Deep Multiagent RL

Gabriele Farina

MIT 6.7980 · Topics in Multiagent Learning · Fall 2026

What makes extensive-form games tractable?

They are convex games

When represented using sequence-form strategies.

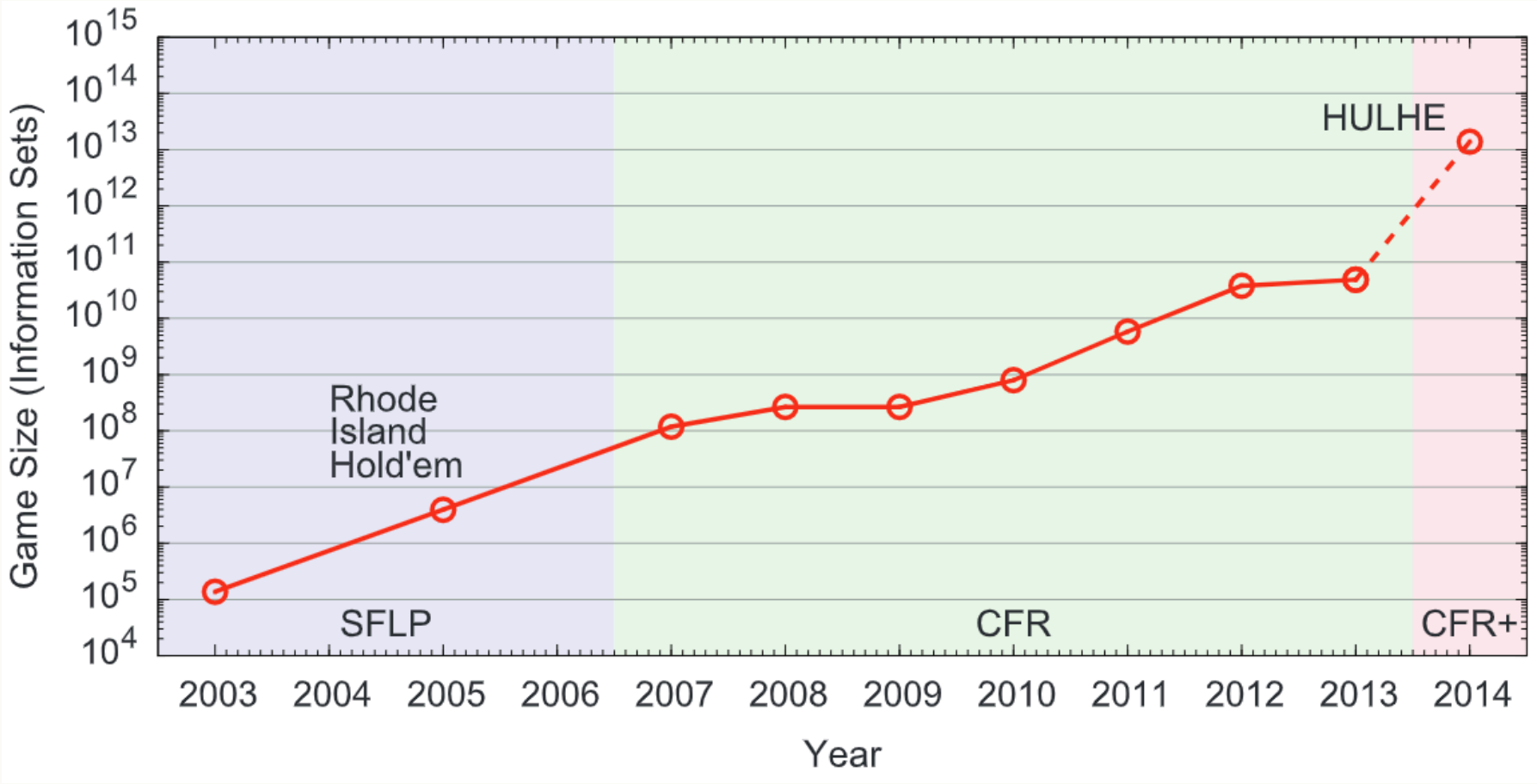
In two-player zero-sum games we can solve them via the *convex optimization* problem

$$\max_{x \in \mathcal{X}} \min_{y \in \mathcal{Y}} \; x^\top U y$$

The sets $\mathcal{X}$ and $\mathcal{Y}$ are convex polytopes. The payoff is bilinear.

Gradient ascent in sequence-form space will work.

How do we scale to large games?

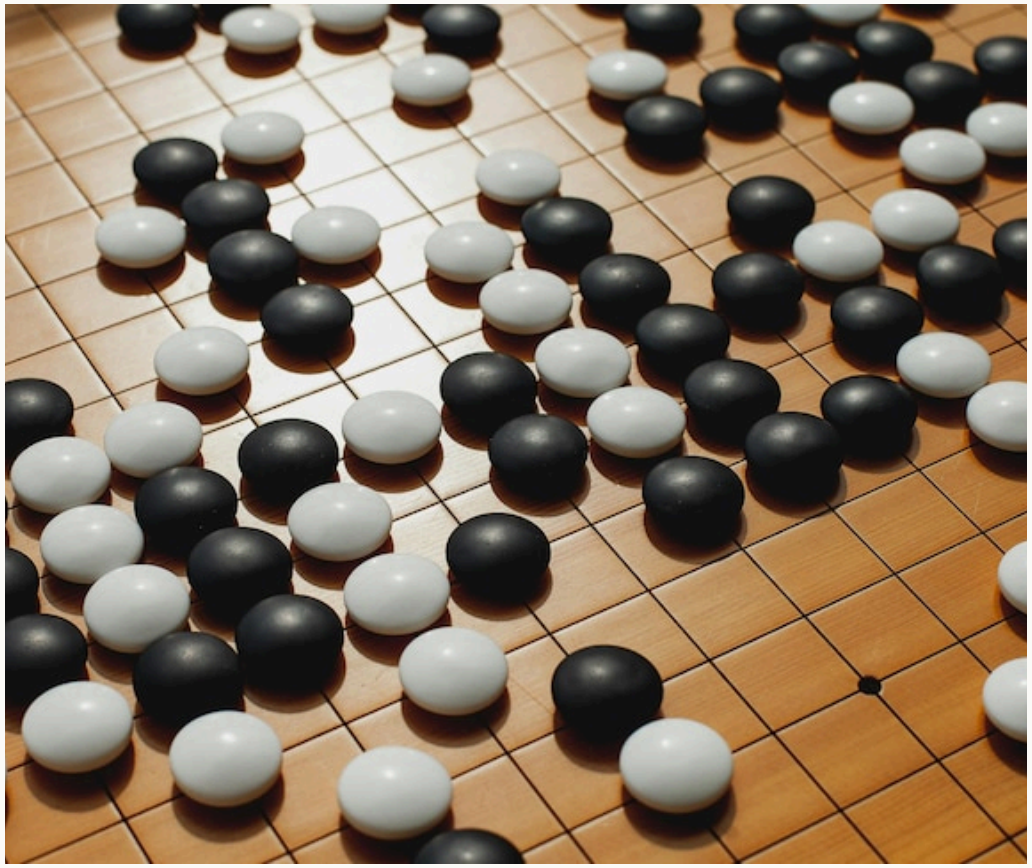


Unique information sets after removing symmetries. Historical results through 2014.

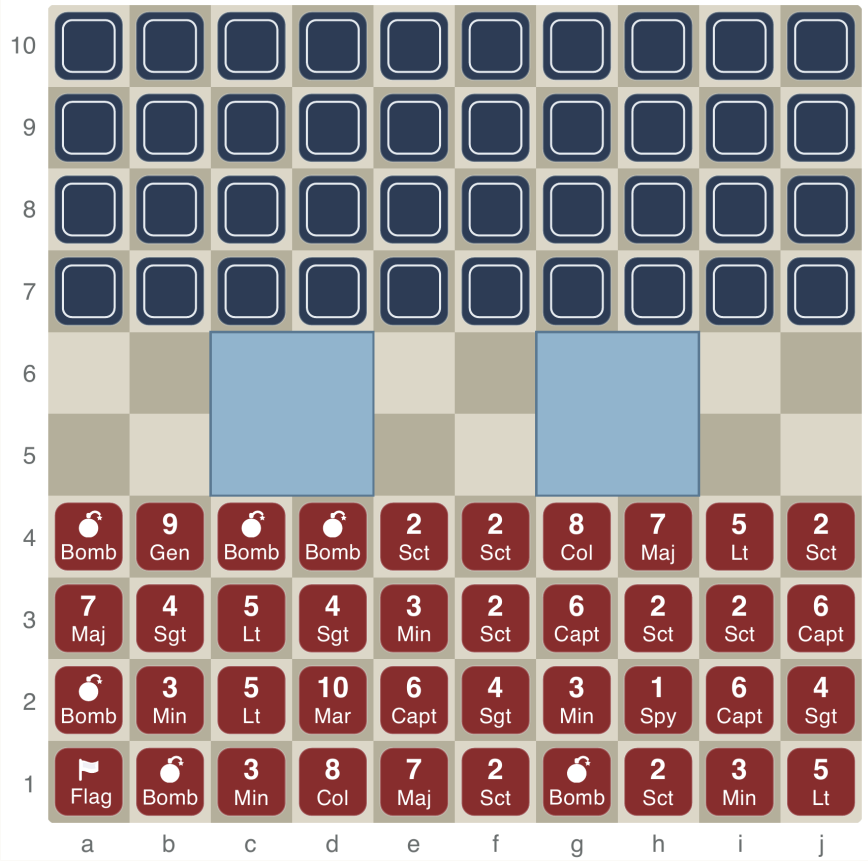
Bowling, Burch, Johanson & Tammelin (2015), Heads-up Limit Hold'em Poker is Solved, Fig. 2.

How do we scale to large games?

Go



Stratego



Go photo: Elena Popova / Unsplash. Stratego board: Sokota et al. (2026), Fig. 2a.

How should we approach large games?

Two directions

1. “Deepify” game theory

Can we learn best responses and regrets?

Neural fictitious play · ESCHER · Deep CFR

↑ **We start here**

2. “Game-theorify” deep RL

Can we make self-play policy gradients converge?

Policy gradient · PPO · MMD

“Deepifying” game-theoretic algorithms

First direction

Keep fictitious play, double oracle, CFR, ... but approximate the expensive step with a neural network.

Idea 1: learn best responses

Use deep RL to approximate the **best response**.

Algorithm 1 Fictitious play

$\bar{\pi}_i \leftarrow \text{uniform_policy}()$
while $\text{is_time_left}()$ **do**
  $b_i \leftarrow \text{best_response}(\bar{\pi}_{-i})$
  $\bar{\pi}_i \leftarrow \text{update_average}(\bar{\pi}_i, b_i)$
end while
return $\bar{\pi}$

Algorithm 2 Double oracle

$S_i \leftarrow \{\text{uniform_policy}()\}$
while $\text{is_time_left}()$ **do**
  $\sigma \leftarrow \text{nash}(S_1 \times S_2)$
  $b_i \leftarrow \text{best_response}(\sigma_{-i})$
  $S_i \leftarrow S_i \cup \{b_i\}$
end while
return σ

Same expensive primitive: best response

How accurately—and how quickly—can deep RL solve it?

Lanctot et al. (2017), PSRO.

Idea 2: learn counterfactual regrets

CFR computes a local learning signal at each information set I.

$$r_t(I, a) = v_i^{\pi_t}(I, a) - v_i^{\pi_t}(I)$$

$$R_t(I, a) = \sum_{\tau=1}^{t} r_\tau(I, a)$$

Deep CFR: learn regrets, then apply regret matching

Fit predictions $\widehat{R}_t(I, a)$ proportional to cumulative regrets.

$$\pi_{t+1}(a \mid I) = \frac{[\widehat{R}_t(I, a)]_+}{\sum_b [\widehat{R}_t(I, b)]_+}.$$

Brown et al. (2019), Deep CFR, §4. If all predictions are nonpositive, select an action with the largest prediction.

Three examples

NFSP

Fictitious play + deep RL

Learn a best-response policy and fit a separate average-policy network from a reservoir of behavior.

PSRO

Response oracles + deep RL

Grow a population of policies and use a meta-solver to mix them in an empirical game.

Deep CFR

CFR + deep learning

Fit sampled counterfactual advantages and the average strategy without a complete tabular representation.

Heinrich & Silver (2016), NFSP; Lanctot et al. (2017), PSRO; Brown et al. (2019), Deep CFR.

Pros and cons of deepified game-theoretic approaches

- \+ Clear theoretical foundations.
- \+ Clear sense for how the tasks are split up.
- − Learning a strong best response is expensive.
- − Growing and evaluating a policy population costs memory and simulation.
- − Even exact fictitious play and double oracle can require many outer iterations.

Can we just do gradient ascent and not think?

This leads us toward self-play policy gradients.

Slow-convergence examples: Wang (2025), fictitious play; Zhang & Sandholm (2024), double oracle.

How should we approach large games?

Two directions

1. “Deepify” game theory

Can we learn best responses and regrets?

Neural fictitious play · ESCHER · Deep CFR

2. “Game-theorify” deep RL

Can we make self-play policy gradients converge?

Policy gradient · PPO · MMD

↑ We now turn here

“Game-theorifying” deep reinforcement learning

Second direction

Let’s start from algorithms designed to scale, and modify them.

Choice I: out with sequence-form, in with behavioral

Sequence-form strategies are constrained and can get very small.

As much as they are the source of tractability, they are the first to go. Instead, we insist on **behavioral** strategies ("*policies*"): a probability distribution at each information state.

In fact, our policy net will output *logits* z_θ:

$$\pi_\theta(a \mid I) = \frac{\exp z_\theta(I, a)}{\sum_{b \in A(I)} \exp z_\theta(I, b)}$$

A nonlinear parametrization of the strategy

The parameters θ determine action probabilities, which multiply along each player sequence.

$$x(\theta)_{I,a} = \prod_{(I',a') \preceq (I,a)} \pi_\theta(a' \mid I'); \qquad J(\theta; y) = x(\theta)^\mathsf{T} U y.$$

Two main questions:

1. Will (stochastic) gradient ascent still work? and
$$\theta_{t+1} = \theta_t + \eta_t \widehat{\nabla_\theta J}(\theta_t; y_t)$$
2. how should we estimate the gradient?

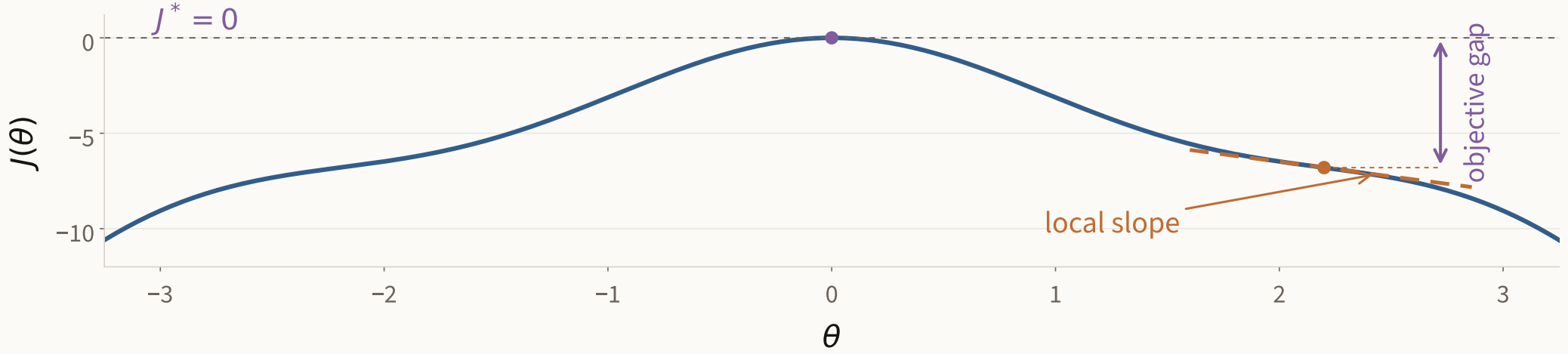
Is there hope for gradient ascent?

For a fixed opponent, the objective is concave in the sequence-form strategy, but not in the parametrization that we picked... We can think of this as *hidden concavity*.

Example of gradient domination: PŁ condition

Polyak–Łojasiewicz (PŁ) condition

$$J^* - J(\theta) \leq \frac{1}{2\mu} \|\nabla J(\theta)\|_2^2, \qquad \mu > 0$$



Can the gradient vanish while the objective gap is positive?

No: every stationary point is globally optimal.

Karimi, Nutini & Schmidt (2016), §2; Kalogiannis & Farina (2025).

Why does gradient ascent converge?

Assume J is L-smooth, attains J^*, and satisfies PŁ with $\mu > 0$.
Take $\theta^{t+1} = \theta^t + L^{-1}\nabla J(\theta^t)$.

1. Smoothness: one step makes progress.

$$J(\theta^{t+1}) \geq J(\theta^t) + \frac{1}{2L}\|\nabla J(\theta^t)\|_2^2$$

2. PŁ condition: the gradient controls the objective gap.

$$\frac{1}{2}\|\nabla J(\theta^t)\|_2^2 \geq \mu\big(J^* - J(\theta^t)\big)$$

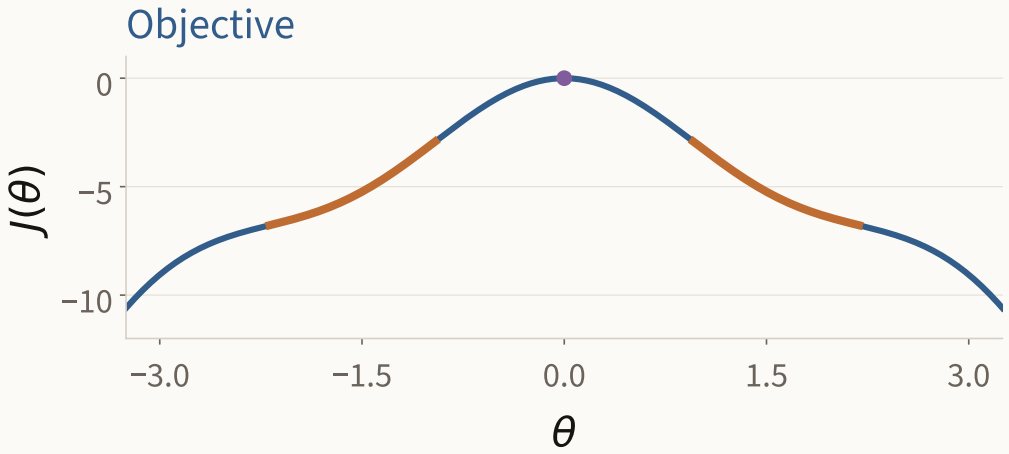
Hence, geometric convergence—even without concavity

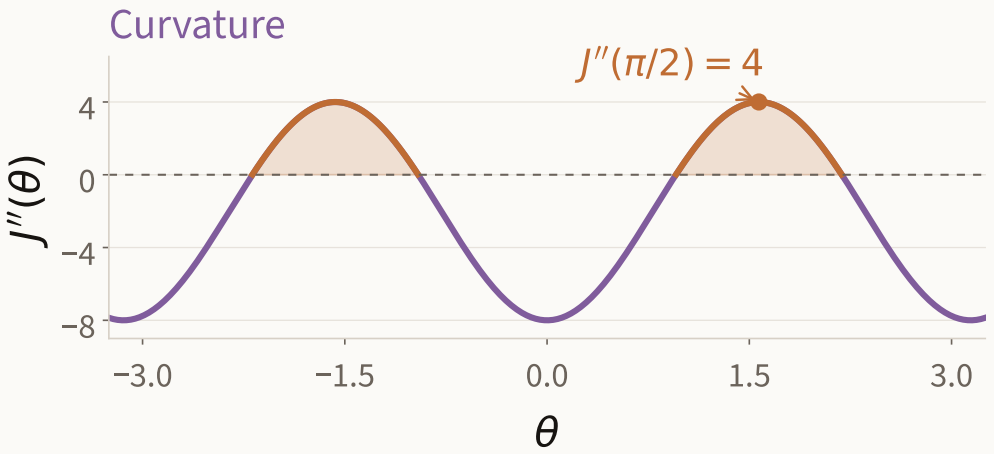
$$J^* - J(\theta^t) \leq \left(1 - \frac{\mu}{L}\right)^t \big(J^* - J(\theta^0)\big)$$

Exact ascent on one fixed, unconstrained objective. Self-play requires additional arguments. Karimi, Nutini & Schmidt (2016), Theorem 2.1, maximization form.

PŁ ⇐≠ Strong Convexity

$$J(\theta) = -\theta^2 - 3\sin^2\theta, \qquad J^* = J(0) = 0$$





$$J''(\pi/2) = 4 > 0, \qquad |J''(\theta)| \le 8$$

Nonconcave, globally smooth, and gradient dominated

PŁ holds with $\mu = 1/32$. Step size $1/8$ gives a global geometric bound on the objective gap.

Karimi, Nutini & Schmidt (2016), §2.2 give $F(\theta) = \theta^2 + 3\sin^2\theta$ as a nonconvex PŁ example. Here $J = -F$ converts it to maximization; figures are drawn from the analytic formulas.

What can we prove?

[KF25] Regularized policy gradient in imperfect-information EFGs

In finite two-player zero-sum extensive-form games with perfect recall, suitable alternating updates converge in the last iterate to an ε-Nash equilibrium, with polynomial iteration and sample complexity.

- Independent tabular behavioral policies.
- Explicit exploration, truncated policies, and suitable Euclidean or entropy regularization.
- Direct, softmax, and natural policy-gradient variants.

Kalogiannis & Farina, NeurIPS (2025), Theorems 3.1 & 3.3.

Choice II: Which gradient estimator?

Begin with a single decision: $J(z) = p(z)^\top q$.

$$\nabla_p J = q$$

Choose any proposal μ with full support. Sample $a \sim \mu$.

$$\hat{q} = \frac{q_a}{\mu_a} e_a, \qquad \mathbb{E}[\hat{q}] = q$$

Importance weights

Low-prob. actions carry a large correction $1/\mu_a$. Which distribution should we choose?

Srinivasan et al. (2018), actor-critic policy optimization; score-function estimation: Williams (1992).

Uniform exploration?

In self-play, we also control the opponent. We can choose both exploration policies.

Theory suggests a uniform strategy: minimize the largest importance weight.

$$\min_{\substack{\mu \in \Delta(A) \\ \mu_a > 0}} \max_{a \in A} \frac{1}{\mu_a} = |A|, \qquad \mu_a = \frac{1}{|A|}$$

But uniform exploration is a bad idea at the scale where you need deep RL.

Why?

Learn on the distribution we expect to encounter

Uniform play visits unlikely states. *On-policy* sampling is desirable to spend resources on parts of the decision space that are important/expected during play.

The minimax bound is for one decision. Trajectory corrections multiply. On-policy sampling still needs exploration for coverage.

Policy gradient estimator: REINFORCE

1. Recall our sequence-form reach

Let σ list our own information-set/action pairs along a generic path.

$$x_\theta(\sigma) = \prod_{(I,a)\in\sigma} \pi_\theta(a \mid I)$$

2. Turn own reach into a full trajectory probability

- τ be a terminal path,
- $R(\tau)$ our payoff
- $\sigma(\tau)$ our own (I, a) trajectory
- $q(\tau)$ is the fixed opponent/chance factor
- $P_\theta(\tau)$ is the full path probability

$$P_\theta(\tau) = q(\tau)\, x_\theta(\sigma(\tau)), \qquad J(\theta) = \sum_\tau P_\theta(\tau)\, R(\tau) = \mathbb{E}_{\tau\sim P_\theta}[R(\tau)]$$

Finite perfect-recall game; positive policy probabilities; opponent and chance held fixed. Williams (1992); Srinivasan et al. (2018), §§2.2, 4 and Appendix E.

Policy gradient estimator: REINFORCE

$$P_\theta(\tau) = q(\tau)x_\theta(\sigma(\tau)), \qquad J(\theta) = \mathbb{E}_{\tau\sim P_\theta}[R(\tau)]$$

3. Differentiate log probability: the trajectory score

$$s_\theta(\tau) := \nabla_\theta \log P_\theta(\tau) = \sum_{(I,a)\in\sigma(\tau)} \nabla_\theta \log \pi_\theta(a \mid I)$$

4. Use $\nabla P = P\nabla \log P$ to differentiate expected payoff

$$\nabla_\theta J(\theta) = \sum\nolimits_\tau P_\theta(\tau)R(\tau)s_\theta(\tau) = \mathbb{E}_{\tau\sim P_\theta}[R(\tau)\, s_\theta(\tau)]$$

REINFORCE: terminal return × trajectory score

$$\tau \sim P_\theta, \qquad \widehat{g}_\theta(\tau) = R(\tau)s_\theta(\tau), \qquad \mathbb{E}[\widehat{g}_\theta] = \nabla_\theta J(\theta)$$

Unbiased, but one noisy terminal return weights every decision's score.

Finite perfect-recall game; positive policy probabilities; opponent and chance held fixed. Williams (1992); Srinivasan et al. (2018), §§2.2, 4 and Appendix E.

Policy gradients with a learned critic

REINFORCE is correct but noisy. We can use standard ideas to reduce variance

From REINFORCE to Actor-Critic

$$Q_\theta(I, a) := \mathbb{E}_{\tau \sim P_\theta}[R(\tau) \mid I \text{ reached}, \ a \text{ chosen}]$$

$$V_\theta(I) := \sum\nolimits_b \pi_\theta(b \mid I) Q_\theta(I, b), \qquad \boxed{A_\theta(I, a) := Q_\theta(I, a) - V_\theta(I)}$$

So we can write

The same expected gradient, using action advantages

$$\nabla_\theta J = \mathbb{E}_{\tau \sim P_\theta}\left[\sum\nolimits_{(I,a) \in \sigma(\tau)} A_\theta(I, a) \nabla_\theta \log \pi_\theta(a \mid I)\right]$$

We also learn an estimate of A_θ (directly or via V/Q): the **critic**. The policy π_θ is the **actor**.

Srinivasan et al. (2018), §4; Sutton & Barto, Ch. 13; GAE; Ataraxos: Methods and Supplementary §S3.4.

Summary: the RL toolbox

Behavioral policies let us tap into the RL toolbox

Behavioral strategies make path probabilities products. We can then use scores (log probs) to obtain a nice estimator and tap into standard RL solutions

We can then use TD(λ)/GAE to "blend" multiple estimators and try to reduce variance

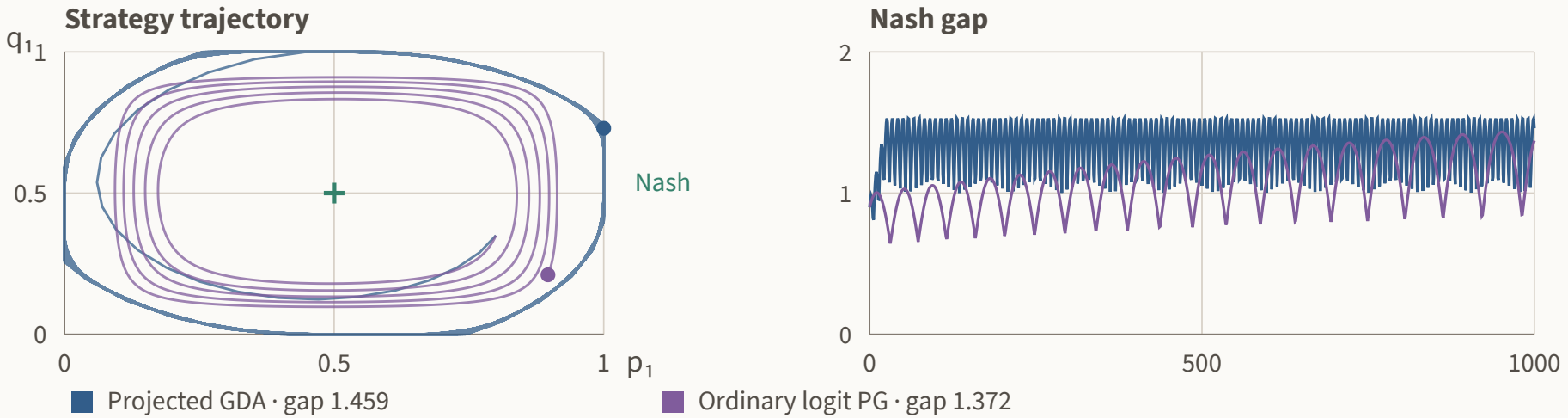
We can think of the resulting algorithm as a form of **CFR with Q values in place of counterfactual values**.

Perhaps reassuring us that we have not strayed too far...

However, both players are updating. Will their current policies converge?

Sutton & Barto (2018), Chs. 6, 12; Schulman et al. (2016), GAE; Srinivasan et al. (2018), §4 and Appendix E.

Exact gradients can still cycle



Simultaneous updates from $p_1 = 0.80$, $q_1 = 0.35$. Exact Q values; no sampling noise. Gap $= |2p_1 - 1| + |2q_1 - 1|$.

Even exact tabular methods can cycle (it is the average strategies that converge to the set of equilibria)

Matching pennies, $U = \left(\begin{smallmatrix} 1 & -1 \\ -1 & 1 \end{smallmatrix}\right)$. Exact gradients with simultaneous updates.

Choice III: add regularization to force convergence

Add attraction toward a full-support reference policy ρ.

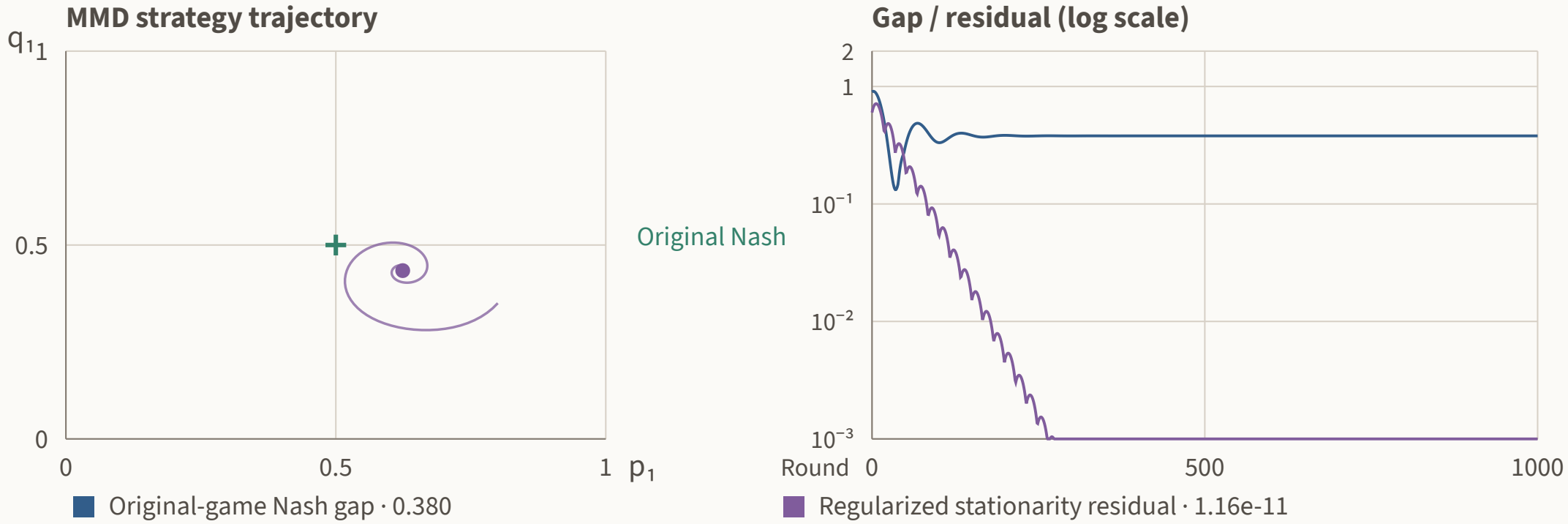
$$\max_{p} \min_{q} \; p^{\mathsf{T}} U q - \alpha \, \mathrm{KL}(p \| \rho_1) + \alpha \, \mathrm{KL}(q \| \rho_2)$$

Annealing

A fixed positive α induces a regularized equilibrium. To approach the original game's equilibrium, a popular approach is to reduce regularization ("anneal") over time

Sokota et al. (2023), Magnetic Mirror Descent; extensive-form entropy needs the corresponding game-specific analysis.

What changes when we add a magnet?



Exact matching-pennies example with a biased, fixed reference policy. A fixed positive KL weight changes the equilibrium we converge to.

Summary: three choices for self-play

A scalable recipe

Choice I: behavioral policies and logits.

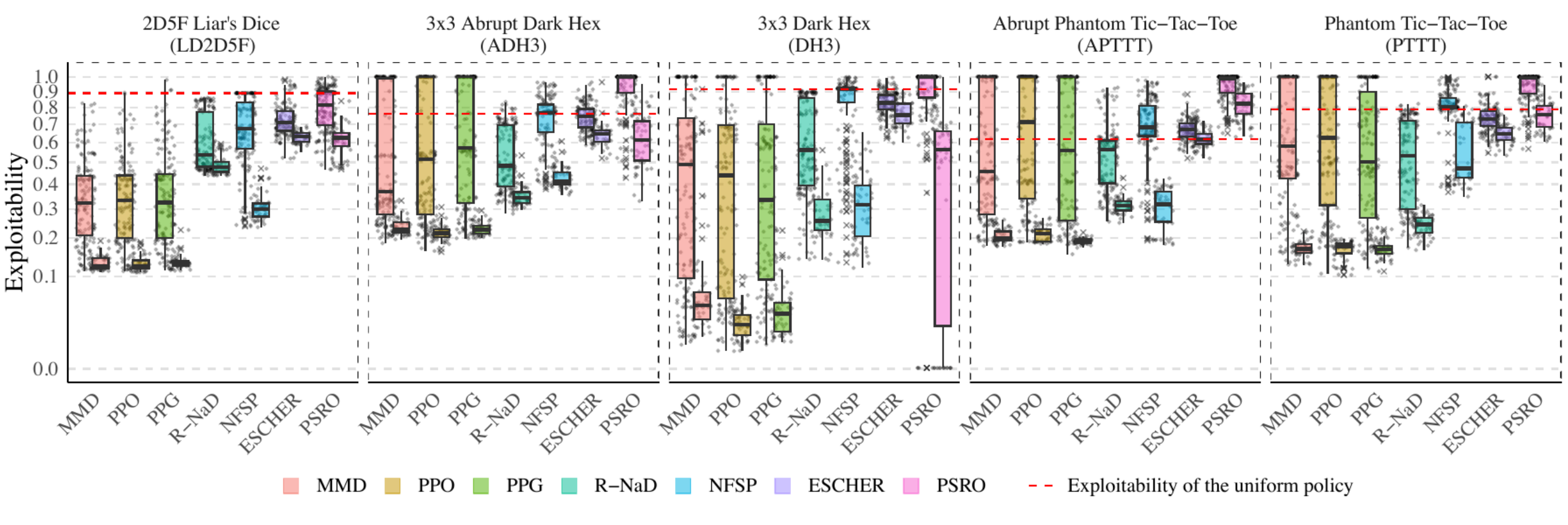
Choice II: on-policy values and policy-gradient estimates.

Choice III: regularization to stabilize self-play; reduce its bias with a controlled schedule.

How well does this work in practice?

Do the resulting policies have low exploitability?

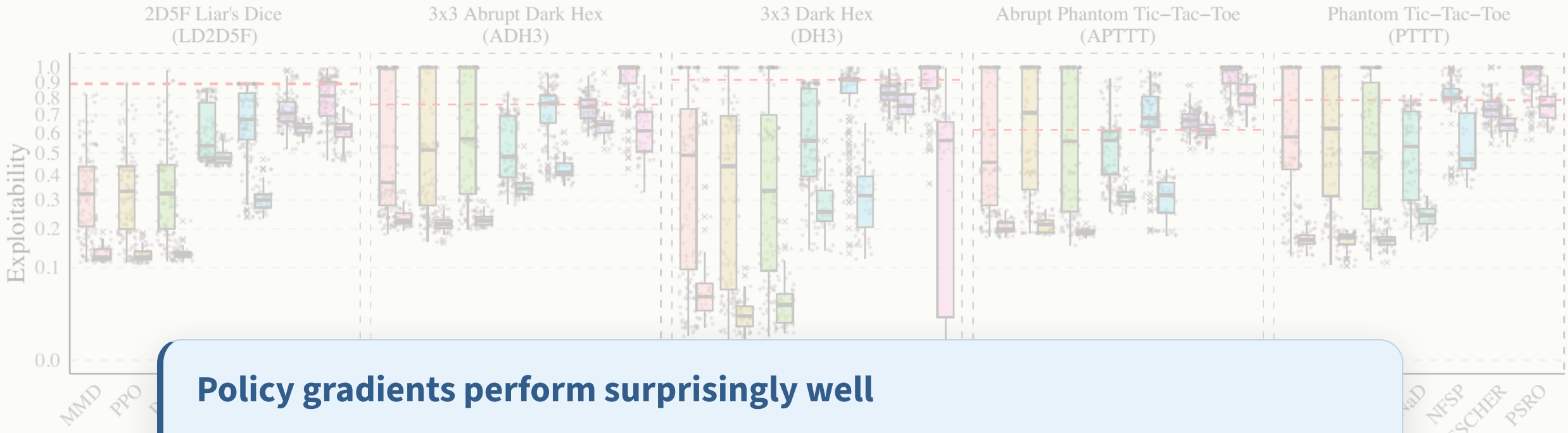
Reevaluating policy gradient methods



Lower exploitability is better. Boxes: tuning / evaluation. Axis: square-root scale.

Rudolph et al., Reevaluating Policy Gradient Methods for Imperfect-Information Games (2026). ICLR 2026, arXiv v4, 27 May 2026.

Reevaluating policy gradient methods



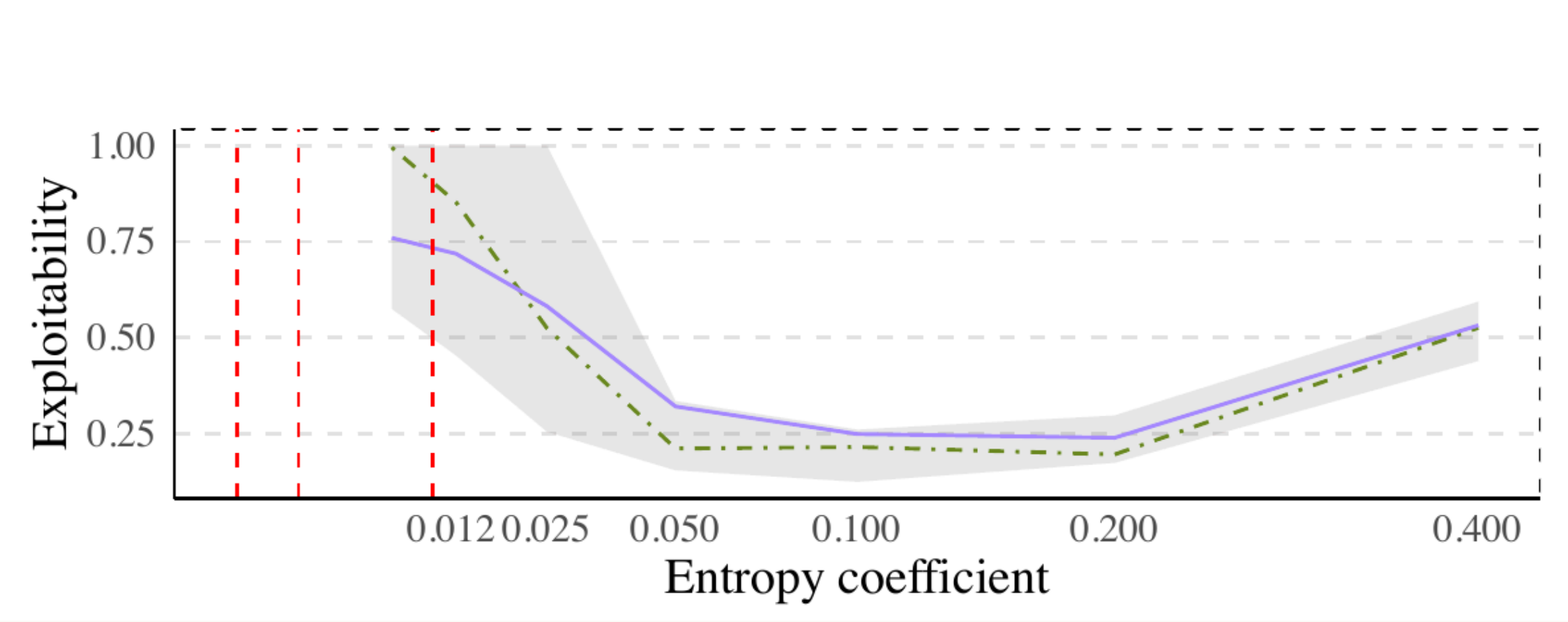
Policy gradients perform surprisingly well

Across 7,000 runs on five benchmarks, MMD, PPO, and PPG matched or exceeded the tested game-theoretic deep RL methods.

Lower exploitability is better. Boxes: tuning / evaluation. Axis: square-root scale.

Rudolph et al., Reevaluating Policy Gradient Methods for Imperfect-Information Games (2026). ICLR 2026, arXiv v4, 27 May 2026.

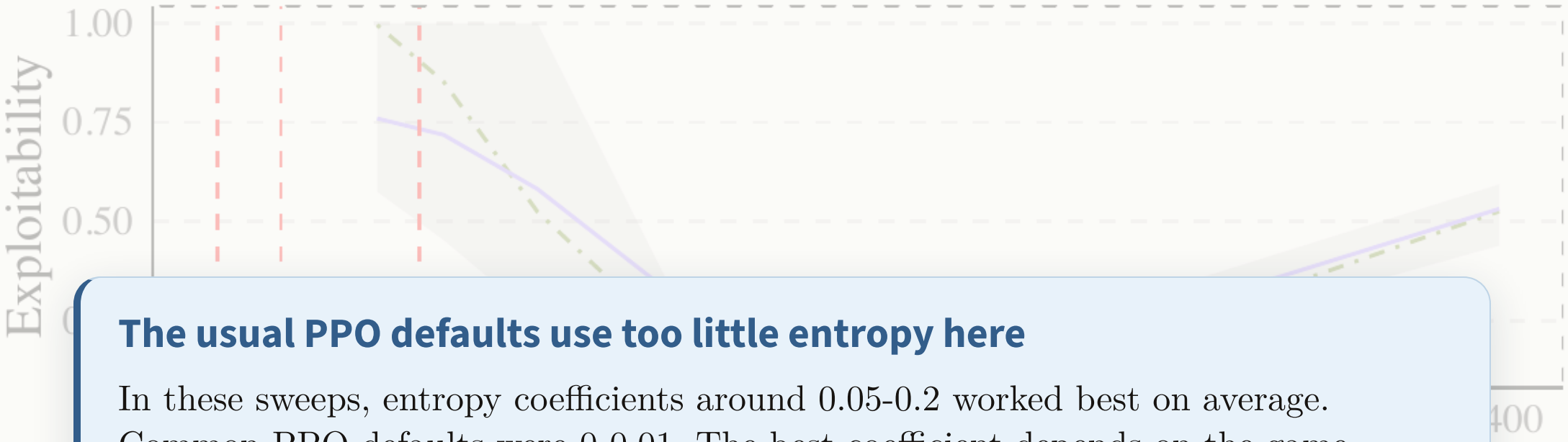
How much entropy regularization?



Pooled over 2D5F Liar's Dice, 3×3 Dark Hex, 3×3 Abrupt Dark Hex,
Phantom Tic-Tac-Toe and Abrupt Phantom Tic-Tac-Toe. Green: mean; purple: median; gray: interquartile range.

Rudolph et al. (2026), Fig. 7. Red dashed lines: common PPO defaults.

How much entropy regularization?



The usual PPO defaults use too little entropy here

In these sweeps, entropy coefficients around 0.05-0.2 worked best on average. Common PPO defaults were 0-0.01. The best coefficient depends on the game.

Pooled over 2D5F Liar's Dice, 3×3 Dark Hex, 3×3 Abrupt Dark Hex,
Phantom Tic-Tac-Toe and Abrupt Phantom Tic-Tac-Toe. Green: mean; purple: median; gray: interquartile range.

Rudolph et al. (2026), Fig. 7. Red dashed lines: common PPO defaults.

Decision-time planning (DTP)

Can we spend more computation at decision time to improve the policy?

Decision-Time Planning

Blueprint policy		DTP		Improved policy
Start from the trained network	→	Spend extra simulation to refine policy	→	Then play the new policy

Update-Equivalent Search [SFW+24]

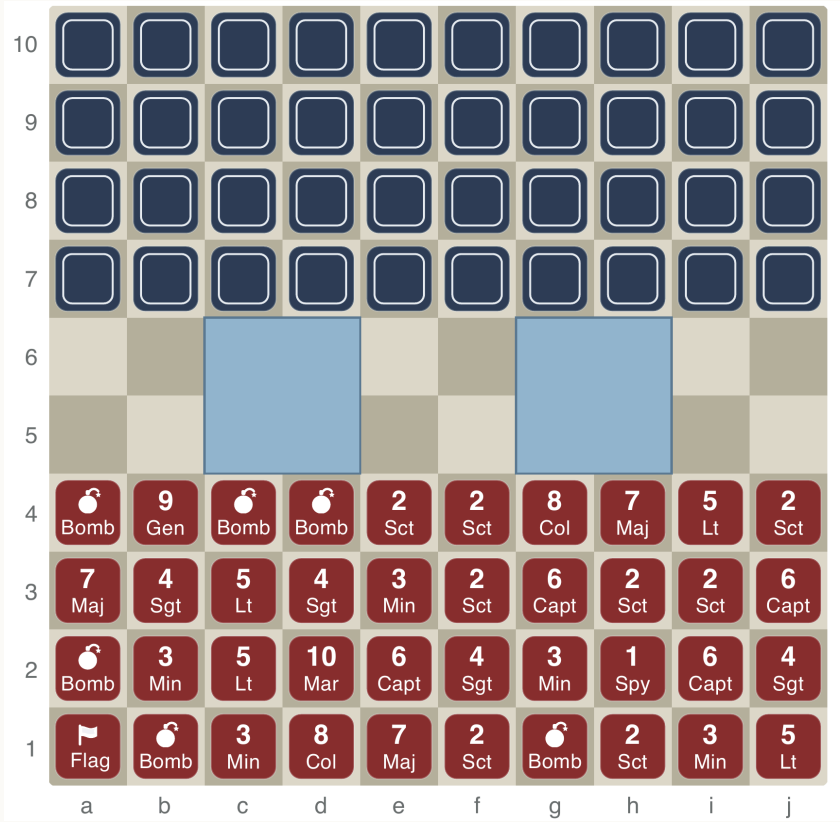
Instead of using gadgets and subgame solving, do something very simple: spend the DTP budget to estimate higher quality Q-values (starting from the *posterior distribution* of the state), and then do one last policy gradient step

We will see an evaluation of this idea in Stratego.

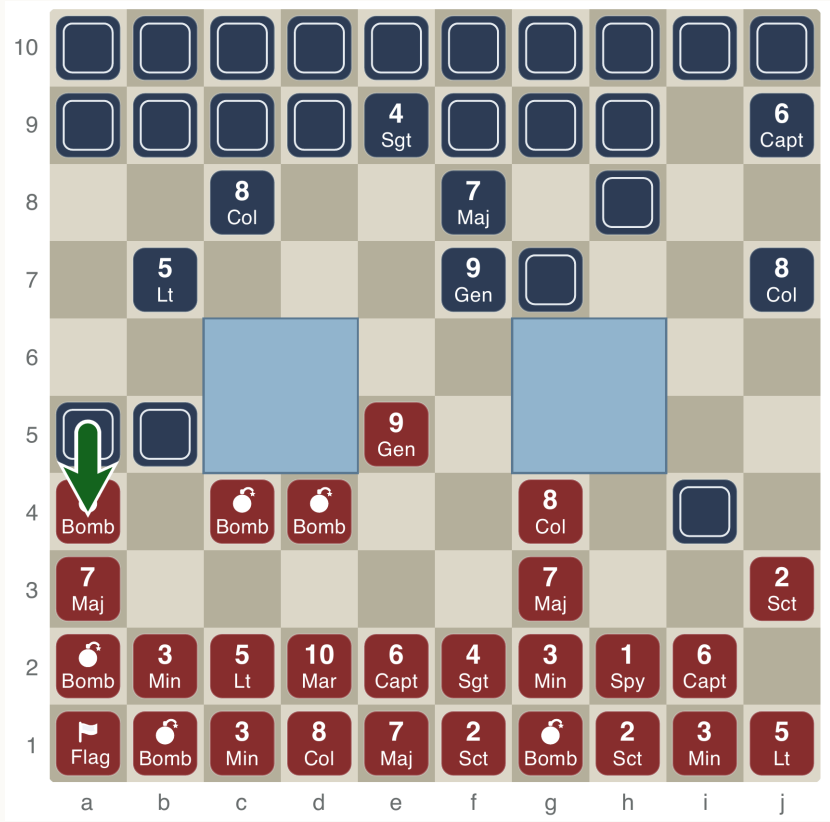
Sokota et al. (ICLR 2024), update-equivalence; Sokota et al., Nature (2026).

Stratego

Hidden pieces, long horizons



a. Secret setups. Each player knows its own pieces.



b. A battle reveals both piece types and captures at least one.

Sokota et al., Nature (2026), Fig. 2a–b.

Stratego before Ataraxos

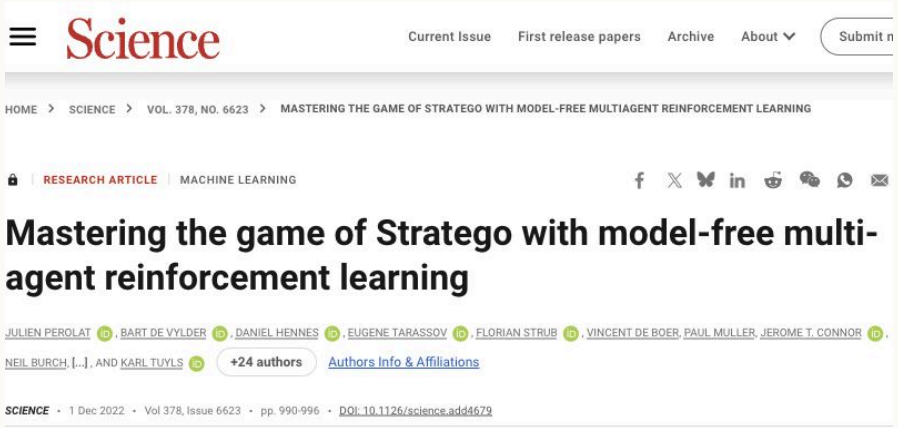
Before 2022, Stratego was one of the last iconic board games without expert-level AI.

2022: DeepMind's DeepNash. R-NaD and self-play, without DTP. 42 wins in 50 ranked Gravon games (84%).

2023: World Championship demo. 19 wins, 9 losses. It lost to most of its highest-ranked opponents, including Pim Niemeijer.

A large compute commitment

1,024 TPU nodes for roughly 2–3 months.
Estimated training-run cost: **US$3–4.5 million** at 2025 prices.



DeepMind's DeepNash. Science, 2022.

Perolat et al., Science (2022); Sokota et al., Nature (2026), Contextualization vis-à-vis DeepNash.
Cost: TPU v3 rental estimate based on an author's recollection; actual expenditure and total R&D budget were not reported.

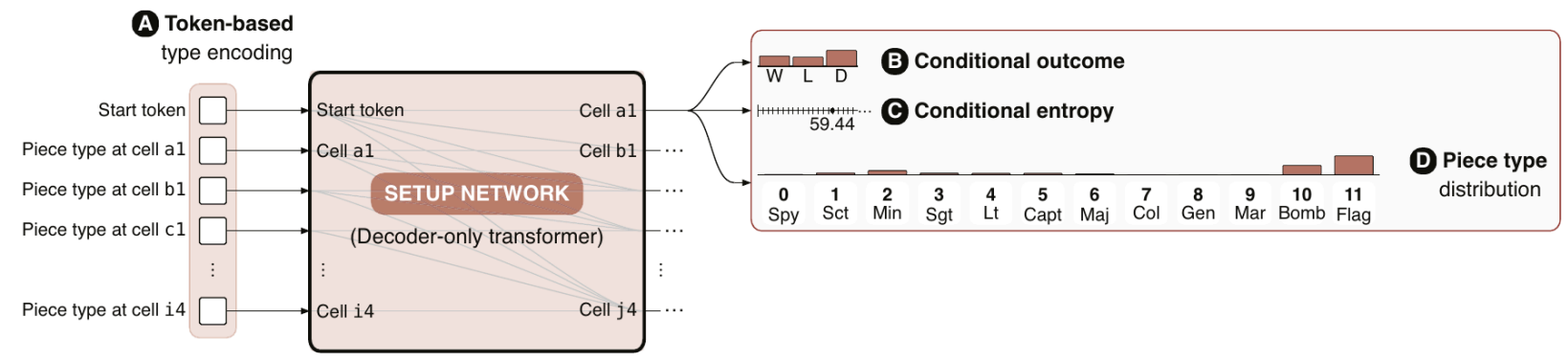
- **Ataraxos (2026)**



Self-play policy gradients + DTP against top humans.

Sokota et al., Nature (2026).

Ataraxos learns where to put the pieces



Extended Data Fig. 1: setup policy.

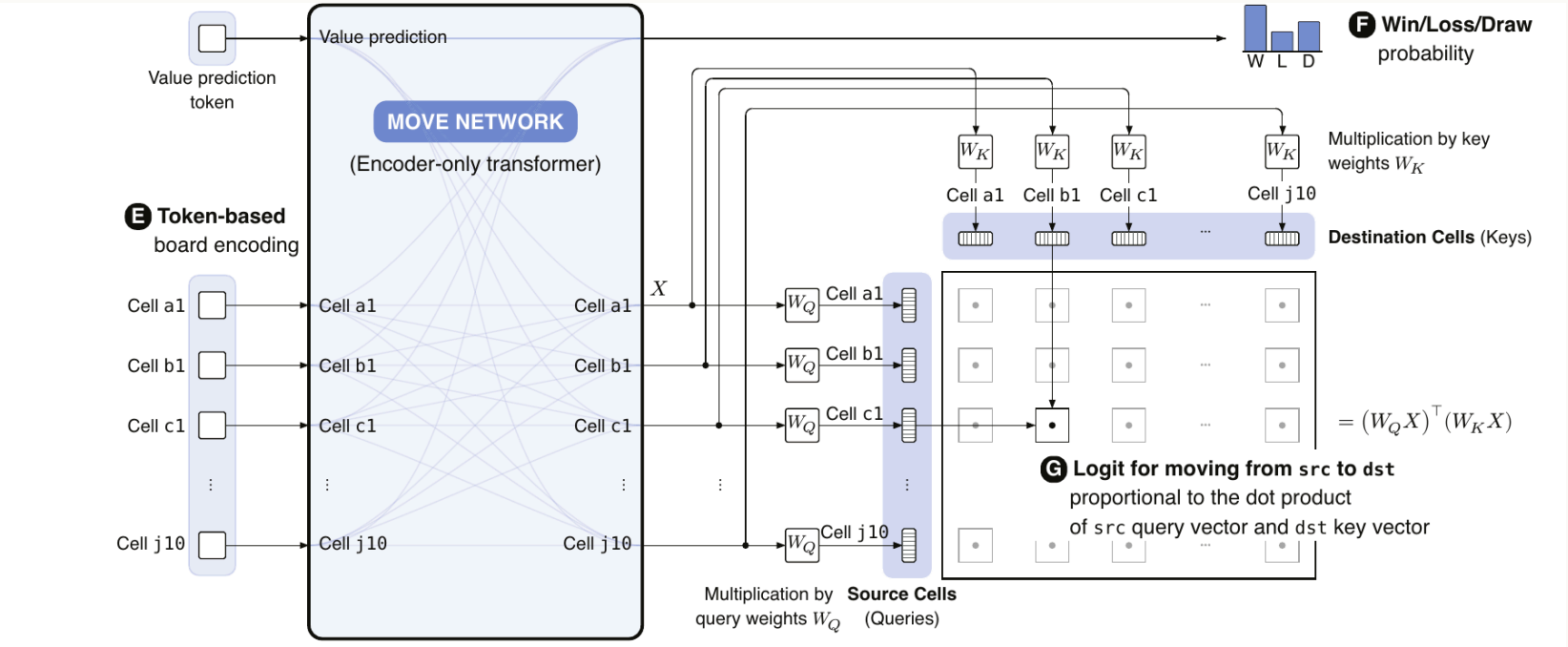
A policy over entire setups

Place 40 pieces autoregressively. Predict outcomes and entropy while learning the setup distribution.

Two self-play processes learn together: setups determine the initial boards; move policies determine the outcomes used to train both.

Sokota et al., Nature (2026), Methods and Extended Data Fig. 1.

Ataraxos learns how to move



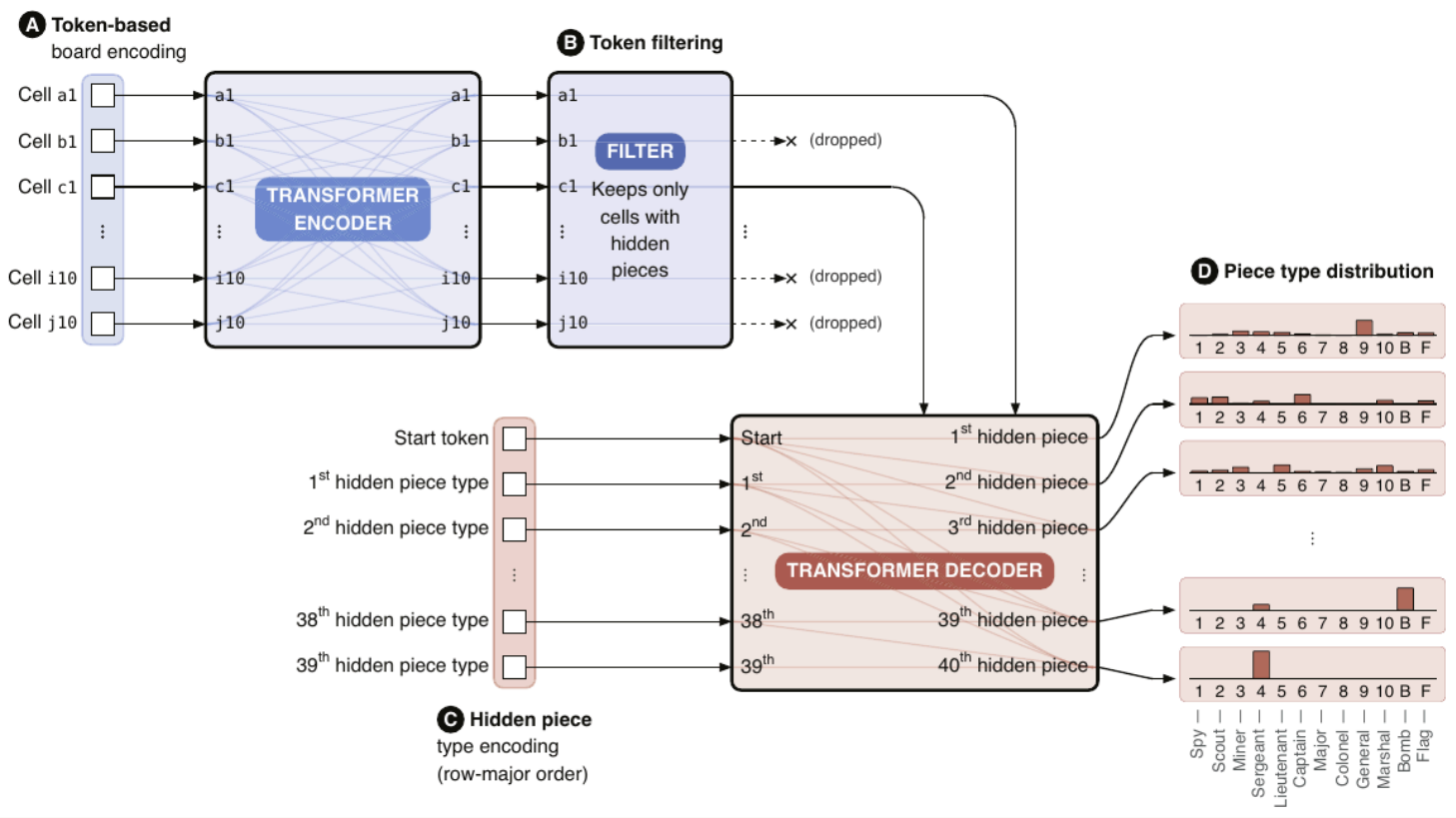
Extended Data Fig. 1, bottom. A source–destination dot product defines each move logit.

A scalable behavioral policy

Board features become move logits and value estimates. Damped self-play trains the network; DTP refines its policy at test time.

Sokota et al., Nature (2026), Extended Data Fig. 1.

Ataraxos learns what the hidden pieces might be



Extended Data Fig. 4: the belief network.

Encode the player's information state. Decode a distribution over hidden piece types **one at a time**.

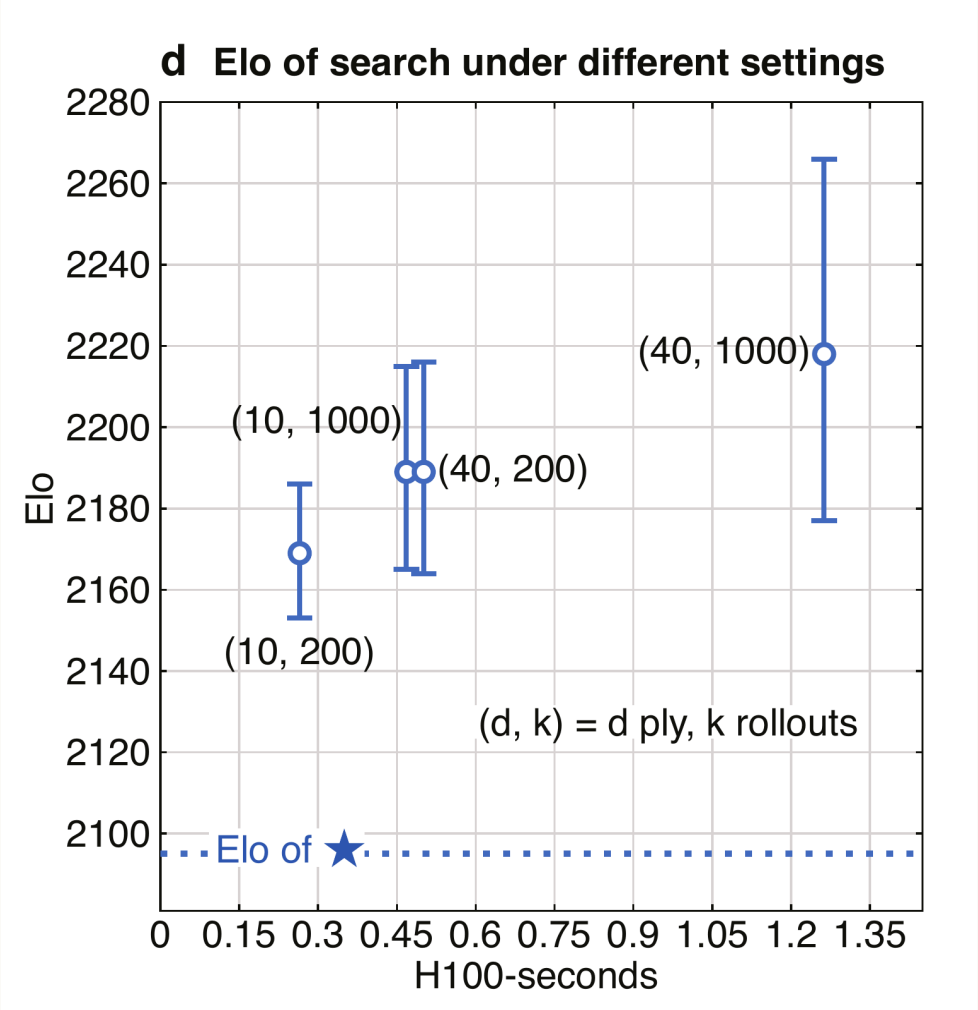
Train on final-policy self-play, using true hidden piece types as supervised labels.

From beliefs to DTP

Sample plausible hidden boards. Average rollout values for each candidate move.

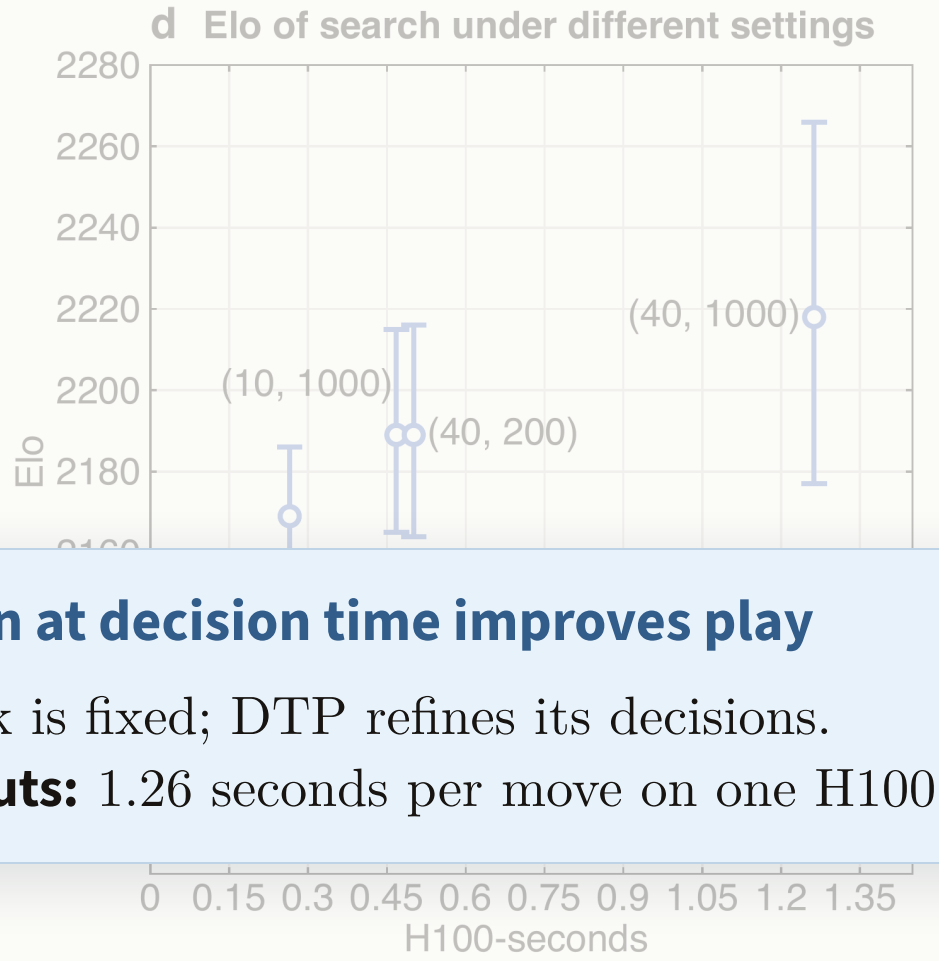
Sokota et al., Nature (2026), Methods: belief modelling and DTP; Extended Data Fig. 4.

Ataraxos: decision-time planning



Sokota et al., Nature (2026), Extended Data Fig. 2d. Bars: 95% confidence intervals. Dotted line: move network without DTP.

Ataraxos: decision-time planning



More computation at decision time improves play

The policy network is fixed; DTP refines its decisions.
40 ply · 1,000 rollouts: 1.26 seconds per move on one H100.

Sokota et al., Nature (2026), Extended Data Fig. 2d. Bars: 95% confidence intervals. Dotted line: move network without DTP.

How strong is Ataraxos?

Pim Niemeijer: the most decorated Stratego player

4 world championships · **15** Dutch championships · **2** online world championships

More than **600 weeks** ranked world #1.

85%

Effective win rate against Pim Niemeijer

15 wins · 1 loss · 4 draws

95%

World Championship attendee demo

38 wins · 2 losses · 0 draws

1.26 s

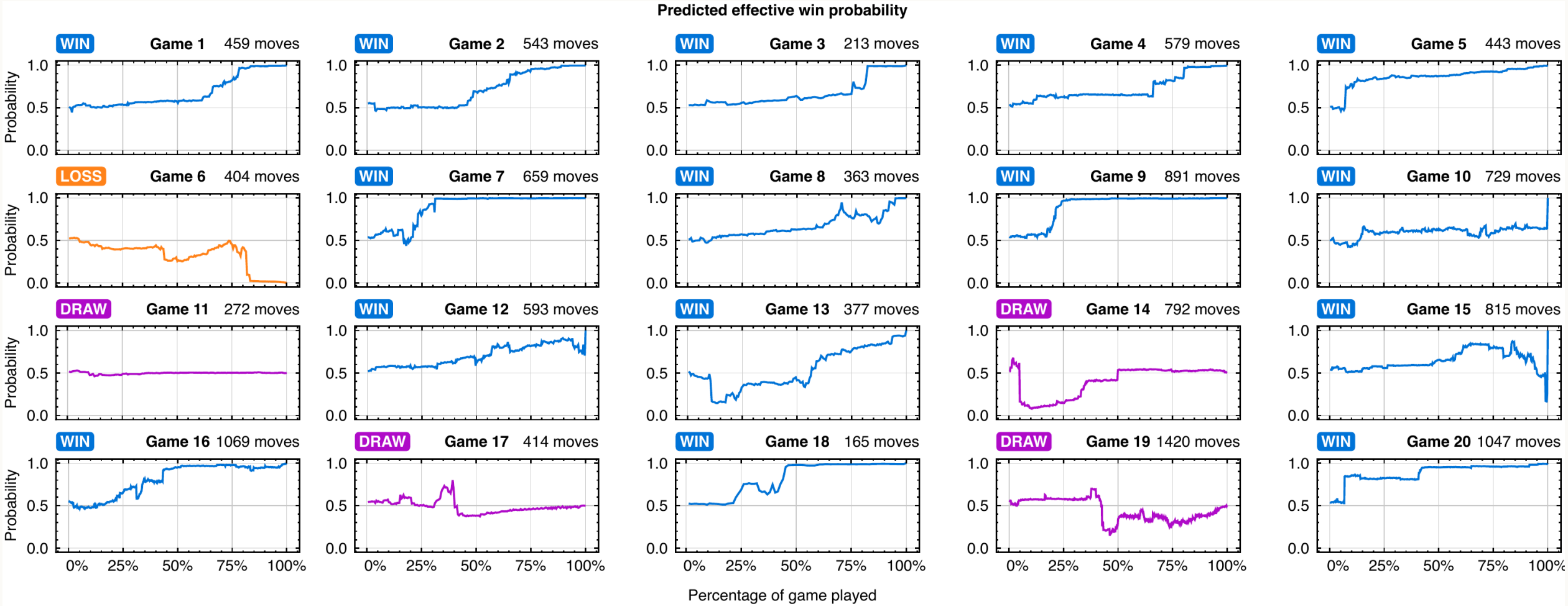
Average time per move on one H100

40-ply · 1,000-rollout DTP

Across the 20-game evaluation, Pim could adapt. Ataraxos kept its strategy fixed.

Accolades and evaluation: Sokota et al., Nature (2026), Evaluation and Extended Data Fig. 2; International Stratego Rating: Pim Niemeijer.

Recovering from difficult positions



Critic predictions over all 20 games.

Sokota et al., Nature (2026), Fig. 3 and Methods: Play Style. Original figure reflowed to 5 × 4 panels.

How much did training cost?

1/500

Approximate compute cost

1/30

Approximate self-play games

1/100

Approximate training examples

Ataraxos relative to DeepMind's DeepNash, as estimated in the paper. Ataraxos trained its RL networks on 16 H100s for one week and its belief network on 4 H100s for four days.

The improvement is also in sample efficiency

Far fewer games and training examples. The systems were evaluated against humans; no direct match was possible.

Sokota et al., Nature (2026), training-cost comparison; Perolat et al. (2022). Compute costs use 2025 price estimates.

Further tricks

More stable advantage estimates (VRPO)

MDPs

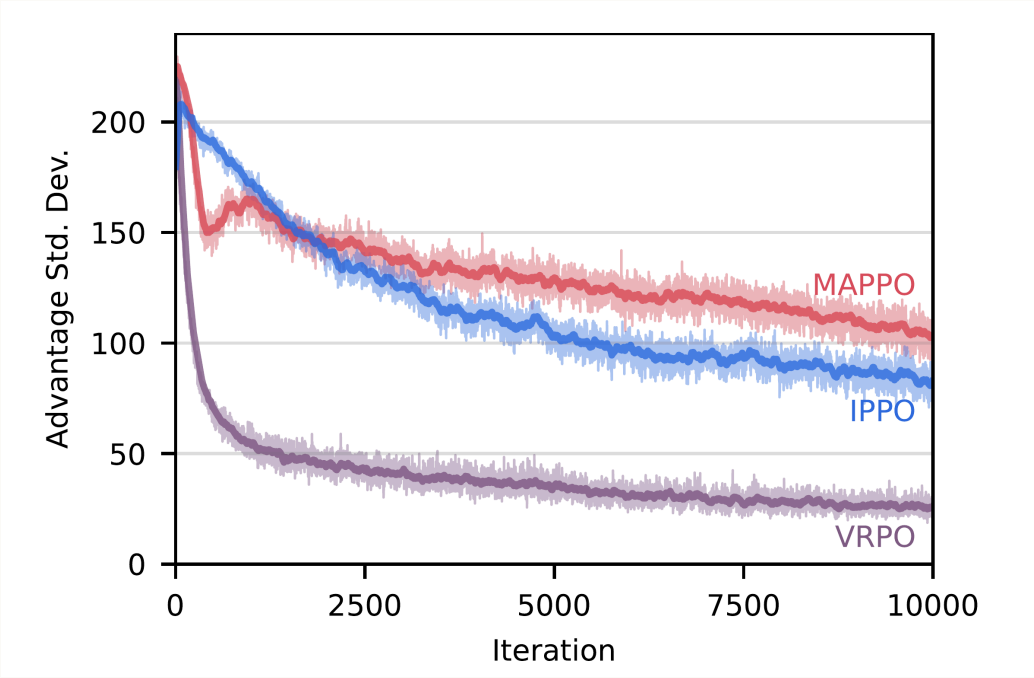
A **deterministic** optimal policy exists.

Imperfect-information games

Equilibrium may **require randomization**.

Baseline corrections can reduce variance

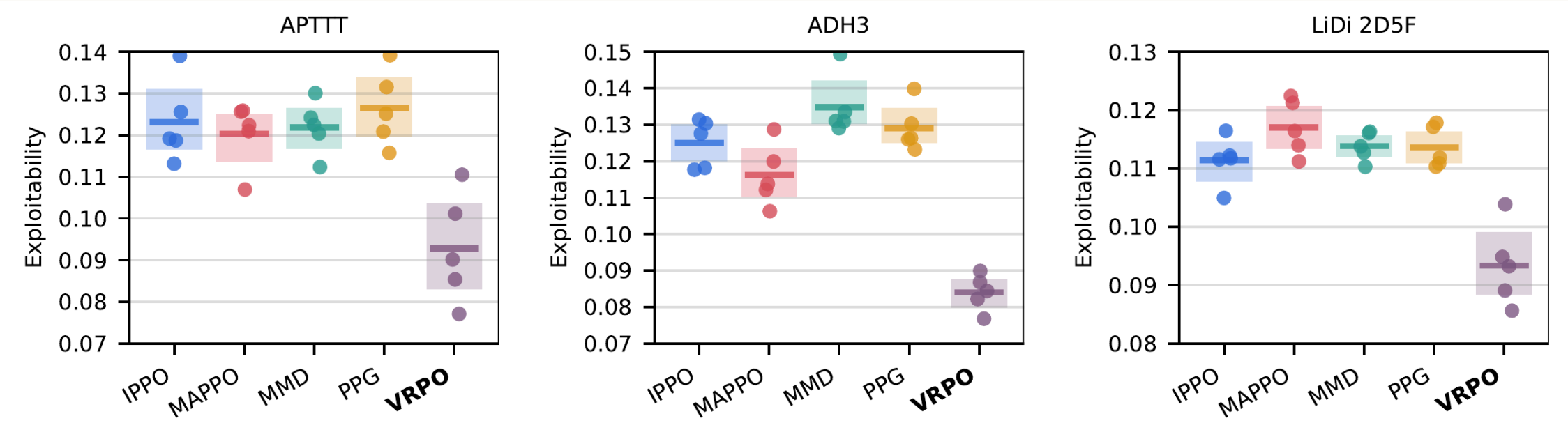
Action-sampling noise can persist at equilibrium. Correct for it using learned action values.



Poker training: standard deviation of the advantage estimates. Lower is better.

Fan & Farina (2026), GAE Falls Short in Imperfect-Information Self-Play Reinforcement Learning. Fig. 4.

VRPO: empirical results



Exact exploitability: lower is better. Shared training schedule; five seeds per configuration.

More than 15% lower exploitability

Across all three benchmarks, compared with every baseline shown.

Dou Dizhu: beats PerfectDou with a comparable training budget.

Fan & Farina (2026), GAE Falls Short in Imperfect-Information Self-Play Reinforcement Learning. Fig. 2; §5.2.

Exponential moving average of parameters

Keep a smoothed copy of the policy network.

$$\bar{\theta}_t = \beta \bar{\theta}_{t-1} + (1-\beta)\theta_t, \qquad 0 < \beta < 1$$

Evaluate the smoothed network

Ataraxos used $\beta = 0.999$ for setup and move networks.

The map from parameters to policies is nonlinear: averaging parameters is generally different from averaging strategies.

Sokota et al., Nature (2026), Supplementary Information, setup and move training.

Conclusions

Main takeaways

(1) Policy gradient methods in imperfect-information games

can be put on solid theoretical footing, and seem to perform very well

(2) Entropy plays an important role

Compared to vanilla PPO, you need more. Other imperfect-information-specific tricks seem to help, for example Q-boosting ("GAE falls short" paper)

(3) DTP plays well with policy gradient

One approach that we found useful is just to perform one additional tabular update on the policy using high-quality Q values ("Update-equivalent search" used by Ataraxos)

Reevaluating Policy Gradient; Ataraxos; GAE falls short.