- Compute equilibrium F in full game
- Compute equilibrium A in abstracted game that omits last row and column
 - essentially “random” abstractions
- Compare $u_1(A_1, F_2)$ to $u_1(\text{pur}(A_1), F_2)$
- **Conclusion: Abstraction+purification outperforms just abstraction (against full equilibrium) at 95% confidence level**

Purification and thresholding

Purified average payoff	-0.050987 +- 0.00042
Unpurified average payoff	-0.054905 +- 0.00044
# games where purification led to improved performance	261569 (17.44 %)
# games where purification led to worse performance	172164 (11.48%)
# games where purification led to no change in performance	1066267 (71.08 %)

- Some conditions when they perform identically:
 1. The abstract equilibrium A is a pure strategy profile
 2. The support of A_1 is a subset of the support of F_1

Purification and thresholding

- Results depend crucially on the support of the full equilibrium
- If we only consider the set of games that have an equilibrium σ with a given support, purification improves performance for each class except for the following, where the performance is statistically indistinguishable:
 - σ is the pure strategy profile in which each player plays his fourth pure strategy
 - σ is a mixed strategy profile in which player 1's support contains his fourth pure strategy, and player 2's support does not contain his fourth pure strategy

New family of post-processing techniques

- 2 main ideas:
 - Bundle similar actions
 - Add preference for conservative actions
- First separate actions into {fold, call, “bet”}
 - If probability of folding exceeds a threshold parameter, fold with prob. 1
 - Else, follow purification between fold, call, and “meta-action” of “bet.”
 - If “bet” is selected, then follow purification within the specific bet actions.
- Many variations: threshold parameter, bucketing of actions, thresholding value among buckets, etc. 40

Post-processing experiments

	Hyperborean.iro	Slumbot	Average	Min
No Thresholding	+30 ± 32	+10 ± 27	+20	+10
Purification	+55 ± 27	+19 ± 22	+37	+19
Thresholding-0.15	+35 ± 30	+19 ± 25	+27	+19
New-0.2	+39 ± 26	+103 ± 21	+71	+39

Computing NE in games with more than two players

- Developed new algorithms for computing ε -equilibrium strategies in multiplayer imperfect-information stochastic games
 - Models multiplayer poker tournament endgames
- Most successful algorithm, called PI-FP, used a two-level iterative procedure
 - Outer loop is variant of policy iteration
 - Inner loop is an extension of fictitious play
- Proposition: If the sequence of strategies determined by iterations of PI-FP converges, then the final strategy profile is an equilibrium.
- We verified that our algorithms did in fact converge to ε -equilibrium strategies for very small ε

New game-solving paradigms

Incorporating qualitative models

Player 1's strategy Player 2's strategy

Weaker hand



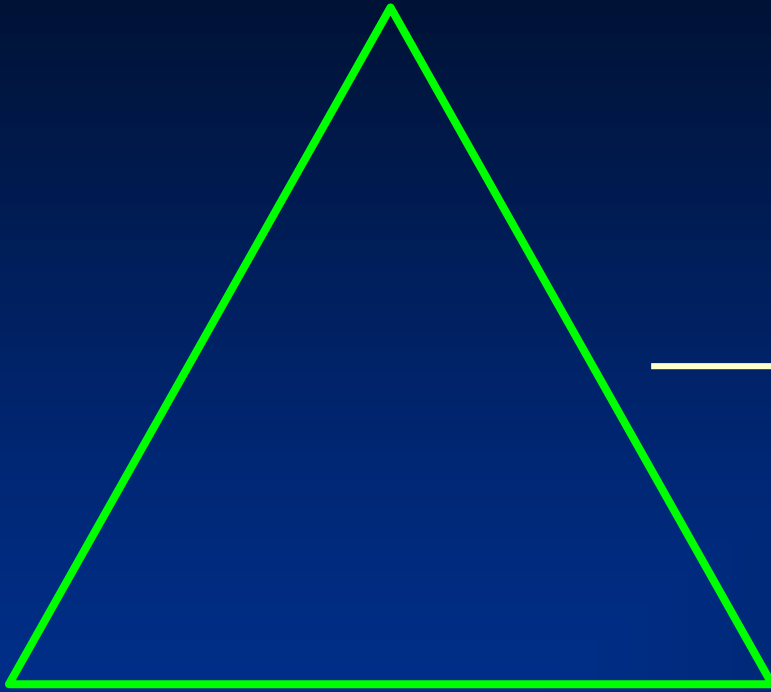
Stronger hand

BLUFF-FOLD	FOLD/BLUFF
CHECK-FOLD	FOLD/CHECK
	BLUFF/CHECK
CHECK-CALL	CALL/CHECK
BET-FOLD	CALL/BET
	RAISE/BET

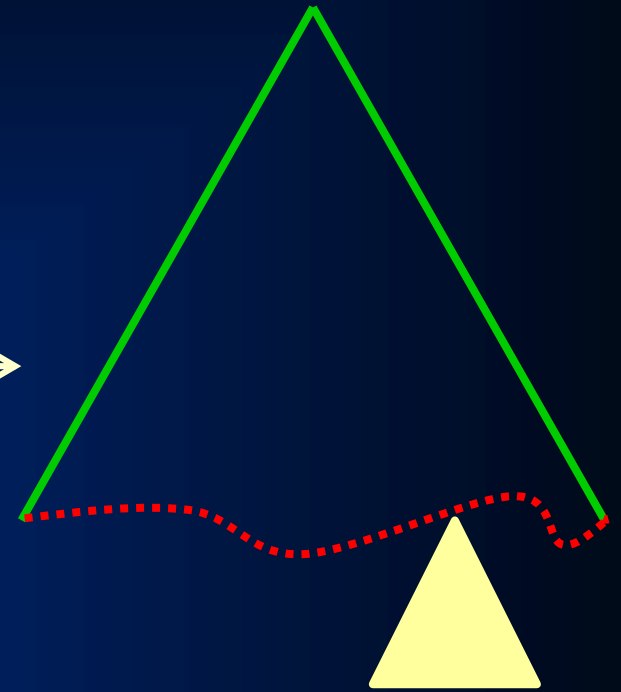
BLUFF-FOLD	FOLD/BLUFF
CHECK-FOLD	FOLD/CHECK
	BLUFF/CHECK
CHECK-CALL	CALL/CHECK
BET-FOLD	CALL/BET
	RAISE/BET

	BLUFF
CHECK-FOLD	
	CHECK
CHECK-CALL	
	BET

Endgame solving



Strategies for entire game
computed offline



Endgame strategies
computed in real time to
greater degree of accuracy

Endgame definition

- E is an **endgame** of a game G if:
 1. Set of E 's nodes is a subset of set of G 's nodes
 2. If s' is a child of s in G and s is a node in E , then s' is also a node in E
 3. If s is in the same information set as s' in G and s is a node in E , then s' is also a node in E

Can endgame solving guarantee equilibrium?

- Suppose that we computed an exact (full-game) equilibrium in the initial portion of the game tree prior to the endgame (the **trunk**), and computed an exact equilibrium in the endgame. Is the combined strategy an equilibrium of the full game?

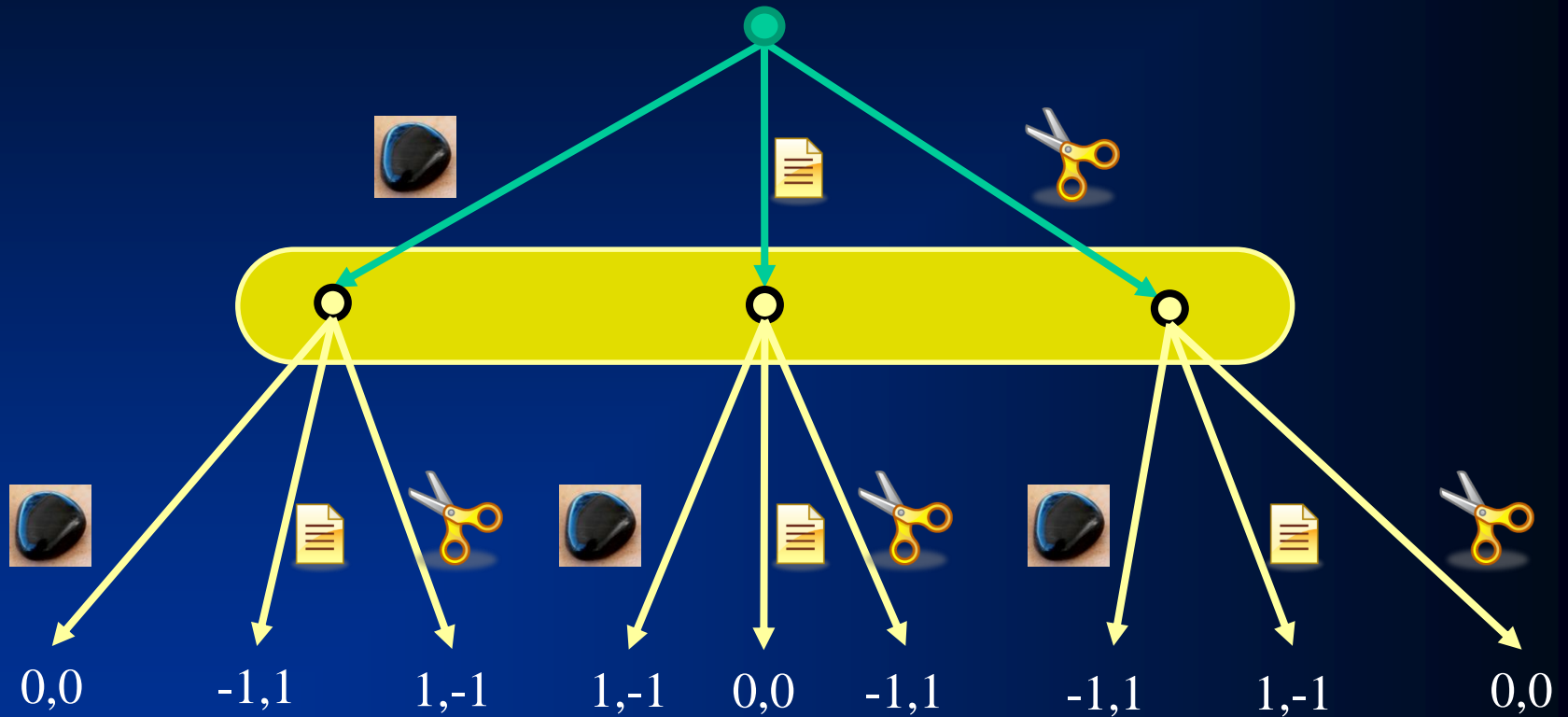
Can endgame solving guarantee equilibrium?

- No!
- Several possible reasons this may fail:
 - The game may have many equilibria, and we might choose one for the trunk that does not match up correctly with the one for the endgame
 - We may compute equilibria in different endgames that do not balance appropriately with each other

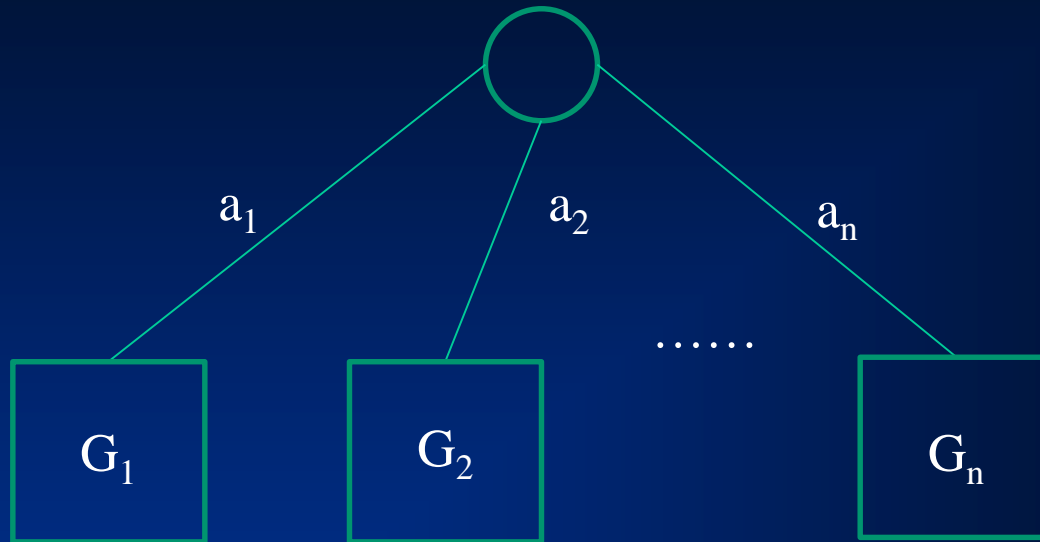
Can endgame solving guarantee equilibrium?

Proposition: There exist games with a unique equilibrium and a single endgame for which endgame solving can produce a non-equilibrium strategy profile in the full game

Limitation of endgame solving



Is there any hope?



Player 1 selects his action a_i , then the players play imperfect-information game G_i .

Is there any hope?

- Endgame solving produces strategies with low exploitability in games where the endgame is a significant strategic portion of the full game.
 - i.e., in games where any endgame strategy with high full-game exploitability can be exploited by the opponent by modifying his strategy just within the endgame.

Is there any hope?

- Proposition: If every strategy that has exploitability strictly more than ε in the full game has exploitability of strictly more than δ within the endgame, then the strategy output by a solver that computes a δ -equilibrium in the endgame induced by a trunk strategy t would constitute an ε -equilibrium of the full game when paired with t .

Endgame property

- We can classify different games according to property described by premise of proposition
 - If premise is satisfied, then we can say game satisfies the (ϵ, δ) -endgame property
- Interesting quantity would be smallest value $\epsilon^*(\delta)$ such that game satisfies the (ϵ, δ) -endgame property for a given δ .
 - Game above has $\epsilon^*(\delta) = \delta$ for each $\delta \geq 0$
 - RPS has $\epsilon^*(\delta) = 1$ for each $\delta \geq 0$

Benefits of endgame solving

- Computation of exact (rather than approximate) equilibrium strategies in the endgames
- Computation of equilibrium refinements (e.g., undominated and ε -quasi-perfect equilibrium)
- Better abstractions in the endgame that is reached
 - Finer-grained abstractions
 - History-aware abstractions
 - Strategy-biased abstractions
- Solving the “off-tree problem”

Efficient algorithm for endgame solving in large imperfect-information games

- Naïve approach requires $O(n^2)$ lookups to the strategy table, where n is the number of possible hands
 - Computationally infeasible (> 1 min/hand)
- Our algorithm uses just $O(n)$ strategy table lookups (8 seconds/hand using Gurobi's LP solver)
- Our approach improved performance against strongest 2013 ACPC agents
 - 87+-50 vs. Hyperborean and 29+-25 vs. Slumbot

The need for opponent exploitation

- Game-solving approach produces unexploitable (i.e., “safe”) strategies in two-player zero-sum games
- But it has no guarantees in general-sum and multiplayer games
- Furthermore, even in two-player zero-sum games, a much higher payoff is achievable against weak opponents by learning and exploiting their mistakes

Opponent exploitation challenges

- Needs prohibitively many repetitions to learn in large games (only 3000 hands per match in the poker competition, so only have observations at a minuscule fraction of information sets)
- Partial observability of opponent's private information
- Often, there is no historical data on the specific opponent
 - Even if there is, it may be unlabelled or semi-labelled
- Recently, game-solving approach has crushed exploitation approaches in Texas hold 'em

Overview of our approach

- Start playing based on game theory approach
- As we learn opponent(s) deviate from equilibrium, adjust our strategy to exploit their weaknesses
 - E.g., the equilibrium raises 90% of the time when first to act, but the opponent only raises 40% of the time
 - Requires no prior knowledge about the opponent
- Find opponent's strategy that is “closest” to a pre-computed approximate equilibrium strategy and consistent with our observations of his actions so far
- Compute and play an (approximate) best response to the opponent model.

Deviation-Based Best Response algorithm

(generalizes to multi-player games)

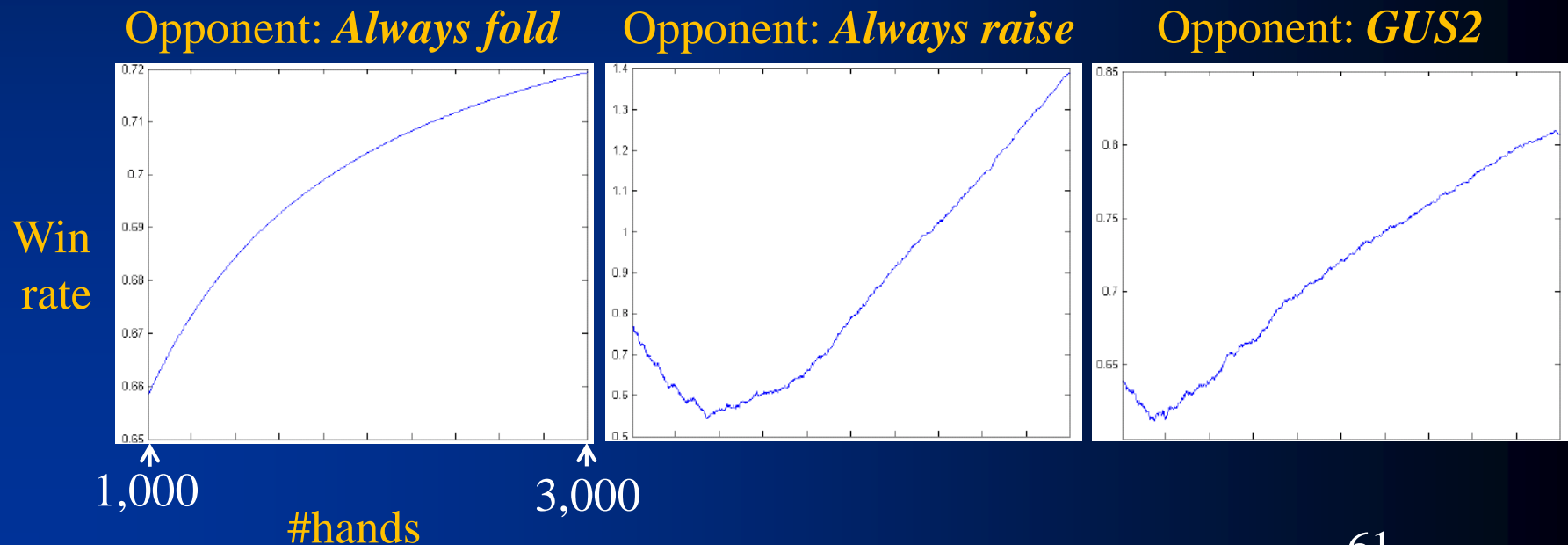
- Compute an approximate equilibrium
- Maintain counters of opponent's play throughout the match
- **for** $n = 1$ **to** |public histories|
 - Compute posterior action probabilities at n (using a Dirichlet prior)
 - Compute posterior bucket probabilities
 - Compute model of opponent's strategy at n
- **return** best response to the opponent model

Many ways to define opponent's "best" strategy that is consistent with bucket probabilities

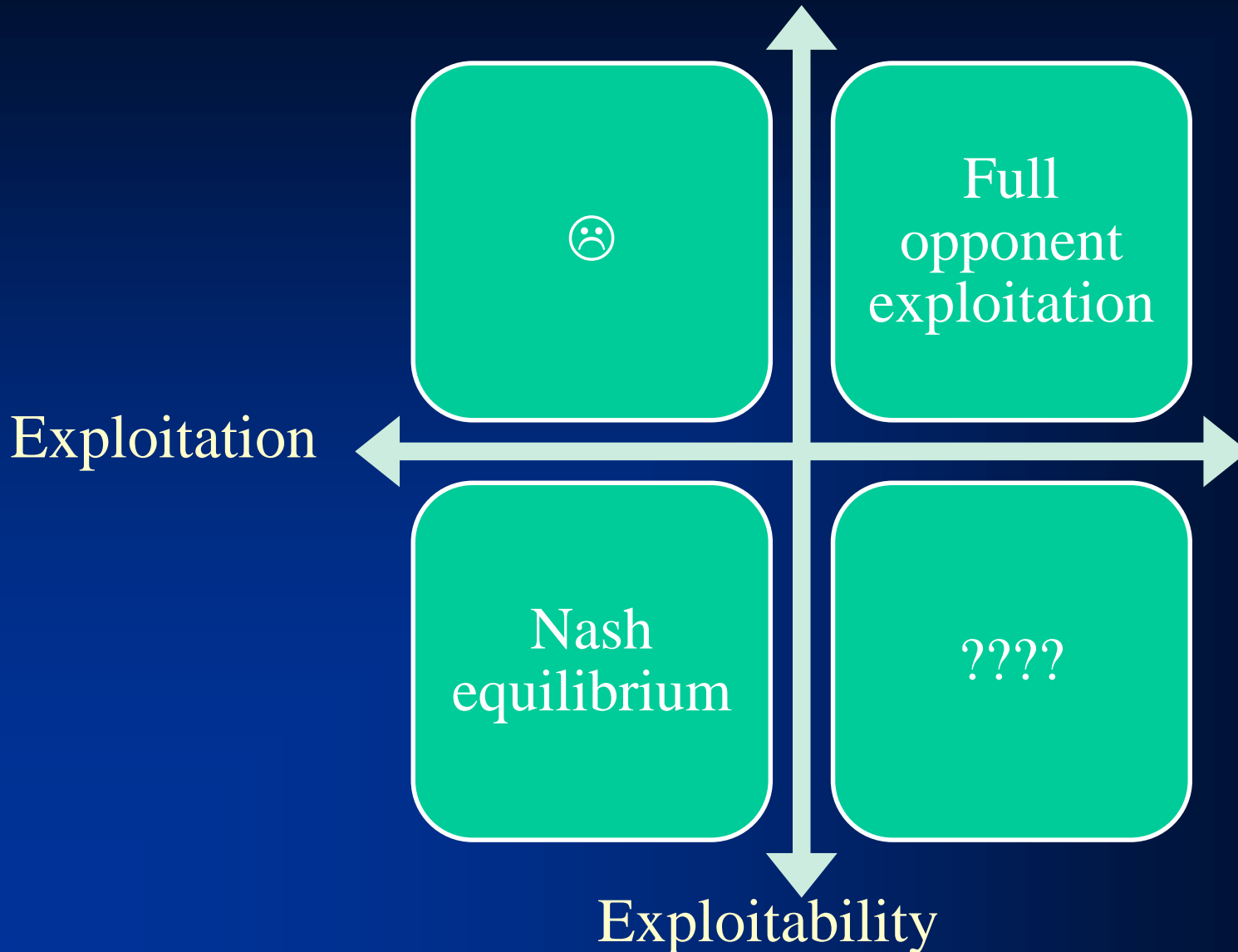
- L_1 or L_2 distance to equilibrium strategy
- Custom weight-shifting algorithm, ...

Experiments on opponent exploitation

- Significantly outperforms game-theory-based base strategy in 2-player limit Texas hold 'em against
 - trivial opponents (e.g., one that always calls and one that plays randomly)
 - weak opponents from AAAI computer poker competitions
- Don't have to turn this on against strong opponents



Exploitation-exploitability tradeoff



Safe opponent exploitation

- Definition. *Safe* strategy achieves at least the value of the (repeated) game in expectation
- Is safe exploitation possible (beyond selecting among equilibrium strategies)?

Rock-Paper-Scissors

- Suppose the opponent has played Rock in each of the first 10 iterations, while we have played the equilibrium σ^*
- Can we exploit him by playing pure strategy Paper in the 11th iteration?
 - Yes, but this would not be safe!
- By similar reasoning, any deviation from σ^* will be unsafe
- So safe exploitation is not possible in Rock-Paper-Scissors

Rock-Paper-Scissors-Toaster

	rock	paper	scissors	toaster
Rock	0,0	-1, 1	1, -1	4, -4
Paper	1,-1	0, 0	-1,1	3, -3
Scissors	-1,1	1,-1	0,0	3, -3

- *t* is *strictly dominated*
 - *s* does strictly better than *t* regardless of P1's strategy
- Suppose we play NE in the first round, and he plays *t*
 - Expected payoff of $10/3$
- Then we can play *R* in the second round and guarantee at least $7/3$ between the two rounds
- Safe exploitation is possible in RPST!
 - Because of presence of 'gift' strategy *t*

When can opponent be exploited safely?

- ~~Opponent played an (iterated weakly) dominated strategy?~~

R is a gift
but not iteratively weakly dominated

	L	M	R
U	3	2	10
D	2	3	0



- ~~Opponent played a strategy that isn't in the support of any eq?~~

R isn't in the support of any equilibrium
but is also not a gift

	L	R
U	0	0
D	-2	1

- Definition.** We received a *gift* if opponent played a strategy such that we have an equilibrium strategy for which the opponent's strategy isn't a best response
- Theorem.** Safe exploitation is possible iff the game has gifts

Exploitation algorithms

1.  Risk what you've won so far
 2.  Risk what you've won so far in expectation (over nature's & own randomization), i.e., risk the gifts received
 - Assuming the opponent plays a nemesis in states we don't observe
- **Theorem.** A strategy for a two-player zero-sum game is safe iff it never risks more than the gifts received according to #2
 - Can be used to make any opponent model / exploitation algorithm safe
 - No prior (non-eq) opponent exploitation algorithms are safe
 - We developed several new algorithms that are safe
 - Present analogous results and algorithms for extensive-form games of perfect and imperfect-information

Risk What You've Won in Expectation (RWYWE)

- Set $k^1 = 0$
- for $t = 1$ to T do
 - Set π_i^t to be k^t -safe best response to M
 - Play action a_i^t according to π_i^t
 - Update M with opponent's action a_{-i}^t
 - Set $k^{t+1} = k^t + u_i(\pi_i^t, a_{-i}^t) - v^*$

Experiments on Kuhn poker

- All the exploitative safe algorithms outperform Best Nash against the static opponents
- RWYWE did best against static opponents
 - Outperformed several more conservative safe exploitation algs
- Against dynamic opponents, best response does much worse than value of the game
 - Safe algorithms obtain payoff higher than the game value

Recap

- Background
- New approaches for game solving within the leading paradigm
- New game-solving paradigms
- Opponent exploitation
- Challenges and directions

Game solving challenges

- Nash equilibrium lacks theoretical justification in certain game classes
 - E.g., games with more than two players
 - Even in two-player zero-sum games, certain refinements are preferable
- Computing Nash equilibrium is PPAD-complete in certain classes
- Even approximating NE in 2p zero-sum games very challenging in practice for many interesting games
 - Huge state spaces
- Robust exploitation is preferable

Frameworks and directions

- Leading paradigm
 - Abstraction, equilibrium-finding, reverse mapping (action translation and post-processing)
- New paradigms
 - Incorporating qualitative models (can be used to generate human-understandable knowledge)
 - Real-time endgame solving
- Domain-independent approaches
- Approaches are applicable to games with more than two players
 - Direct: abstraction, translation, post-processing, endgame solving, qualitative models, exploitation algorithm
 - Equilibrium algorithms also, but lose guarantees
 - Safe exploitation, but guarantees maximin instead of value