CSCE 631: Module 9, Lecture 1

Game-Playing AI at Scale:
From Self-Play to Leagues to Language

Fall 2026 · Alan Kuhnle · Texas A&M University

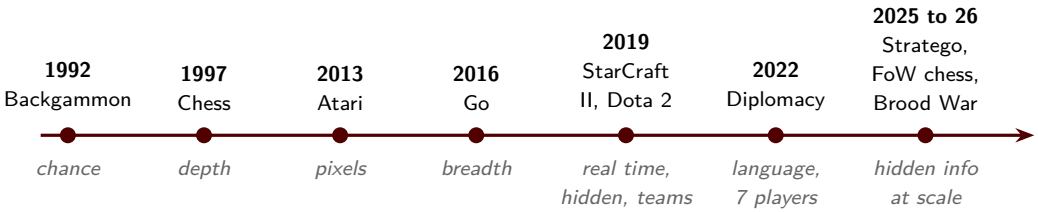
AlphaZero, AlphaStar, OpenAI Five, Cicero, and Pluto, read as game-theory algorithms run at enormous scale.

Why Games Keep Being the Benchmark

And what changed when the games got real time, hidden, and crowded

Games as AI's Yardstick, 1992 to 2026

Hook ▷ Each milestone removed one simplifying assumption



- Games give fixed rules, an unambiguous score, and human experts to compare against.
- Backgammon, chess, and Go are *two-player zero-sum* (2p0s). Atari is single-agent; Dota 2 is two teams; Diplomacy is seven-player general-sum.

Source: Berner et al. (2019), §1 (1992 to 2016); later slides for 2019 to 2026; Stratego and Fog of War chess in the notes.

What Makes the Newer Games Hard

Hook ▷ Five properties, four games

	Go	StarCraft II	Dota 2	Diplomacy
Information	perfect	hidden (fog, camera)	hidden (fog of war)	private talk
Timing	alternating	real time	real time	simultaneous
Actions per step	≈ 250	$\approx 10^{26}$	8,000 to 80,000	orders plus language
Players	2	2	2 teams of 5	7
Payoffs	zero-sum	zero-sum, cyclic	team zero-sum	general-sum

Nontransitivity. A beats B, B beats C, C beats A. Then "train against your latest self" can circle without improving.

Source: Silver et al. (2016) p. 484; Vinyals et al. (2019) pp. 1 to 2; Berner et al. (2019) §2; FAIR et al. (2022) p. 1067.

Today: Learning Objectives and Plan

By the end you should be able to:

1. explain why self-play tends to work in 2p0s perfect-information games;
2. compute prioritized fictitious self-play (PFSP) probabilities;
3. distinguish average-case strength from exploitability;
4. state the piKL objective and why Cicero does not just play an equilibrium.

Part	Topic (minutes)	Recall from
	Why games; poker recap (6)	CFR (M5)
I	AlphaGo and AlphaZero (9)	minimax, backward induction (M2, M4)
II	AlphaStar: PFSP and the league (18)	fictitious play (M3), PSRO (M7)
III	OpenAI Five: teams, scale, surgery (10)	team games
IV	Exploitability: attacking KataGo (7)	Nash, maxmin (M2)
V	Cicero: general-sum, language, piKL (11)	cheap talk (M7), monitoring (M8)
VI	Pluto (2026); synthesis (9)	bandits (M3)
	Buffer: questions or one discussion prompt (5)	

Recall: Where Poker Left Us

Bridge ▷ Module 5 (CFR); the summer poker case study

Libratus (heads-up no-limit, 2p0s)

- Blueprint from a CFR-family solve of an abstraction.
- Nested subgame solving during play.
- Beat four top specialists over 120,000 hands.

Pluribus (six-player no-limit)

- Blueprint by Monte Carlo CFR self-play: 8 days, one 64-core server.
- *No guarantee* of a Nash equilibrium outside 2p0s; the goal was empirical strength.

The thread for today. In 2p0s games a Nash equilibrium is a target with a guarantee: it is unexploitable in expectation. Today's games stretch that target: too large to solve, cyclic, team-based, and finally general-sum with talk.

Source: Brown & Sandholm (2018) pp. 418 to 419; Brown & Sandholm (2019) pp. 885 to 887.

Part I: AlphaGo and AlphaZero

Search plus learned intuition, improved by playing yourself

AlphaGo (2016): Two Networks Inside a Tree Search

Part I ▷ Cut the tree's breadth with a policy, its depth with a value

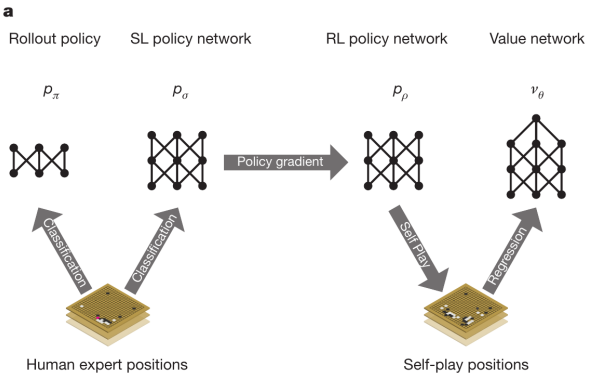
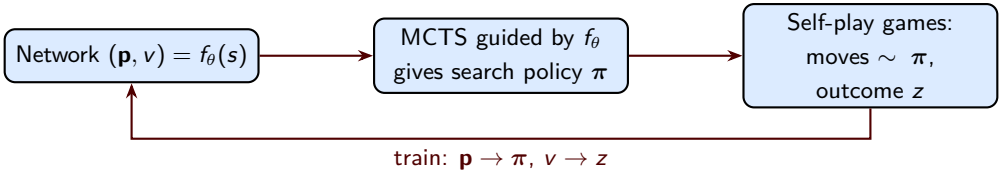


Figure: Silver et al., Nature 2016, Fig. 1a.

Source: Silver et al. (2016), Nature 529, abstract, pp. 484 to 485.

- p_σ imitates expert moves.
- p_ρ improves p_σ by policy gradient in self-play against earlier versions.
- $v_\theta(s)$ predicts the winner of p_ρ's games from position s.
- Monte Carlo tree search (MCTS) uses them at play time.
- Beat the European champion 5 to 0, the first program to beat a human professional on a full board.

AlphaGo Zero and AlphaZero: One Network, No Human Data



Loss. $\ell = (z - v)^2 - \boldsymbol{\pi}^\top \log \mathbf{p} + c\|\theta\|^2$, with outcome $z \in \{-1, 0, +1\}$ and L_2 weight c. Search makes $\boldsymbol{\pi}$ stronger than $\mathbf{p}$; fitting $\mathbf{p}$ to it is a policy-improvement step.

- **AlphaGo Zero (2017):** no human data; beat the published AlphaGo 100 games to 0.
- **AlphaZero (2018):** chess, shogi, and Go; superhuman within 24 hours.
- **MuZero (2020)** learns the model too, so search no longer needs the rules (notes).

Source: Silver et al. (2017) pp. 1, 3, Eq. 1; Silver et al. (2018) and Schrittwieser et al. (2020), abstracts.

Why Self-Play Tends to Work Here, and What It Assumes

Three facts from earlier modules.

1. *Zero-sum:* all equilibria share one value and are interchangeable (M2), so there is one well-defined target.
2. *Perfect information:* backward induction gives each position a value $v^*(s)$ (M4), independent of how play reached it.
3. *Search improves the policy:* the AlphaZero loop is approximate policy iteration; the exact version has v^* as its fixed point.

Caveat. A target exists; that does not mean a given learner reaches it. There is no convergence theorem for these deep networks, naive self-play "may chase cycles," and in Part IV a superhuman AlphaZero-style Go program loses to a weak adversary.

Pause and predict. Which of the three facts fail in StarCraft II? In Diplomacy?

Source: Course material (M2, M4); Silver et al. (2017) p. 3; Vinyals et al. (2019) p. 3.

Part II: AlphaStar and the League

When self-play chases its tail, train against a population

AlphaStar (Nature 2019): The Game and the Interface



Figure: DeepMind blog, Jan. 2019: AlphaStar vs. TLO, Dec. 2018, an earlier agent than the Nature 2019 one.

Full 1v1 StarCraft II, all three races, imperfect information. Constraints set with professional players:

- *Camera:* detail only inside a movable view.
- *Action rate:* at most 22 non-duplicate actions per 5 s, enforced by a monitoring layer (an M8-style monitor).
- *Delay:* about 110 ms from observation to action.

Training: imitate 971,000 human replays, then reinforcement learning in a league.

Source: Vinyals et al. (2019), author manuscript pp. 1 to 4; Methods pp. 11, 13.

Why Plain Self-Play Is Not Enough: Cycles

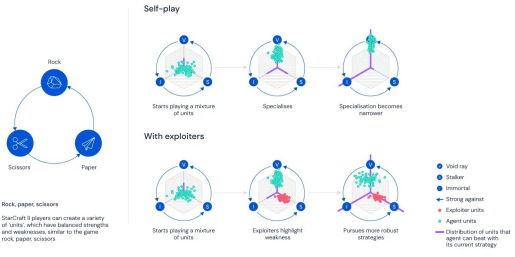


Figure: DeepMind blog, Oct. 2019, Fig. 1.

Self-play (SP). Best-respond to your latest self. On a cycle it circles without progress.

Fictitious self-play (FSP). Best-respond to the uniform mixture of all past policies. With *exact* best responses the mixture converges to a Nash equilibrium in 2p0s games (M3); deep RL only approximates this.

The catch. FSP wastes games on long-beaten opponents.

Pause and predict. In rock-paper-scissors, what does SP's sequence do? What does FSP's *average* policy converge to?

Source: Vinyals et al. (2019), p. 3 and Methods p. 15.

Prioritized Fictitious Self-Play (PFSP)

Learner A samples frozen opponent B from candidate set $\mathcal{C}$ with

$$\Pr[B] = \frac{f(x_B)}{\sum_{C \in \mathcal{C}} f(x_C)}, \qquad x_B = \Pr[A \text{ beats } B], \qquad f : [0,1] \to [0,\infty),$$

assuming some candidate has $f(x_C) > 0$. $f_{\text{hard}}(x) = (1-x)^p$ $(p > 0)$ favors opponents you lose to; $f_{\text{var}}(x) = x(1-x)$ favors even matches; uniform FSP is $f \equiv 1$.

Worked example. A beats B_1, B_2, B_3 with probability 0.9, 0.5, 0.2.

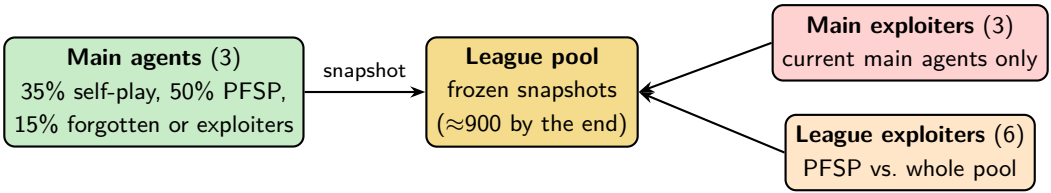
Weighting	raw weights $f(x)$	B_1	B_2	B_3
f_{hard}, $p = 1$	0.1, 0.5, 0.8	0.071	0.357	0.571
f_{hard}, $p = 2$	0.01, 0.25, 0.64	0.011	0.278	0.711
f_{var}	0.09, 0.25, 0.16	0.18	0.50	0.32

Reading. As $p \to \infty$, all weight moves to the opponent(s) you beat least often: a "smooth approximation of max-min."

Source: Vinyals et al. (2019), Methods pp. 15 to 16; example computed for this lecture.

The League: Three Kinds of Players

Part II ▷ Vinyals et al. 2019, Fig. 1C and Methods



- **Main agents** are the product; they never reset.
- **Main exploiters** expose weaknesses of the *current* main agents, then reset.
- **League exploiters** find "global blind spots": strategies no league player beats.

12 learners (per race: one main agent, one main exploiter, two league exploiters), each on 32 TPUs for 44 days.

Source: Vinyals et al. (2019) p. 4; Methods pp. 16 to 17.

The League as PSRO in Practice

Part II ▷ Recall Module 7, Lecture 2: double oracle and PSRO

PSRO loop (Lanctot et al. 2017)

1. Policy sets Π_i; payoff table U^{Π} by simulation.
2. Meta-solver: mixture σ over Π from U^{Π}.
3. Oracle: train π_i' by RL vs. $\pi_{-i} \sim \sigma_{-i}$.
4. Add π_i' to Π_i; repeat.

Meta-solver σ_{-i}	**Gives**
Latest policy only	self-play
Uniform over past	fictitious play
Nash of meta-game	double oracle
PFSP + exploiters	AlphaStar's league

The AlphaStar paper: PSRO "is related to League training, although League training specifies unique targets for approximate best responses (i.e. PFSP and exploiters)."

Our reading. Exploiters are extra oracles aimed at the current main agents instead of the meta-strategy. PSRO allows this; it does not prescribe it.

Source: Lanctot et al. (2017), Alg. 1 and p. 4; Vinyals et al. (2019), Methods p. 13.

Real-World Games Look Like Spinning Tops

Part II ▷ Czarnecki et al., NeurIPS 2020

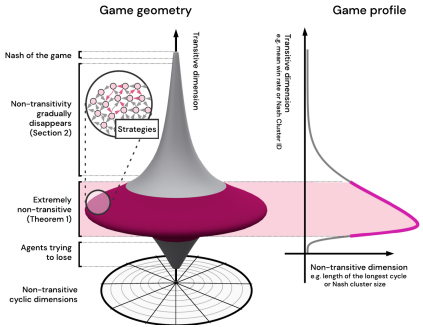


Figure: Czarnecki et al., NeurIPS 2020, Fig. 1.

- Strategies vary along a transitive axis (skill) and cyclic axes. Cycles are widest at middling skill.
- Go contains a cycle of length at least 2^{1000}.
- **Thm. 3:** if the population contains a full Nash cluster, a policy that beats every member is strictly stronger.
- Not universal: some games cycle for every population size.

So: a league *tries* to keep such a cluster in the training set; nothing guarantees it does. Exploiters add width, main agents add height.

Source: Czarnecki et al. (2020), Fig. 1, Prop. 1 (p. 5), Thm. 3 (p. 7), Table 1 caption (p. 8).

Measuring the League With Nash

Treat each league player as a pure strategy of a meta-game whose payoff matrix P is estimated from games between players.

- **Nash distribution over the league.** Solve the meta-game. AlphaStar's Nash put most weight on recent players, which the authors read as little forgetting or cycling.
- **Relative population performance (RPP).** For leagues A, B with payoff matrix P_{AB} and meta-game Nash $(\mathbf{p}_A, \mathbf{p}_B)$: $\mathrm{RPP} = \mathbf{p}_A^\top P_{AB}\, \mathbf{p}_B$. It compares two finite populations; it is not worst-case exploitability.
- **Cycles.** About 3,000,000 rock-paper-scissors cycles (each edge $\geq$70% wins) involve an exploiter; about 200 involve only main agents.

Reading. The exploiters carry the cycles; the main agents climb almost transitively.

Source: Vinyals et al. (2019) p. 4; Fig. 4 caption; Methods p. 19; Ext. Data Fig. 8 caption.

AlphaStar's Results

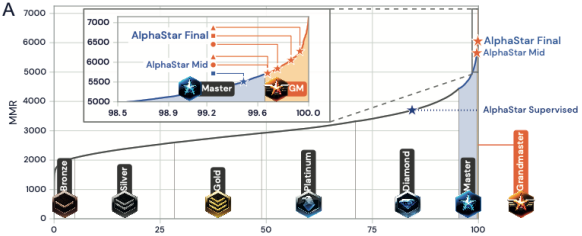


Figure: Vinyals et al., Nature 2019, Fig. 2A.

- MMR is Blizzard's Elo-like rating. AlphaStar Final: 6,275 / 6,048 / 5,835 (Protoss / Terran / Zerg), above 99.8% of ranked players.
- Ablation, test Elo: 1540 at the chosen action limit, 1145 at 10% of it, 1392 with *no* limit.

Pause and predict. Why might removing the action limit make the agent weaker?

Source: Vinyals et al. (2019) pp. 4 to 5; Fig. 3G; Methods pp. 13, 17.

Part III: OpenAI Five

Five players, one team, ten months of training

OpenAI Five: Mostly Self-Play, at Very Large Scale

Part III ▷ Berner et al. 2019 (arXiv preprint)



Figure: Berner et al., arXiv 2019, Fig. 11a: Finals vs. OG, game 1.

- Dota 2, 5v5, 17 of 117 heroes. One policy (about 159M parameters) controls each hero.
- Proximal Policy Optimization (PPO), an actor-critic method.
- **Latest self in 80% of games, older versions in 20%.** Like PFSP, the sampler down-weights opponents it already beats (notes).
- 180 days of training over 10 months.

Source: Berner et al. (2019), §§2 to 4 (pp. 2 to 8), Appendix N (Eq. 14).

Team Spirit: Tuning Cooperation Inside a Team

Two reward transformations.

1. *Zero-sum across teams:* subtract the enemies' average reward from each hero's reward.
2. *Team spirit* $\tau \in [0, 1]$: with raw rewards ρ_i and team mean $\bar{\rho}$,

$$r_i = (1 - \tau)\, \rho_i + \tau\, \bar{\rho}.$$

τ was annealed from 0.3 to 1.0: the goal is $\tau = 1$, but lower τ "reduces gradient variance in early training."

Worked example. Hero 1 earns 5, the others 0, so $\bar{\rho} = 1$.

	$\tau = 0$	$\tau = 0.3$	$\tau = 1$
Hero 1	5	3.8	1
Heroes 2 to 5	0	0.3	1

Pause and predict. At $\tau = 1$ the five heroes share one payoff: a team game. What credit-assignment problem does that create? What problem does $\tau = 0$ create?

Source: Berner et al. (2019), Appendix G (p. 43; Eq. 12, p. 45), Table 2 (p. 31).

Surgery: Keep Training While the Game Changes

Part III ▷ §3.3, §4.2

Problem. Over ten months the code, observation and action spaces, network size, and game version all changed.

Surgery. Offline edits that turn the old policy into a policy for the new environment, where possible computing the *same function* from states to action probabilities.

- Over twenty surgeries, about one every two weeks.
- A from-scratch "Rerun" with the final setup used about 20% of the compute and passed 98% wins against OpenAI Five.
- Retraining after each of twenty major surgeries: an estimated 40 months instead of 10.

Course reading. Surgery warm-starts a best-response oracle in a game whose rules drift, as for a deployed LLM agent whose tools change.

Source: Berner et al. (2019), §3.3 pp. 7 to 8, §4.2 p. 10.

Results: World Champions, Then the Public

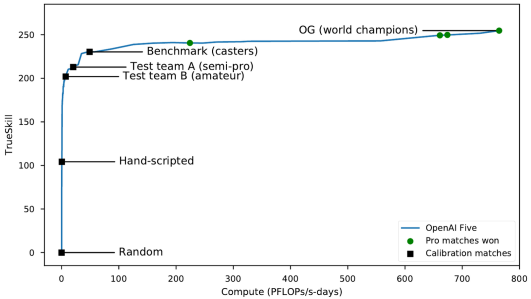


Figure: Berner et al., arXiv 2019, Fig. 3.

Apr. 13, 2019: beat world champions OG 2 to 0.

Arena, Apr. 18 to 21: 3,193 teams, 7,257 games, 99.4% won; 29 teams won 42 games. The paper counts abandoned games as wins (3,140 of 7,215).

How people won (press, Vice 2019): winners "never engage Five in a fair fight," per Brockman.

Average case vs. worst case. 99.4% against whoever showed up says little about the best response to the policy.

Source: Berner et al. (2019), §4 p. 8 and footnote 9; Vice/Motherboard (Apr. 2019), journalism.

Part IV: Strong Is Not Unexploitable

A superhuman Go program, beaten by a policy that loses to amateurs

Two Ways to Measure a Policy

Part IV ▷ Recall M2: Nash and maxmin

Fix a 2p0s game with value v^* and payoff u_1 to the row player. For a row strategy π and a population $\mathcal{D}$ of column strategies:

$$\underbrace{\mathbb{E}_{\sigma\sim\mathcal{D}}\, u_1(\pi,\sigma)}_{\text{average-case strength}} \qquad \text{versus} \qquad \underbrace{\mathrm{expl}(\pi) := v^* - \min_{\sigma} u_1(\pi,\sigma)}_{\text{exploitability } \geq 0}$$

$\mathrm{expl}(\pi) = 0$ exactly when π is a maxmin (Nash) strategy.

- Elo, MMR, and Arena win rates measure the left side, with $\mathcal{D}$ = whoever you played.
- For Go or Dota no one can compute $\min_\sigma$. Proxy: *train* a best response to a frozen π and see how often it wins.

Pause and predict. A policy wins 99% of games against every human it has played. What can you justify about its exploitability?

Source: Standard definitions (M2, M5).

Adversarial Policies Beat Superhuman Go AIs (ICML 2023)

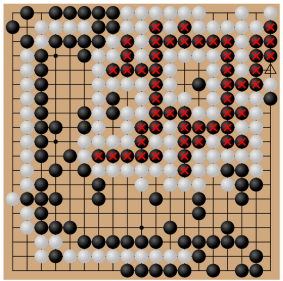


Figure: Wang et al., ICML 2023, Fig. 1.1a.

Freeze KataGo (AlphaZero-style; the strongest public Go AI then) and train an adversary against it by RL.

- $>$ 97% wins at superhuman search settings; $>$ 99% without search; under 14% of KataGo's training compute.
- A human Go expert learned the attack: 14 of 15 wins.
- The adversary (white, left) lures KataGo into a large group ($\times$) that it then captures.
- The adversary *loses to amateurs*: a cycle through KataGo, pros, and amateurs.

Lesson. Self-play made KataGo strong where self-play goes; the adversary steers to positions it never reached.

Source: Wang et al. (2023), PMLR 202: abstract, §1, Fig. 2.1, Appendix G (arXiv v4 pp. 1 to 4, 43).

Part V: Cicero and Diplomacy

Seven players, private talk, and too many equilibria to choose from

Beyond Two-Player Zero-Sum: What Is the Target?

Diplomacy. Seven players compete for supply centers. Each turn: private, nonbinding negotiation, then simultaneous moves.

- General-sum games have many equilibria with different payoffs, and they are not interchangeable (M2). The issue is *which* one.
- A dialogue-free self-play agent, superhuman in 2p0s variants, "performed poorly in games with multiple human players."
- Self-play "is no longer guaranteed to find a policy that performs well with humans, even with infinite compute."

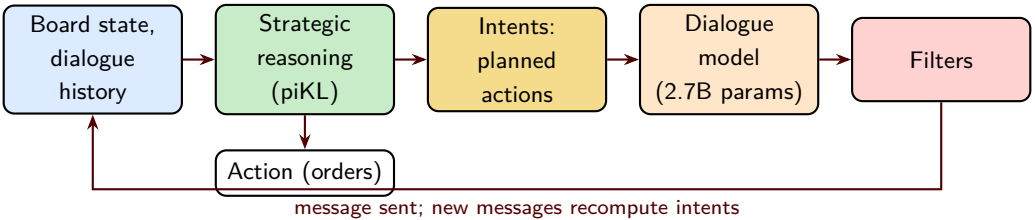
New target. A strong best response to a realistic, data-built model of the humans at the table.

Pause and predict. Pluribus had six players and did well with self-play. What differs in Diplomacy?

Source: FAIR et al. (2022), Science 378, p. 1067.

Cicero's Architecture

Part V ▷ FAIR et al. 2022, Fig. 1



- **Intents** are planned actions for Cicero and its partner. The dialogue model is conditioned on them, so messages are about a plan the planner chose.
- Filters drop messages that are "nonsensical, inconsistent with intents, or strategically poor."

Source: FAIR et al. (2022), p. 1068, Fig. 1.

Intents in Action: Same Board, Planned Moves, Generated Talk

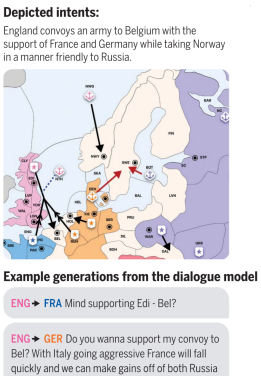


Figure: FAIR et al., Science 2022, Fig. 3.

- The arrows are *intents*: planned orders for England and its partners this turn.
- Given those intents, the dialogue model writes requests that match them: "Mind supporting Edi - Bel?"
- The paper shows the same position with three different intents; each yields different messages.
- Intents are recomputed whenever Cicero sends or receives a message; it plays its latest intent at the end of the turn.

Source: FAIR et al. (2022), pp. 1068 to 1070, Fig. 3 (p. 1070).

piKL: Best-Respond, but Stay Close to the Humans

Part V ▷ Jacob et al., ICML 2022

Player i has an *anchor policy* τ_i learned by imitating humans (τ here is unrelated to team spirit) and an anchor strength $\lambda_i > 0$. piKL models player i as maximizing

$$U_i(\pi_i, \pi_{-i}) \;=\; u_i(\pi_i, \pi_{-i}) \;-\; \lambda_i \, D_{\mathrm{KL}}(\pi_i \,\|\, \tau_i)\,.$$

- Against fixed π_{-i}, on actions with $\tau_i(a) > 0$: $\pi_i(a) \propto \tau_i(a) \exp\big(u_i(a, \pi_{-i})/\lambda_i\big)$. Large λ_i: human-like. Small λ_i: near a best response.
- **piKL-Hedge** runs Hedge on U_i. Each player's regret is $O(\log T)$. In 2p0s games the average policies converge to a Nash equilibrium of the *modified* game; in general games the empirical distribution of joint play approaches the coarse correlated equilibria (CCE) of that game.
- **Cicero** plays $\arg\max_a \; u_i(a, \pi_{-i}) + \lambda \log \tau_i(a)$ with $\lambda = 3 \times 10^{-3}$, looser than for predicting others.

Source: Jacob et al. (2022), pp. 4 to 5, App. A (Props. 1, 2); FAIR et al. (2022), pp. 1071 to 1072. Derivation in the notes.

Talk Tied to Intent: Cheap Talk, and Cicero's Results

Part V ▷ Recall M7: cheap talk; M8: monitoring



Figure: FAIR et al., Science 2022, Fig. 6.

- In this real game Cicero is Austria. It talks about the action it intends, which "maximizes Cicero's honesty and its ability to coordinate but risks leaking information."
- **Cheap talk (M7):** messages are free and nonbinding. Tying them to real intent is a commitment the designers built in.
- **Results:** 40 anonymous online games; more than double the average human score; top 10% of players with 2+ games.
- **For LLM agents,** where the model itself plans, a monitor must check that talk matches action (M8).

Source: FAIR et al. (2022), pp. 1067, 1069 to 1070; Fig. 6 (p. 1072).

Part VI: Pluto (2026), and the Big Picture

Pure self-play again, in StarCraft: Brood War

Pluto (2026): A Brood War Bot From Pure Self-Play

Part VI ▷ Competition report and the author's README; not peer-reviewed

Bot	Games	Win	Loss	Win %	Bots Beaten
Pluto	2884	2844	40	98.61	9
PurpleWave	2883	2037	846	70.66	8
Stardust	2884	2237	647	77.57	7
BananaBrain	2889	2135	754	73.9	6
Microwave	2884	1318	1566	45.7	4
McRave	2889	1313	1576	45.45	4
Steamhammer	2883	1025	1858	35.55	4
UAlbertaBot	2882	904	1978	31.37	2
Venator	2883	483	2400	16.75	1
InfestedArtosis	2889	129	2760	4.47	0
Total	**14425**	**14425**	**14425**	**N/A**	**N/A**

Figure: Churchill, CoG 2026 report (round robin, 6 of 11 columns).

- By Vegard Mella. Won the round robin and the final (4 to 1 over PurpleWave).
- Organizer: the first Brood War competition bot "made completely by reinforcement learning."
- One 315M-parameter network for all races. PPO-style self-play against its *current* weights only; no replays, no search.
- Picks openings with a bandit over per-opponent results (M3).

Source: Churchill (2026), competition report; Mella (2026), README, Model section. Not peer-reviewed.

Whom Do You Train Against? Three Answers

	AlphaStar (2019)	**OpenAI Five (2019)**	**Pluto (2026)**
Human data	971k replays, then RL	none	none
Opponents	league: SP, PFSP, exploiters	80% latest, 20% past	latest self only
Diversity from	exploiters; strategy statistic	past-opponent pool	sampled openings; diversity bonus
Adapts in play	not reported	not reported	bandit over openings (M3)
Source	Nature	arXiv preprint	README, report

A puzzle. AlphaStar found naive self-play "more forgetful," yet Pluto uses only self-play. *Hypotheses only:* sampled openings and a diversity bonus may supply a league's width.

Source: Vinyals et al. (2019) Fig. 3; Berner et al. (2019) p. 5; Mella (2026), Training signal section.

Synthesis: One Question, Many Answers

Part VI ▷ Core technique and the course concept behind it

System	Core technique	Course concept
Libratus	CFR blueprint; subgame solving	Nash, CFR (M5, M6)
Pluribus	Monte Carlo CFR blueprint; limited search	no guarantee beyond 2p0s
AlphaZero	MCTS plus self-play	minimax, policy iteration (M4)
AlphaStar	imitation; league with PFSP	fictitious play, PSRO (M3, M7)
OpenAI Five	PPO; 80/20 self-play; team spirit	team games
KataGo attack	RL best response to a frozen victim	exploitability
Cicero	piKL; intent-grounded dialogue	equilibrium selection; cheap talk
Pluto	pure self-play; opening bandit	limits of self-play; bandits

Source: Previous slides.

Three Things to Keep

1. **Self-play is an equilibrium heuristic with a home turf.** In 2p0s perfect-information games it acts like approximate policy iteration toward the minimax value, without a convergence guarantee. Cycles, teams, and general-sum payoffs each break a different piece.

2. **Leagues are PSRO with engineering.** Keep a population, choose opponents with a meta-strategy (PFSP), add best-response oracles (exploiters), and measure with a Nash distribution.

3. **Strength is not robustness, and equilibrium is not always the goal.** A 99% win rate does not bound exploitability. With many players and language, the target becomes a best response to a realistic model of others (piKL).

Discussion Questions

Use the 5-minute buffer for one; the others make good project or reading-response prompts.

1. **Leagues for LLM agents.** Map AlphaStar's three roles onto a red-teaming program for a deployed LLM agent. What plays the role of PFSP? What is the meta-game payoff matrix?

2. **Is 99.4% good?** Propose an evaluation protocol for a game-playing agent that reports something closer to exploitability than to Elo. What does it cost?

3. **Grounded talk.** Would you require an LLM negotiating for you to talk only about actions it intends? What would it leak, and how would a monitor (M8) check?

What Comes Next

- **M9, Lecture 2:** agent benchmarks (SWE-bench, WebArena, GAIA) and evaluation: cost, variance, contamination, reproducibility. Arena and KataGo are the cautionary tales.
- **Reading:** FAIR et al. (2022), Cicero, *Science* (syllabus). For the league: Vinyals et al. (2019), Methods, "Multi-agent learning."
- **In the notes only:** MuZero; the 2025 to 2026 hidden-information results Ataraxos (Stratego) and Obscuro (Fog of War chess).

Source status. Every system today has a peer-reviewed paper except OpenAI Five (arXiv preprint) and Pluto (author README and competition report). The Vice account is journalism; the DeepMind blog posts supply figures only.

Backup: What Defends Against Exploitation, and What It Costs

Backup (Part IV) ▷ Tools from the course, mapped to these systems

Tool	Where we saw it	In today's systems
Certified equilibrium	CFR, exploitability bounds (M5)	Libratus, Pluribus blueprints
Population training	fictitious play (M3); PSRO (M7)	AlphaStar league; OpenAI Five's 20%
Dedicated exploiters	best-response oracles (M7)	AlphaStar's exploiters
Red-teaming after training	monitoring (M8)	OpenAI Five Arena; KataGo attack
Safe exploitation	safe best response (M2)	not used by these systems

Each row trades compute or average strength for robustness. KataGo study: "the core vulnerability uncovered by our attack persists even in KataGo agents adversarially trained to defend against our attack."

Source: Vinyals et al. (2019) p. 4; Berner et al. (2019) p. 5; Wang et al. (2023), abstract.

References i

- Silver, D., Huang, A., et al. (2016). *Mastering the game of Go with deep neural networks and tree search.* Nature 529: 484-489. [peer-reviewed]
- Silver, D., Schrittwieser, J., et al. (2017). *Mastering the game of Go without human knowledge.* Nature 550: 354-359. Author manuscript used. [peer-reviewed]
- Silver, D., Hubert, T., et al. (2018). *A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play.* Science 362: 1140-1144; preprint arXiv:1712.01815 used. [peer-reviewed]
- Schrittwieser, J., Antonoglou, I., et al. (2020). *Mastering Atari, Go, chess and shogi by planning with a learned model.* Nature 588: 604-609; arXiv:1911.08265 used. [peer-reviewed]
- Vinyals, O., Babuschkin, I., Czarnecki, W. M., et al. (2019). *Grandmaster level in StarCraft II using multi-agent reinforcement learning.* Nature 575: 350-354. Author manuscript used. [peer-reviewed]
- DeepMind (Jan. 24, 2019). *AlphaStar: Mastering the real-time strategy game StarCraft II.* Lab blog post (figure source). [lab blog, not peer-reviewed]
- DeepMind (Oct. 30, 2019). *AlphaStar: Grandmaster level in StarCraft II using multi-agent reinforcement learning.* Lab blog post (figure source). [lab blog, not peer-reviewed]
- Berner, C., Brockman, G., et al. (2019). *Dota 2 with Large Scale Deep Reinforcement Learning.* arXiv:1912.06680. [preprint]
- Lanctot, M., Zambaldi, V., et al. (2017). *A Unified Game-Theoretic Approach to Multiagent Reinforcement Learning.* NIPS 2017; arXiv:1711.00832. [peer-reviewed] (PSRO)
- Czarnecki, W. M., Gidel, G., Tracey, B., Tuyls, K., Omidshafiei, S., Balduzzi, D. & Jaderberg, M. (2020). *Real World Games Look Like Spinning Tops.* NeurIPS 2020; arXiv:2004.09468. [peer-reviewed]
- Wang, T. T., Gleave, A., et al. (2023). *Adversarial Policies Beat Superhuman Go AIs.* ICML 2023, PMLR 202; arXiv:2211.00241v4. [peer-reviewed]

References ii

- Meta FAIR Diplomacy Team, Bakhtin, A., Brown, N., et al. (2022). *Human-level play in the game of Diplomacy by combining language models with strategic reasoning*. Science 378: 1067-1074. [peer-reviewed]
- Jacob, A. P., Wu, D. J., Farina, G., Lerer, A., Hu, H., Bakhtin, A., Andreas, J. & Brown, N. (2022). *Modeling Strong and Human-Like Gameplay with KL-Regularized Search*. ICML 2022, PMLR 162; arXiv:2112.07544. [peer-reviewed] (piKL)
- Brown, N. & Sandholm, T. (2018). *Superhuman AI for heads-up no-limit poker: Libratus beats top professionals*. Science 359: 418-424. [peer-reviewed]
- Brown, N. & Sandholm, T. (2019). *Superhuman AI for multiplayer poker*. Science 365: 885-890. [peer-reviewed]
- Sokota, S., Vinitsky, E., Hu, H., Fan, Z., Kolter, J. Z. & Farina, G. (2026). *Scalable decision-making for games of imperfect information*. Nature, published Sept. 30, 2026, doi:10.1038/s41586-026-11036-y; arXiv:2511.07312. [peer-reviewed] (notes only)
- Zhang, B. H. & Sandholm, T. (2026). *General search techniques without common knowledge for imperfect-information games, and application to superhuman Fog of War chess*. ICLR 2026; arXiv:2506.01242. [peer-reviewed] (notes only)
- Churchill, D. (2026). *2026 CoG StarCraft AI Competition Report*. davechurchill.ca/starcraft/cog/results/2026/report/. [competition report; figure source]
- Mella, V. (2026). *Pluto: StarCraft: Brood War AI, CoG 2026 entry* (PLUTO_README.txt, linked from the competition report; also github.com/tscmoo/pluto). [author documentation, not peer-reviewed]
- Vice / Motherboard (April 2019). *'We Will Win': OpenAI's 'Dota 2' Team Is Crushing Humans Online, But Players Are Not Giving Up*. [journalism]