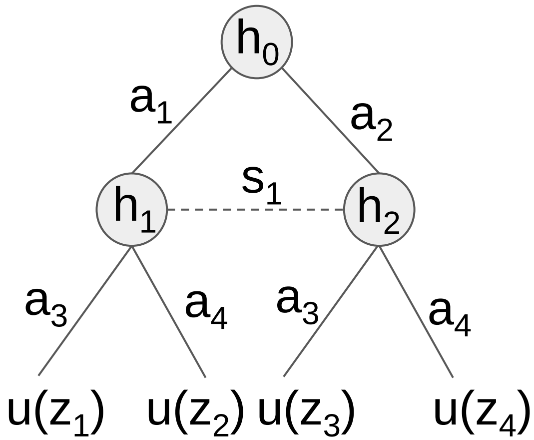
Counterfactual Regret Minimization

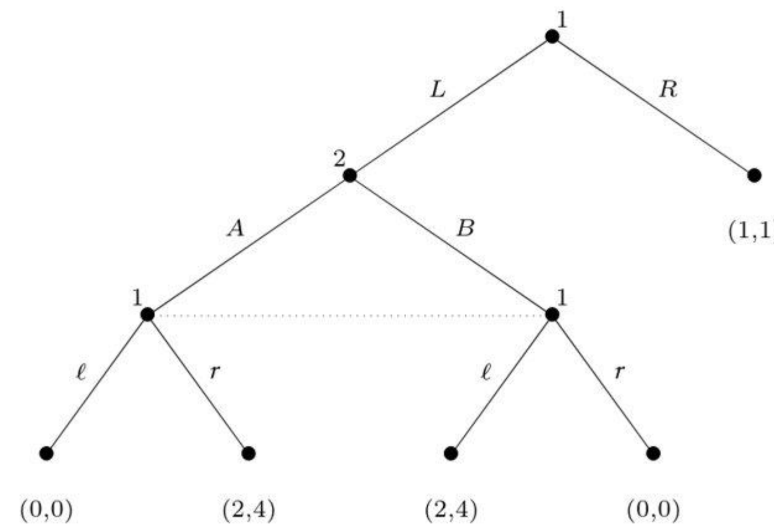
Extensive-Form Games Recap

- ***History h*** is ground truth state of the game
  - All cards for all players
- ***Information set s*** is observation for one player
  - Set of histories consistent with observation
  - The hand for one player
- ***Strategy (policy) $\pi_i(a|s)$*** gives distribution over actions at information set s
- ***Terminal history z*** is history at end of game
- ***Utility $u_i(z)$*** is utility for player i
- ***Reach probability $\eta^{\pi}(h)$*** is joint probability of reaching history h under π



Sequence Form Recap

- Store probabilities for sequence of actions
- Set of strategies is convex
- Expected utility of game is bilinear



	$\emptyset$	A	B
$\emptyset$	0, 0	0, 0	0, 0
L	0, 0	0, 0	0, 0
R	1, 1	0, 0	0, 0
$L\ell$	0, 0	0, 0	2, 4
Lr	0, 0	2, 4	0, 0

Figure 5.13: The sequence form of the game from Figure 5.10.

	A	B
$L\ell$	0, 0	2, 4
Lr	2, 4	0, 0
$R\ell$	1, 1	1, 1
Rr	1, 1	1, 1

Figure 5.14: The induced normal form of the game from Figure 5.10.

Sequence Form Recap

A realization plan x of player 1 fulfills

$$
\begin{aligned}
x(\sigma) &\geq 0 && \text{for all } \sigma \in S_1, \\
x(\emptyset) &= 1, && \\
\sum_{a \in A_s} x(\sigma_s a) &= x(\sigma_s) && \text{for every information set } s \in S.
\end{aligned}
$$

If x satisfies these constraints, it is in the *sequence-form polytope X*

$$\mathcal{X} = \{\boldsymbol{x} : \mathbf{F}_1 \boldsymbol{x} = \boldsymbol{f}_1, \boldsymbol{x} \geq \mathbf{0}\}$$

Goal is to solve this bilinear saddle point problem, where *X* and *Y* are the sequence-form polytopes

$$\max_{\boldsymbol{x} \in \mathcal{X}} \min_{\boldsymbol{y} \in \mathcal{Y}} \boldsymbol{x}^\top \mathbf{A} \boldsymbol{y}$$

Sequence-Form Strategies

Consequence: a lot of results carry over from normal form games when using sequence-form strategies!

- Nash equilibrium is a bilinear saddle point problem

$$\min_{x \in X} \max_{y \in Y} x^\top A y$$

where

$$x^\top A y = \sum_{z \in Z} u_1(z)\, p_{chance}(z)\, x[\sigma_1(z)]\, y[\sigma_2(z)]$$

- As long as we can construct regret minimizers for the sets of sequence-form strategies, we can use them to converge to Nash equilibrium in self play

Recall: Regret Minimization on Δ^n

for $t = 1, \dots, T$:

- Agent chooses an *action distribution* x^t
- Environment chooses a *utility vector* $u^t \in [0,1]^n$
- Agent observes u^t and gets utility $\langle u^t, x^t \rangle$

Agent goal: Minimize *regret*.
"How well do we do against **best, fixed** strategy in hindsight?"

$$R^T := \max_{\hat{x} \in X} \boxed{\left\{ \sum_{t=1}^{T} \langle u^t, \hat{x} \rangle \right\}} - \boxed{\sum_{t=1}^{T} \langle u^t, x^t \rangle}$$

Maximum utility that was achievable by the **best fixed** action in hindsight

Utility that was actually accumulated

Regret Minimization on Sequence-Form Decision Problems

for $t = 1, \ldots, T$:

- Agent chooses a sequence-form strategy x^t
- Environment chooses a *utility vector* $u^t \in [0,1]^n$
- Agent observes u^t and gets utility $\langle u^t, x^t \rangle$

Agent goal: Minimize *regret*.
"How well do we do against **best, fixed** strategy in hindsight?"

$$R^T := \max_{\hat{x} \in X} \boxed{\left\{ \sum_{t=1}^{T} \langle u^t, \hat{x} \rangle \right\}} - \boxed{\sum_{t=1}^{T} \langle u^t, x^t \rangle}$$

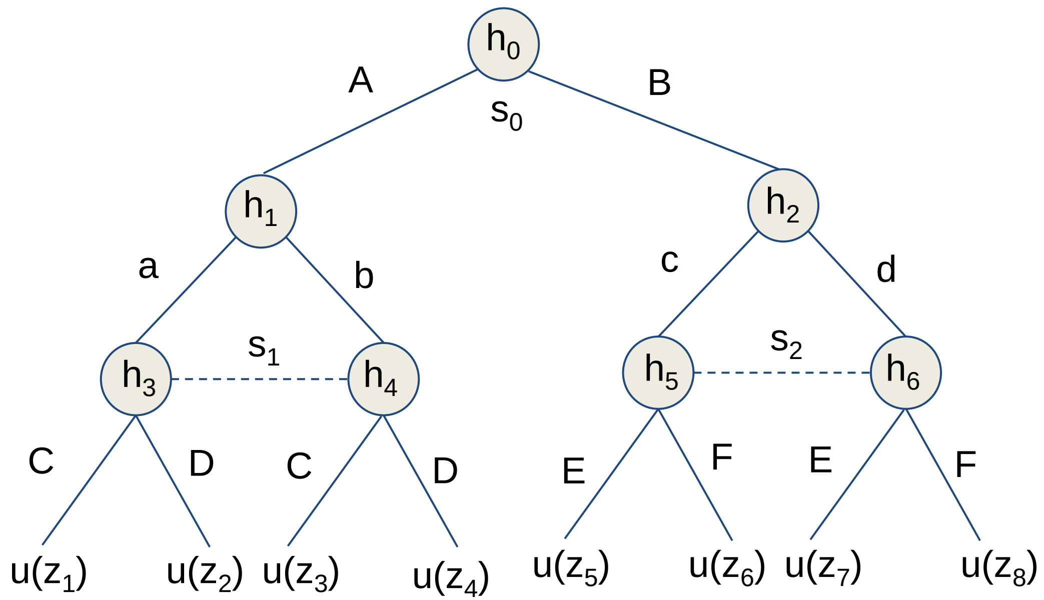
Maximum utility that was achievable by the **best fixed** action in hindsight

Utility that was actually accumulated

If we can do **regret minimization for sequence-form strategy spaces**, then we can **solve zero-sum extensive-form games!**

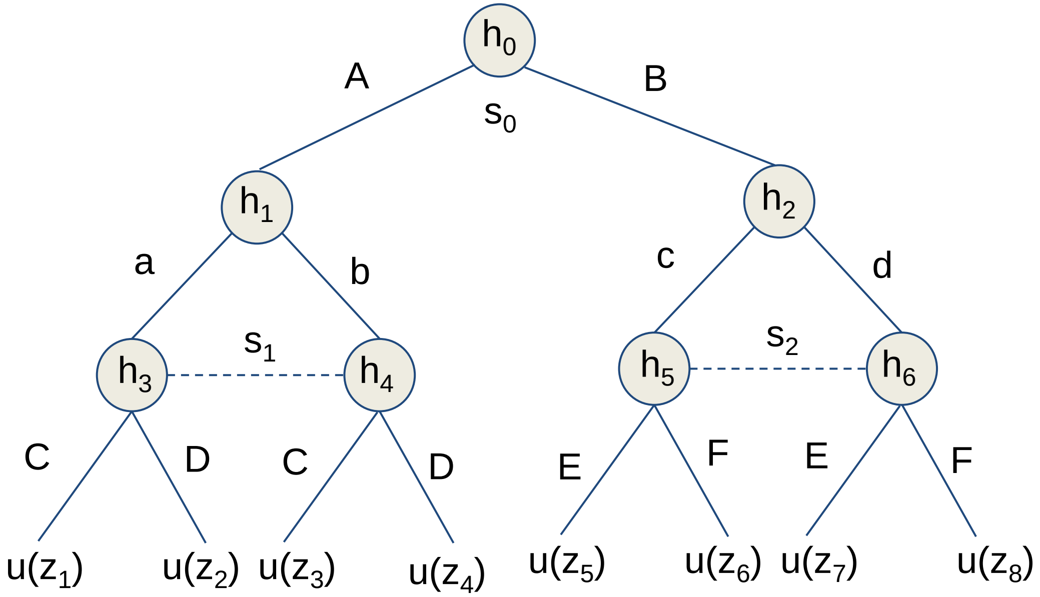
Regret Circuits

- Player 1 chooses A or B to decide which normal-form game to play
- Then the players play the normal-form game



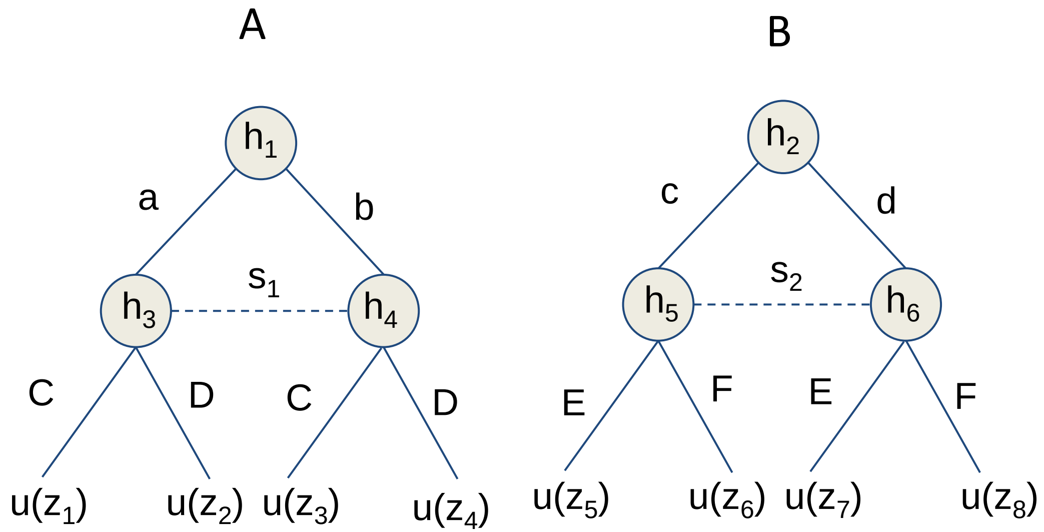
Regret Circuits

- No-regret approach:
  - Each iteration choose a behavior strategy
  - Opponent also chooses behavior strategy
  - Want average regret to go to zero



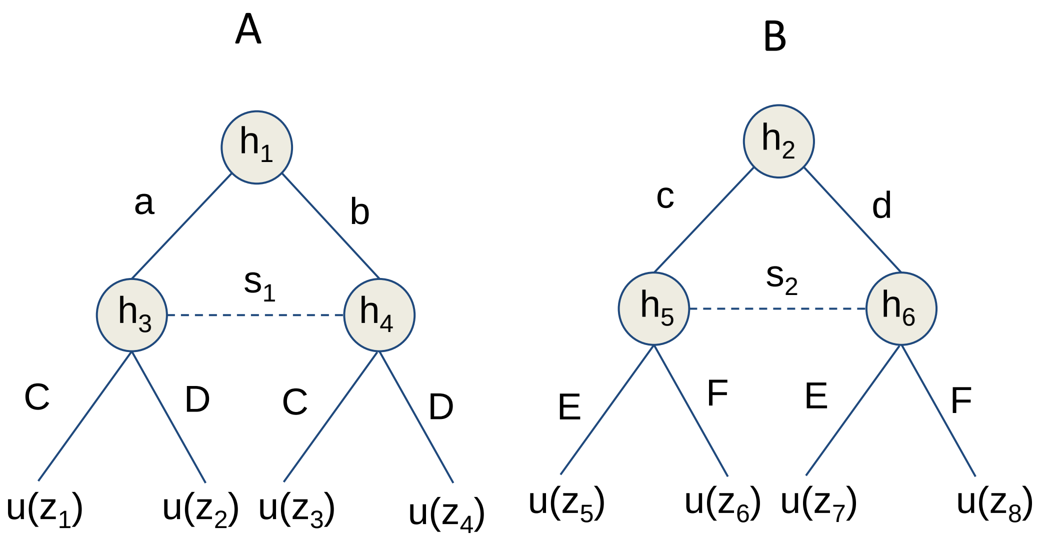
Regret Circuits

- Let's decompose this game into the two subgames
- Opponent chooses strategy for each subgame
  - a or b / c or d
- We choose strategy for each subgame
  - C or D / E or F



Regret Circuits

- We know how to solve this
- Just use a no-regret algorithm!
  - Let's use RM
- Denote regret in each subgame

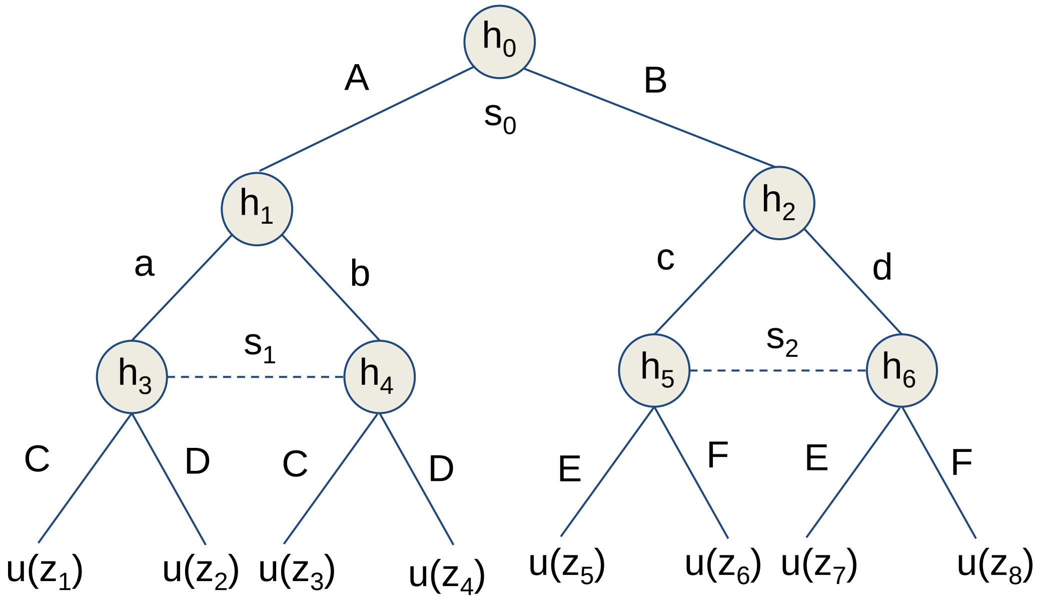


$$R_A^T = \arg\max_{\hat{x} \in \{C,D\}} \left\{ \sum_{t=1}^{T} \left(\langle u_t, \hat{x} \rangle - \langle u_t, x_t^A \rangle \right) \right\}$$

$$R_B^T = \arg\max_{\hat{x} \in \{E,F\}} \left\{ \sum_{t=1}^{T} \left(\langle u_t, \hat{x} \rangle - \langle u_t, x_t^B \rangle \right) \right\}$$

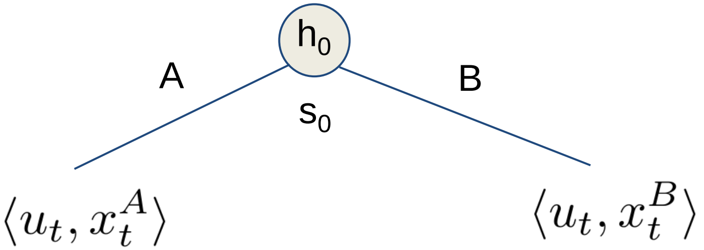
Regret Circuits

- Now let's add back root decision node
- Same setting: both players choose behavior strategy every timestep
- We know how to update s_1 and s_2
  - How to update s_0?



Regret Circuits

- Main idea: we pass values of subgames back up as if they were utilities
- Then run no-regret on this new problem



$$R_{\Delta}^T := \arg\max_{\hat{\lambda} \in \Delta^2} \left\{ \sum_{t=1}^{T} \hat{\lambda}_1 \langle u_t, x_t^A \rangle + \hat{\lambda}_2 \langle u_t, x_t^B \rangle \right\} - \sum_{t=1}^{T} \left(\lambda_1^t \langle u_t, x_t^A \rangle + \lambda_2^t \langle u_t, x_t^B \rangle \right)$$

Regret Circuits

- Can bound total regret by regret accumulated in subgames plus root regret

$$R^T \leq R^T_\Delta + \max\{R^T_A, R^T_B\}$$

- Intuition: we pay for learning top-level decision plus subgame decision
- Still get same order regret

Counterfactual Regret Minimization (CFR)

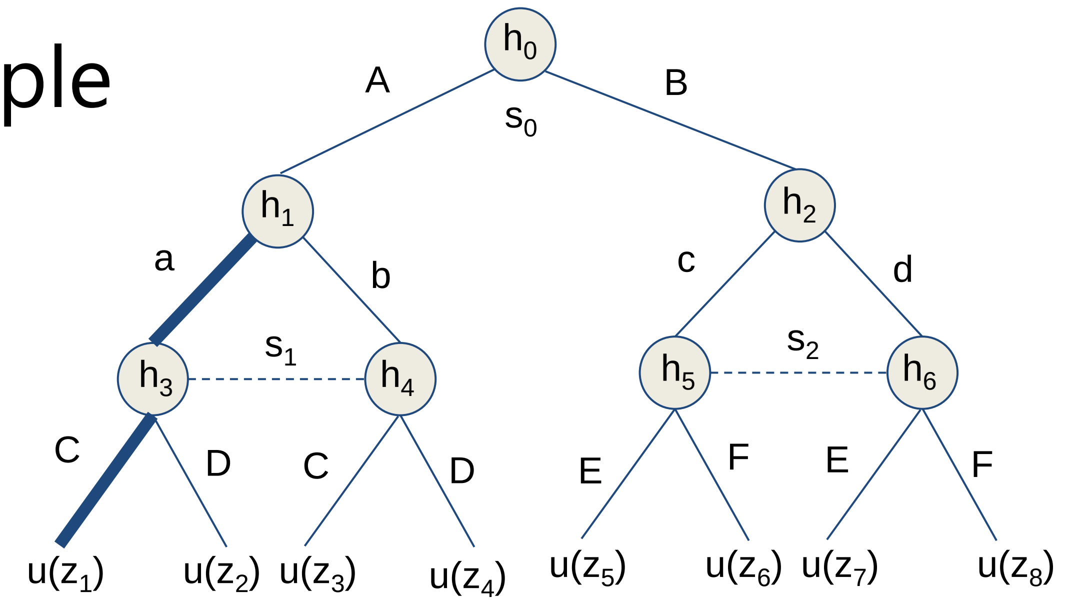
- Can extend this approach for arbitrary treeplexes
- Simply run RM on each individual information set
- Use "counterfactual values"

$$v_i(\pi, h) = \sum_{z \sqsupset h} \eta^{\pi}(h, z) u_i(z)$$

$$v_i^c(\pi, s) = \sum_{h \in s} \eta^{\pi}_{-i}(h) v_i(\pi, h)$$

Counterfactual Regret Minimization (CFR)

- Back to our example
- $\pi_1(C \mid s_1) = 1$
- $\pi_2(a \mid h_1) = 1$



$$v_1(\pi, h_3) = \sum_{z \sqsupset h} \eta^\pi(h, z) u_1(z) = u_1(z_1)$$

$$v_1^c(\pi, s_1) = \sum_{h \in s} \eta^\pi_{-i}(h) v_i(\pi, h) = u_1(z_1) = \boxed{\langle u_t, x_t^A \rangle}$$

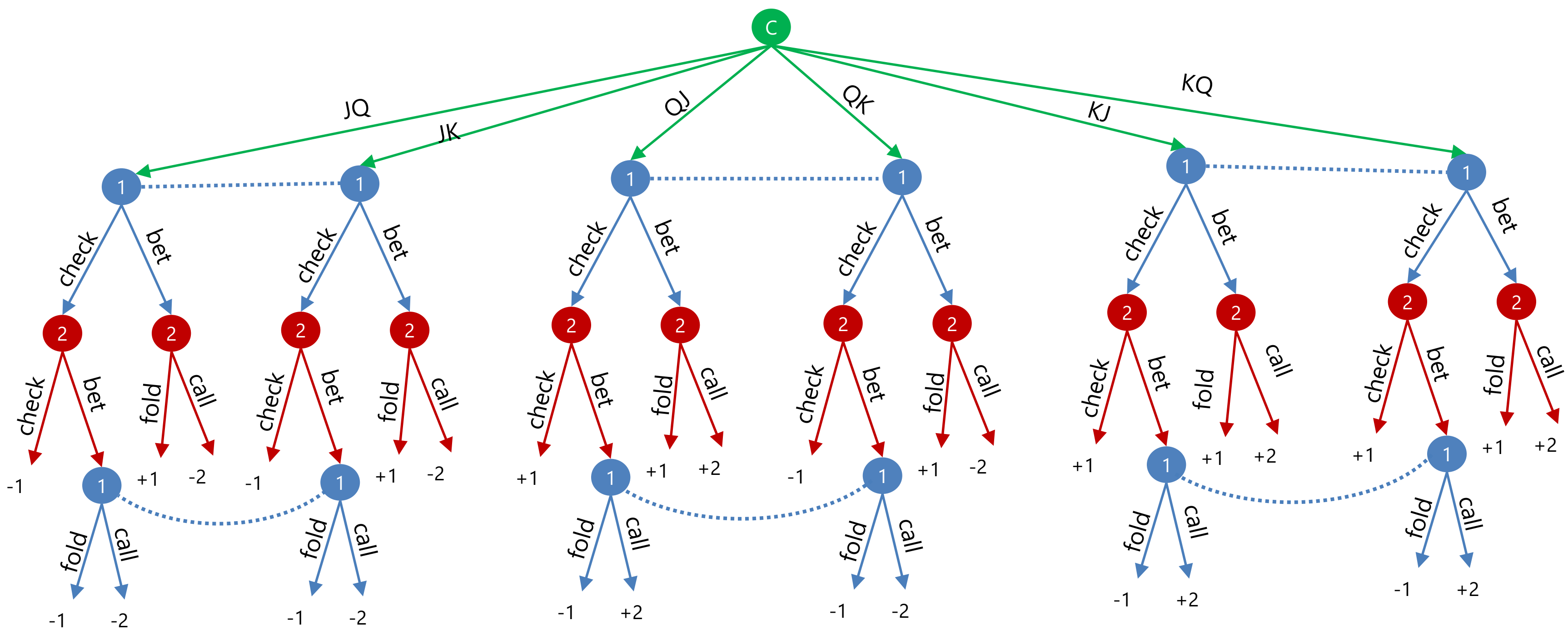
Regret circuit notation

Counterfactual Regret Minimization (CFR)

- Run Regret Matching at every decision point
- Feed counterfactual values to regret minimizer

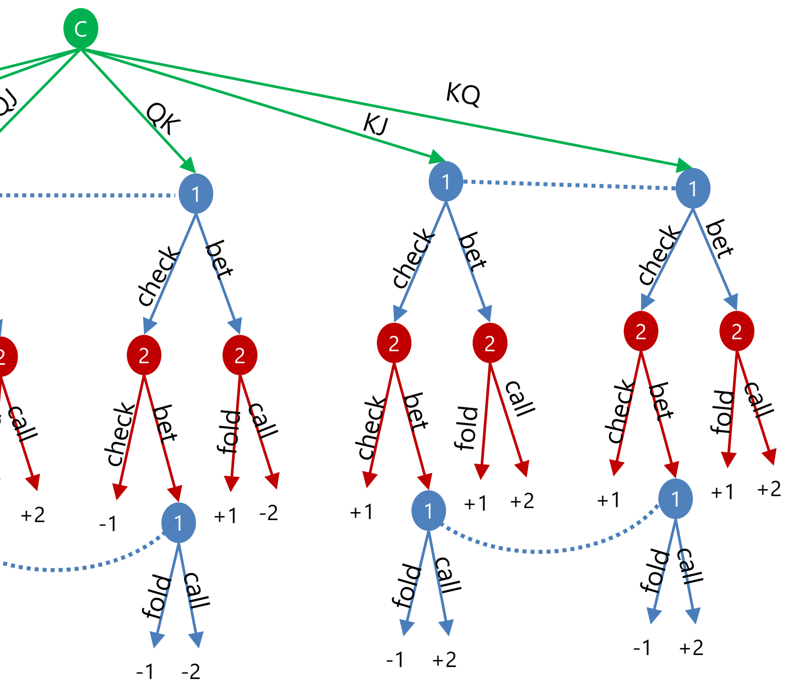
$$R_s^T := \max_{\hat{a} \in A_s} \sum_{t=1}^{T} r_i^c(\pi^t, s, \hat{a}) = \max_{\hat{a} \in A_s} \sum_{t=1}^{T} q_i^c(\pi^t, s, \hat{a}) - v_i^c(\pi^t, s)$$

Another CFR Example on Kuhn Poker



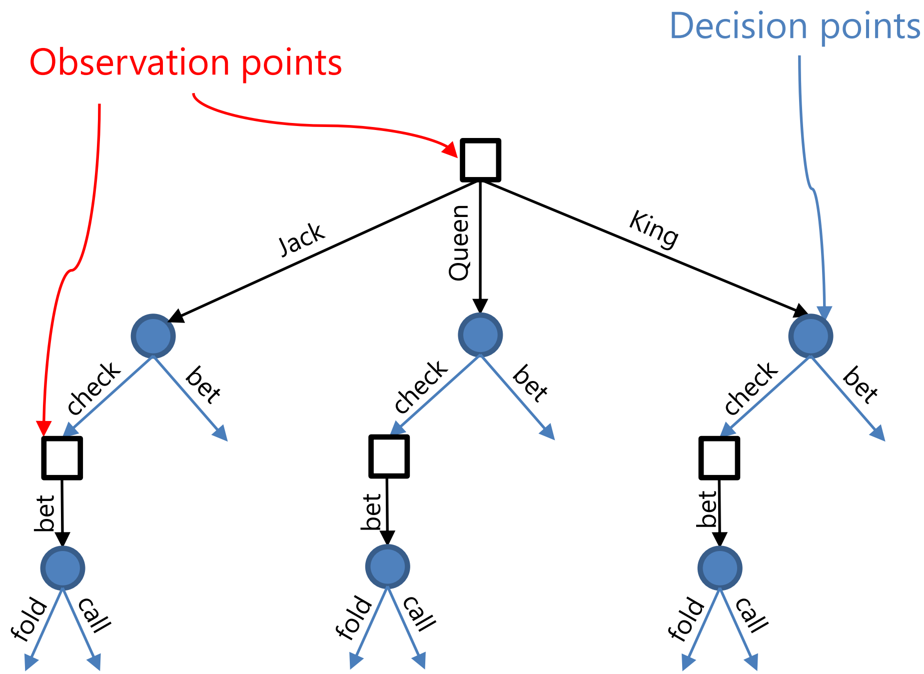
Information sets for red player (Player 2) are not shown

Two Representations



Game tree

Each node belongs to a specific player or chance



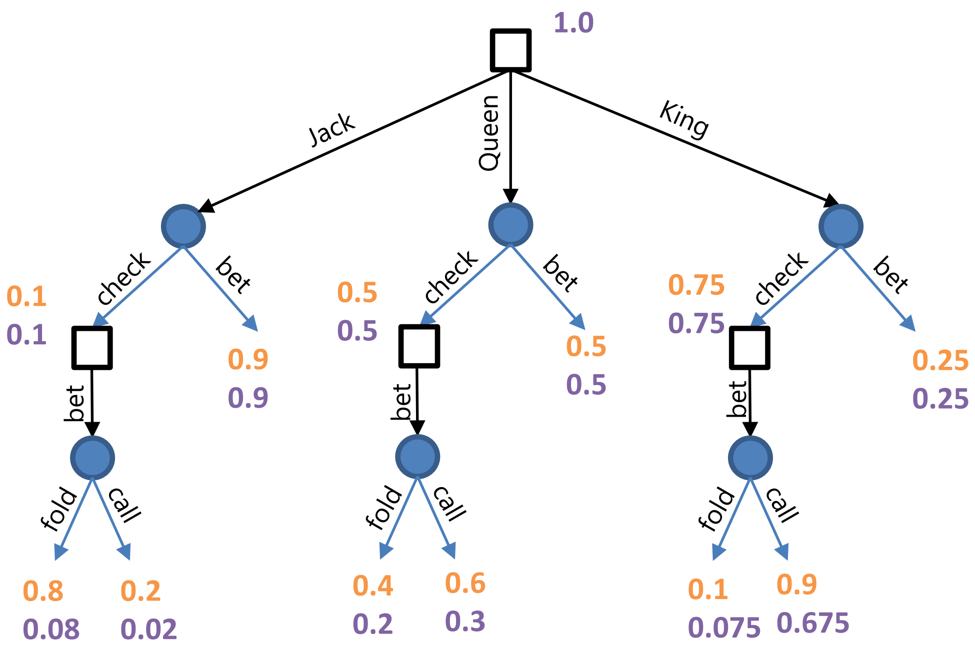
Tree-Form (Sequential) Decision Problem
aka. sequence-form decision problem
aka. treeplex
(for P1)

Represents the game from viewpoint of one player

This is the representation in regret minimization

Counterfactual Regret Minimization

Suppose the **orange** local strategies were output at time t by the regret minimizers at each decision point.

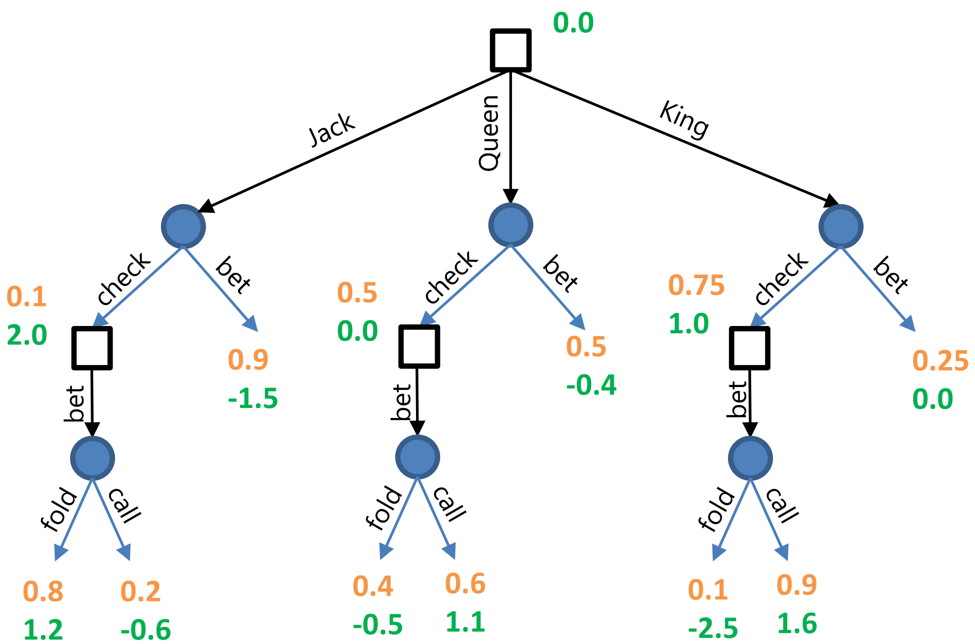


Step 1: Compute and output the sequence-form strategy x^t

Counterfactual Regret Minimization

Suppose the **orange** local strategies were output at time t by the regret minimizers at each decision point.

Now we observe some utility vector u^t



Remember: in regret minimization we make no assumption as to how the environment picked the utility vector. So, the green utilities may not actually be "real" payoffs in the game, which are

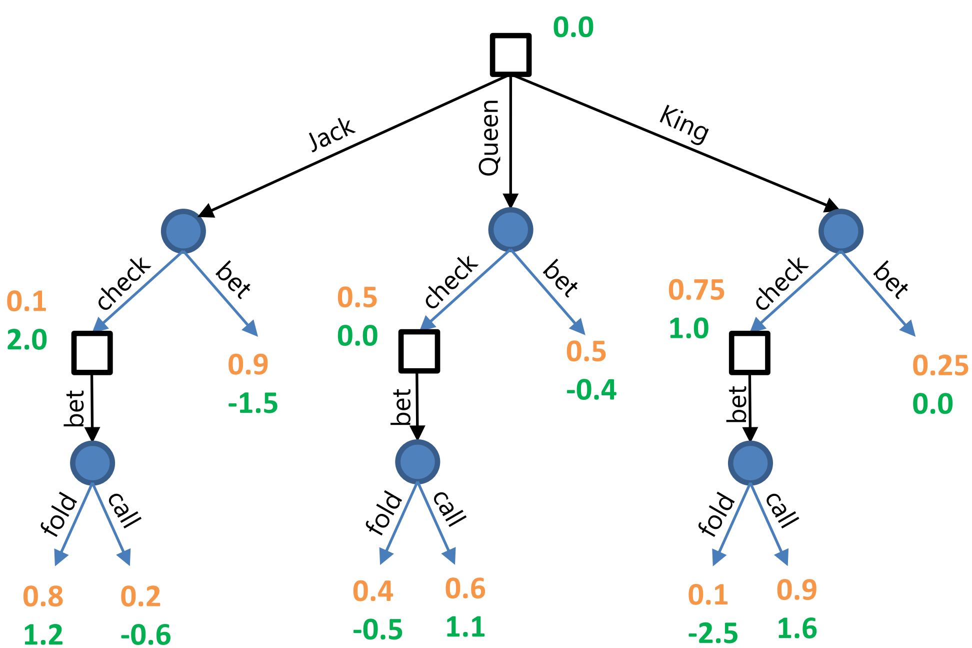
$$u^t[s] = \sum_{\substack{z \in Z: \\ \sigma_1(z) = s}} u(z)\, \mathrm{p}_{\text{chance}}(z)\, y^t[\sigma_2(z)]$$

(Not too important; this is just the vector such that $\langle u^t, x^t \rangle$ matches the expected value calculation from a few slides ago)

Counterfactual Regret Minimization

Suppose the **orange** local strategies were output at time t by the regret minimizers at each decision point.

Now we observe some utility vector u^t



Step 2: Compute **counterfactual utilities**

*At **observation points and leaves:***

counterfactual utility =

sum of counterfactual utilities of children +

utility value given at that observation point

*At **decision points**:*

counterfactual utility =

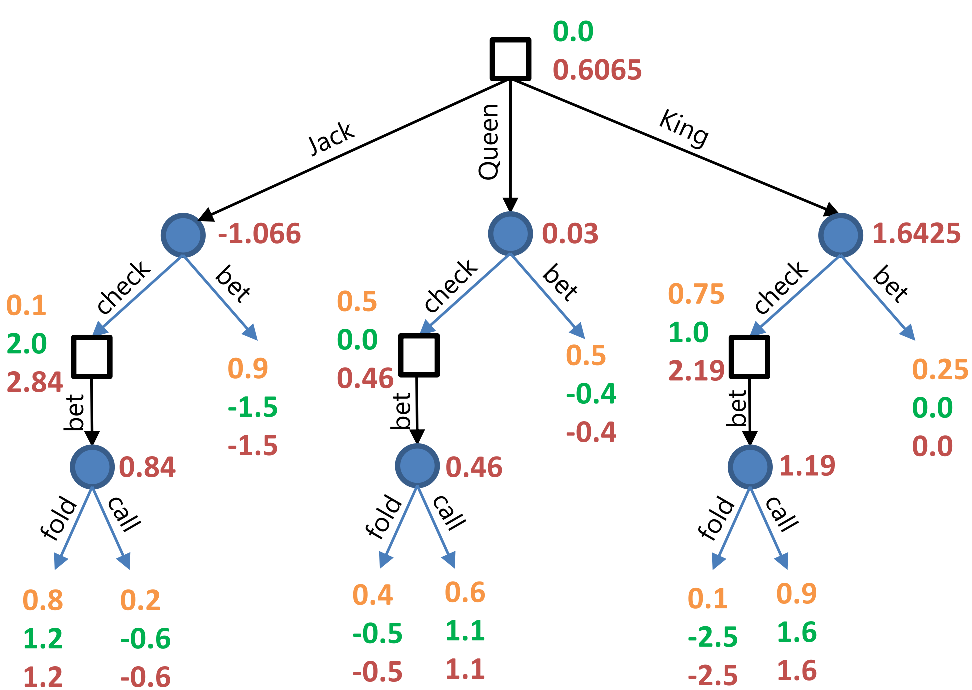
expected counterfactual utility of children

under local strategy at that decision point

Counterfactual Regret Minimization

Suppose the **orange** local strategies were output at time t by the regret minimizers at each decision point.

Now we observe some utility vector u^t



Step 2: Compute **counterfactual utilities**

*At **observation points and leaves:***

counterfactual utility =

sum of counterfactual utilities of children +

utility value given at that observation point

*At **decision points**:*

counterfactual utility =

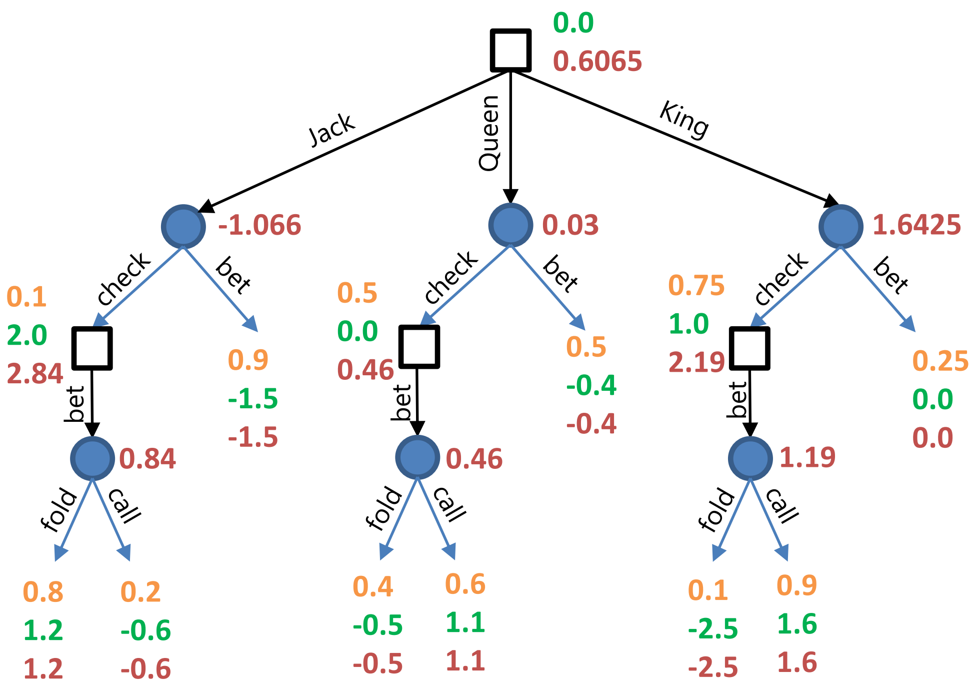
expected counterfactual utility of children

under local strategy at that decision point

Counterfactual Regret Minimization

Suppose the **orange** local strategies were output at time t by the regret minimizers at each decision point.

Now we observe some utility vector u^t



Step 2: Feed the counterfactual utilities to the regret minimizers at each decision point

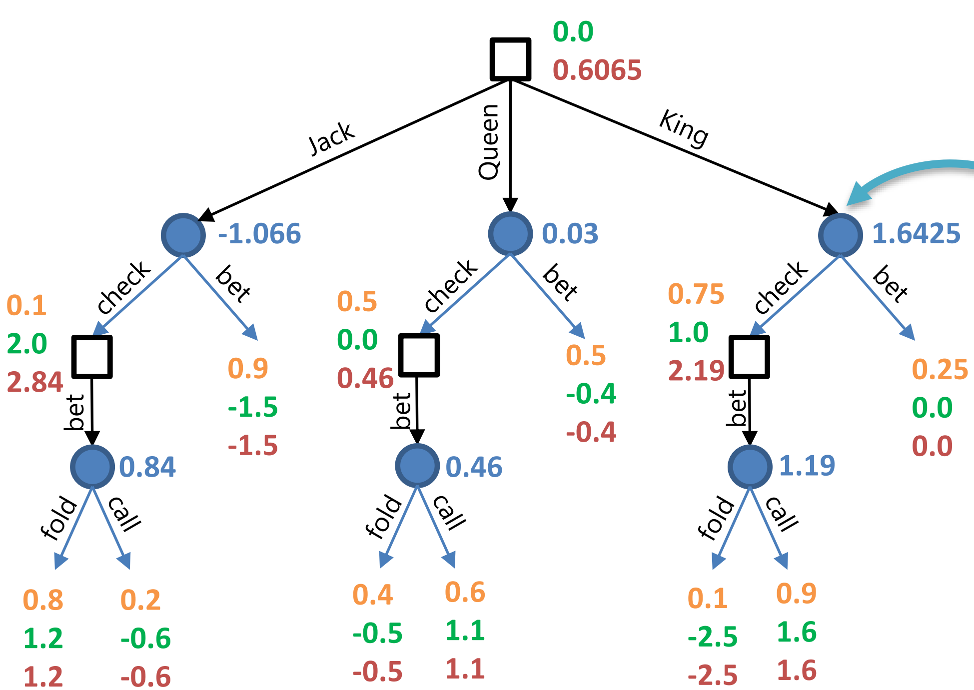
Note: Steps 1 & 2 can be done in a single bottom-up pass

Remember: this will work *no matter the choice of regret minimizers* (MWU, RM, RM+, Discounted RM, Optimistic RM+, etc). **We can also use different regret minimizers at different nodes, unlike the original CFR paper, which used RM**

Counterfactual Regret Minimization

Suppose the **orange** local strategies were output at time t by the regret minimizers at each decision point.

Now we observe some utility vector u^t



Step 2: Feed the counterfactual utilities to the regret minimizers at each decision point

Example:

Feed utilities (2.19, 0) to the regret minimizer for this decision point

CFR Guarantees

- **Theorem**: the regret cumulated by CFR can be bounded as

$$R_{CFR} \leq \underbrace{\sum_{j \in J}}_{\text{Decision points}} \max\{0, \underbrace{R_j}_{\text{Regret of local regret minimizer for decision point } j}\}$$

- *Therefore*: if the local regret minimizers all have regret $O(\sqrt{T})$, then CFR has regret $O(\sqrt{T})$ (where the O hides game-dependent constants)

Therefore: if both players in a zero-sum extensive-form game play according to CFR, the average strategy converges to Nash equilibrium at rate $O(1/\sqrt{T})$

[Farina, Kroer & Sandholm, Regret Circuits: Composability of Regret Minimizers, ICML 2019]

Why is CFR Superior in Practice?

... to second-order methods (which can offer convergence rate $1/e^T$)?

- Does not require solving large linear systems
- Second-order methods (interior point, ...) don't fit in memory for large games

... to general-purpose regret minimizers (FTRL & OMD)?

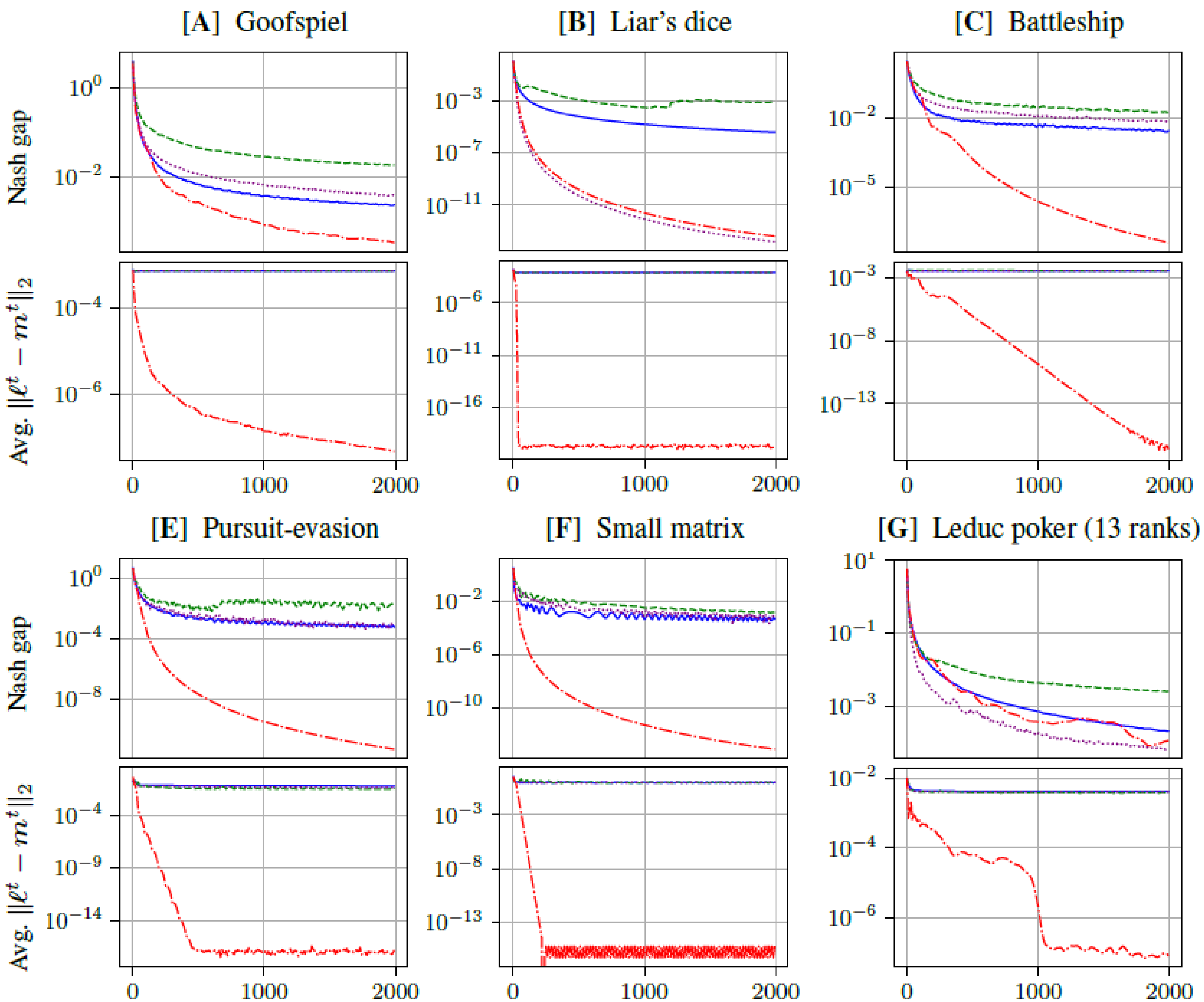
- CFR uses an approach local to each decision point (easier to parallelize, warm-start, etc.)
  - [Brown & Sandholm, Reduced Space and Faster Convergence in Imperfect-Information Games via Pruning. ICML-17]
  - [Brown & Sandholm, Strategy-based warm starting for regret minimization in games, AAAI 2016]
- No need for expensive projections onto feasible strategy polytope (think projected gradient descent)

Other approaches

- Offline first-order methods:
  - E.g., mirror prox (MP) or excessive gap technique (EGT)
    - O(1/T) convergence instead of CFR's O(1 / $\sqrt{T}$)
  - Regret minimization is decentralized, and with optimism it matches the same theoretical rates. Also, it performs better empirically

- *All in all, regret-based methods are today the scalable state of the art*

CFR Framework + Predictivity



Dimension of the games

	Decision points	Actions	Leaves
[A]	4.4×10^6	5.3×10^6	1.7×10^6
[B]	2.5×10^4	4.9×10^4	1.5×10^5
[C]	1.1×10^6	3.2×10^6	3.5×10^6
[D]	4.4×10^4	1.3×10^5	5.4×10^7
[E]	1.2×10^4	6.9×10^4	1.2×10^5
[F]	2	4	4
[G]	5.1×10^3	1.2×10^4	9.9×10^4

Important Takeaways

- You can construct a regret minimizer for **sequential** decision making problems by combining regret minimizers for individual decision points

  $\Rightarrow$ Improvements on simplex domains carry over to extensive-form domains!

- Predictivity works well also in extensive-form domains

Techniques to Further Increase Scalability of CFR

- Using utility estimators
  - Similar idea as stochastic gradient descent vs gradient descent
  - Instead of exactly computing the green numbers (gradients of the utility function), we use cheap unbiased estimators
  - Popular estimator: sample a trajectory in the game tree and use importance sampling
  - **"Monte Carlo CFR"** [Monte Carlo Sampling for Regret Minimization in Extensive Games; Lanctot, Waugh, Zinkevich, Bowling NIPS 2009]
  - Even better algorithm, **ESCHER**, does not use importance sampling [McAleer, Farina, Lanctot & Sandholm *ICLR*-23]