COUPLED DUAL-ADVERSARIAL REGULARIZED SELF-PLAY

Anonymous authors
Paper under double-blind review

ABSTRACT

Naive self-play for two-player zero-sum games can cycle in nontransitive games. Regularized self-play stabilizes training with a single network but only plays against itself, so it forgets earlier opponents and cannot use external ones; population methods keep an opponent pool, but their equilibrium lives in a mixture over the pool, and heuristic weightings such as PFSP carry no convergence guarantee. We propose Coupled Dual-Adversarial Regularized self-play (CoDA-R). Each round, CoDA-R sets the pool weights by a Euclidean projection anchored at pure self-play, so that only opponents currently beating the latest policy receive weight, and then takes a proximal policy step against the resulting mixture. The coupling yields one energy inequality that charges every policy step and every unit of pool weight to the remaining distance to a Nash equilibrium. For the exact algorithm, this gives last-iterate convergence of the latest policy to a Nash equilibrium however oracle opponents enter or leave the pool, plus certificates computable during training and error-tolerant versions of all bounds. In OpenSpiel at a matched budget, the latest network of CoDA-R cuts the tail exploitability on Goofspiel(4) from 0.797 for R-NaD-PPO to 0.083, on par with a multi-network PSRO mixture; on Leduc poker, with the dual strength scaled to the payoff range and an exact best-response oracle, it roughly halves that of R-NaD-PPO; on Liar's Dice the two tie.

1 INTRODUCTION

Two-player zero-sum games are the most basic model of adversarial decision making. Board games, poker, and real-time strategy games are its most direct instances (Silver et al., 2018; Brown & Sandholm, 2018; Vinyals et al., 2019; Pérolat et al., 2022), and the training objectives of generative adversarial networks and robust optimization can also be written as min–max problems (Goodfellow et al., 2014; Daskalakis et al., 2018). The natural target is a Nash strategy, commonly obtained by self-play, in which the current policy plays against itself or its past versions and is updated by reinforcement learning, requiring neither human data nor explicit opponent modeling.

However, self-play has a well-known failure mode. Real games commonly contain nontransitive components (Czarnecki et al., 2020), and naive self-play, which updates only against the current strongest opponent, spins around in rock–paper–scissors-style cycles and fails to converge (Balduzzi et al., 2018; 2019). The two families of remedies each leave something to be desired. Regularized self-play (R-NaD, MMD) adds a periodically moving anchor in payoff or policy space, which turns the cyclic dynamics into a contraction toward a regularized equilibrium, with the guarantee landing on a single network (Pérolat et al., 2021; 2022; Sokota et al., 2023). The price is that it only plays against itself, so scripted opponents, exploiters trained to counter the current policy, and historical snapshots cannot be plugged in, vulnerabilities off the training trajectory are neither found nor repaired, and earlier strategies are easily forgotten (OpenAI et al., 2019; Bansal et al., 2018). Population methods (Double Oracle, PSRO, PFSP, and league training) retain historical opponents (McMahan et al., 2003; Lanctot et al., 2017; Vinyals et al., 2019), but their equilibrium is a mixture over the pool or a separately learned average network, and the network being trained carries no guarantee itself. Heuristic weightings that allocate sampling proportions by win rate or cumulative score (PFSP, the OpenAI Five sampler) almost all lack convergence proofs.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

We propose Coupled Dual-Adversarial Regularized self-play (CoDA-R), whose pool weights are a dual variable that plays against the current policy (hereafter *latest*), and whose two updates are coupled into an energy inequality that deducts both the movement of latest and the weight held by the pool from the squared distance from latest to a Nash equilibrium (§4). Since this distance is finite, both must vanish, and a policy that no longer moves while the pool holds no weight can only be a Nash strategy of the full game, however imperfect the exploiter and however the pool changes. This targets settings that neither family covers, for example black-box opponents available only through play (scripted bots, exploiters, legacy agents of other architectures), games too long or costly to evaluate a large pool exhaustively, a pool that changes online, and a single deployed network. PSRO needs the full $N \times N$ meta-game payoff matrix and deploys a mixture, whereas CoDA-R needs only the N loss rates of latest, estimated from its own training games; R-NaD-style methods cannot use such opponents, and PFSP uses them without a guarantee. Our novelty lies in the dual rule, which anchors the pool weights at pure self-play and thereby lets an open pool coexist with last-iterate convergence of one network.

CoDA-R grew out of an internal predecessor, CoDA, whose dual anchor moves with training and which converges at rate $O(1/K)$ in the restricted game defined by the pool (§3). Once the oracle misses a vulnerability, however, latest may stop at a best response to the pool that outside opponents easily beat. Adding latest itself to the pool seems to patch this, but weight then flows to this member with constant loss rate $\frac{1}{2}$ and the moving anchor retains it, pinning latest at a non-Nash point (Proposition 3.2). CoDA-R moves the moving anchor to the primal side and fixes the dual anchor at pure self-play; the energy inequality above comes from this swap. Our contributions are as follows.

- **Algorithm.** To our knowledge, CoDA-R is the first self-play method that accepts arbitrary external opponents (scripted agents, exploiters, historical snapshots) while guaranteeing that the trained policy itself converges to a Nash equilibrium in the last iterate (exact updates, normal-form games). Prior pool methods guarantee only a mixture or a time average, and regularized self-play admits no external opponents. It needs only game outcomes against each opponent, so any agent that can play may join the pool, and it wraps an unmodified PPO learner up to a proximal reward term (§4).
- **Theory.** A single energy inequality (Theorem 5.3) charges both policy movement and pool weight to the distance to a Nash equilibrium, so the pool can hold weight only by buying progress toward it. This yields last-iterate convergence of the trained policy, with an explicit $O(K^{-1/4})$ last-iterate rate once the pool stops changing (Rule A), together with online certificates and error-tolerant versions, for any sequence of pool admissions and any oracle (§5).
- **Experiments.** No inequality is violated in about 3.7 million per-step finite-dimensional records. On OpenSpiel at a matched budget, the single latest network of CoDA-R cuts the tail exploitability of R-NaD-PPO on Goofspiel(4) nearly tenfold on all 4 seeds, matching a multi-network PSRO mixture, and roughly halves it on Leduc with an exact best-response (BR) oracle; we also report when the gains depend on oracle strength and payoff scale (§6).

2 Related Work and Preliminaries

2.1 Related Work

Population methods. Double Oracle and PSRO (McMahan et al., 2003; Lanctot et al., 2017; Muller et al., 2020; McAleer et al., 2020) solve a meta-game externally (Wellman, 2006), PFSP and league training (Vinyals et al., 2019) allocate sampling proportions by win rate, and NFSP (Heinrich et al., 2015; Heinrich & Silver, 2016) learns the historical average strategy with a separate network; all of them deploy a mixture of checkpoints or the average network.

Regularized self-play. R-NaD and MMD (Pérolat et al., 2021; 2022; Sokota et al., 2023) use an anchor in payoff or policy space that moves periodically or is annealed, and optimistic and extragradient methods (Daskalakis et al., 2018; Mertikopoulos et al., 2019; Nemirovski, 2004; Cen et al., 2021) stabilize similarly with an extrapolation term; both are explicit approximations (Mokhtari et al., 2020) of the proximal point method (Rockafellar, 1976). They guarantee a single network but train only along the self-play trajectory, which is why OpenAI Five and AlphaStar also keep historical opponents (OpenAI et al., 2019; Vinyals et al., 2019; Czarnecki et al., 2020).

Multiplicative weights and sleeper members. The meta-solvers of the PSRO family (Hedge, projected replicator dynamics, MWU) and the opponent sampler of OpenAI Five mostly use multiplicative weights (Lanctot et al., 2017; McAleer et al., 2022; Dinh et al., 2022; OpenAI et al., 2019; Freund & Schapire, 1999), which accumulate history, so after a member has been suppressed for T rounds, even if it starts countering latest again, it has to win for roughly another T rounds to regain its weight. We call this the *sleeper* problem, which Fixed Share-type algorithms in online learning address (Herbster & Warmuth, 1998). Conversely, tied members keep positive weight indefinitely. Methods that deploy the time average are unaffected, but the last iterate of multiplicative weights does not converge in zero-sum games in the first place (Bailey & Piliouras, 2018), and members that tie without dropping to zero can further pin latest at a non-Nash point (Proposition 3.2). We believe this is an important reason why existing pool methods cannot combine a convergence proof with single-network convergence. CoDA-R therefore uses a stateless projection (Appendix Table 2 compares methods).

2.2 Preliminaries

Unless otherwise stated, $\|\cdot\|$ denotes the Euclidean (L_2) norm throughout.

Consider a finite zero-sum game in which seats are assigned at random in each game, so that the game is symmetric for the two players. A player has M pure strategies, and the mixed strategy space is $X = \Delta_M$. A sequential game can be unfolded over its pure strategies into this form; the theoretical analysis is carried out at this idealized level, and the gap between it and the neural implementation is absorbed into the error terms of §5.3. Let $\ell(\boldsymbol{x}; \boldsymbol{y}) \in [0, 1]$ denote the expected loss rate of policy $\boldsymbol{x}$ against $\boldsymbol{y}$. Symmetry gives $\ell(\boldsymbol{x}; \boldsymbol{y}) + \ell(\boldsymbol{y}; \boldsymbol{x}) = 1$, so there exists a skew-symmetric matrix $S = -S^\top$ such that

$$\ell(\boldsymbol{x}; \boldsymbol{y}) = \tfrac{1}{2} + \boldsymbol{x}^\top S \boldsymbol{y}, \qquad \boldsymbol{x}^\top S \boldsymbol{x} = 0.$$

For a frozen opponent $\boldsymbol{y}_i$, write $\ell_i(\boldsymbol{x}) := \ell(\boldsymbol{x}; \boldsymbol{y}_i)$, which is an affine function of $\boldsymbol{x}$ with constant gradient $S\boldsymbol{y}_i$. Policy quality is measured by exploitability,

$$\mathrm{Expl}(\boldsymbol{x}) := \max_{\boldsymbol{y} \in X} \ell(\boldsymbol{x}; \boldsymbol{y}) - \tfrac{1}{2} = \max_j (S^\top \boldsymbol{x})_j \;\geq\; 0.$$

A Nash strategy satisfies $\mathrm{Expl}(\boldsymbol{x}_N) = 0$ and exists by the minimax theorem (von Neumann, 1928); moreover,

$$\boldsymbol{x}_N \text{ is Nash} \iff \boldsymbol{x}_N^\top S \boldsymbol{y} \leq 0 \;\; \forall \boldsymbol{y} \in X \iff \boldsymbol{y}^\top S \boldsymbol{x}_N \geq 0 \;\; \forall \boldsymbol{y} \in X. \tag{1}$$

Π_C denotes the Euclidean projection onto a closed convex set C. Throughout the paper we repeatedly use the variational characterization of the projection (hereafter the obtuse-angle principle)

$$\boldsymbol{w} = \Pi_C(\boldsymbol{v}) \iff (\boldsymbol{v} - \boldsymbol{w})^\top (\boldsymbol{u} - \boldsymbol{w}) \leq 0 \quad \forall \boldsymbol{u} \in C, \tag{2}$$

and the monotonicity and nonexpansiveness it implies (Bauschke & Combettes, 2011). The diameter of the simplex is $D = \sqrt{2}$. Projection onto the simplex takes an $O(N \log N)$ thresholding algorithm, and components below the threshold are exactly zero (Duchi et al., 2008).

3 CoDA: A Moving Dual Anchor in the Restricted Game

CoDA solves a *restricted game*. Fix an opponent pool $P = \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N\}$ and write $\boldsymbol{\ell}(\boldsymbol{x}) := (\ell_1(\boldsymbol{x}), \ldots, \ell_N(\boldsymbol{x}))$. The restricted game between the pool and latest is $\min_{\boldsymbol{x} \in X} \max_{\boldsymbol{\rho} \in \Delta_N} \boldsymbol{\rho}^\top \boldsymbol{\ell}(\boldsymbol{x})$, whose value is denoted v^*. The dual function $\varphi(\boldsymbol{\rho}) := \min_{\boldsymbol{x}} \boldsymbol{\rho}^\top \boldsymbol{\ell}(\boldsymbol{x})$ is concave, and $v^* = \max_{\boldsymbol{\rho}} \varphi(\boldsymbol{\rho})$. One step of CoDA is parameterized by a dual anchor $\boldsymbol{\rho}_k \in \Delta_N$ and computes the regularized saddle point

$$(\boldsymbol{x}_{k+1}, \boldsymbol{w}_{k+1}) \in \operatorname*{saddle}_{\boldsymbol{x} \in X,\, \boldsymbol{\rho} \in \Delta_N} \Big[\boldsymbol{\rho}^\top \boldsymbol{\ell}(\boldsymbol{x}) - \tfrac{1}{2c} \|\boldsymbol{\rho} - \boldsymbol{\rho}_k\|^2 \Big], \qquad \boldsymbol{\rho}_{k+1} := \boldsymbol{w}_{k+1}. \tag{3}$$

Given $\boldsymbol{x}$, completing the square gives the dual solution in closed form, $\Pi_{\Delta_N}(\boldsymbol{\rho}_k + c\, \boldsymbol{\ell}(\boldsymbol{x}))$, which gives weight to opponents with high loss rates and exactly zero weight to those below the threshold. $\boldsymbol{x}_{k+1}$ is a best response to $\boldsymbol{w}_{k+1}$.

Proposition 3.1 (CoDA converges in the restricted game)**.** *For a fixed pool and exact steps* (3)*,* $\varphi(\boldsymbol{\rho}_{k+1}) \geq \varphi(\boldsymbol{\rho}_k) + \frac{1}{2c}\|\boldsymbol{\rho}_{k+1} - \boldsymbol{\rho}_k\|^2$*, and hence* $\sum_k \|\boldsymbol{\rho}_{k+1} - \boldsymbol{\rho}_k\|^2 \leq 2c$*. Moreover,*

$$\max_{i \in P} \ell_i(\boldsymbol{x}_{k+1}) - v^* \;\leq\; \frac{\sqrt{2}}{c}\,\|\boldsymbol{\rho}_{k+1} - \boldsymbol{\rho}_k\| \;\longrightarrow\; 0, \qquad v^* - \varphi(\boldsymbol{\rho}_K) \;\leq\; \frac{1}{cK}.$$

The proof is in Appendix A.1.

3.1 Why the Mirror Arm Breaks CoDA

Proposition 3.1 covers only the restricted game, and if the oracle misses vulnerabilities, latest may be easily beaten outside the pool. A direct remedy is to put latest itself into the pool as well, which we call the *mirror arm* m. It is the $(N+1)$-th member, and since any strategy has loss rate constantly $\frac{1}{2}$ against itself, the augmented loss vector is

$$\tilde{\boldsymbol{\ell}}(\boldsymbol{x}) := \big(\ell_1(\boldsymbol{x}), \ldots, \ell_N(\boldsymbol{x}), \tfrac{1}{2}\big).$$

The mirror arm breaks the argument of Proposition 3.1, which pushes the primal side through the nonzero gradient of the weighted loss rate $\boldsymbol{\rho}^\top \boldsymbol{\ell}(\boldsymbol{x})$ in $\boldsymbol{x}$, whereas the mirror coordinate is constantly $\frac{1}{2}$. Once the other members are pushed close to a tie, the mirror arm becomes the member with the highest loss rate, the dual side shifts weight to it, and the part of the weighted objective with a gradient in $\boldsymbol{x}$ vanishes. The moving anchor also retains the pool weight accumulated earlier. When all loss rates are tied, $c\,\tilde{\boldsymbol{\ell}}(\boldsymbol{x})$ is a constant vector and $\Pi_\Delta(\boldsymbol{\rho} + c\tilde{\boldsymbol{\ell}}) = \boldsymbol{\rho}$, so these weights never exit and may exactly cancel the self-play field, as the following example shows.

Proposition 3.2 (Fixed-point counterexample for the moving anchor)**.** *In rock–paper–scissors with* $S = \frac{1}{2}\left(\begin{smallmatrix} 0 & -1 & 1 \\ 1 & 0 & -1 \\ -1 & 1 & 0 \end{smallmatrix}\right)$*, take* $\boldsymbol{x} = \boldsymbol{e}_1$*,* $P = \{\boldsymbol{y}\}$ *with* $\boldsymbol{y} = (0, \frac{1}{2}, \frac{1}{2})$*, dual anchor* $\boldsymbol{\rho} = (\frac{2}{3}, \frac{1}{3})$*, and self-play share* $\lambda = \frac{1}{3}$*. Then both the dual and primal residuals vanish, yet* $\mathrm{Expl}(\boldsymbol{e}_1) = \frac{1}{2}$.

The verification is in Appendix A.2. Here the equilibrium $\frac{1}{3}\boldsymbol{e}_1 + \frac{2}{3}\boldsymbol{y}$ is carried jointly by the pool and latest, while latest itself is not a Nash strategy. More generally, any Nash strategy $\boldsymbol{q}^*$ that can be decomposed as $w_m\boldsymbol{x} + \sum_i w_i\boldsymbol{y}_i$, with the positively weighted members exactly tied against $\boldsymbol{x}$, constitutes a fixed point of the same kind.

4 CoDA-R: Moving the Anchor to the Primal Side

CoDA-R maintains the latest policy $\boldsymbol{x}_k$, a pool of frozen opponents P, and a self-play share λ_k; Algorithm 1 lists one step. The dual side gives weight only to members currently beating latest, according to their lead, and leaves the rest to the mirror arm; the primal side then takes a proximal step for latest against the resulting mixture. In the exact algorithm, lines 1–3 are evaluated at the new policy $\boldsymbol{x}_{k+1}$, i.e., the update is implicit. Unlike in CoDA, the dual anchor is fixed at pure self-play $\boldsymbol{e}_m := (0, \ldots, 0, 1)$, and the moving anchor sits in the primal proximal term, as in R-NaD and MMD. The next three subsections explain this design.

4.1 Dual Side: Regularized Maximization at a Fixed Anchor

The anchor can be fixed because the object being solved for has changed. CoDA seeks a saddle point of the restricted game, and the anchor must move to catch up with an equilibrium dual point that is unknown in advance. CoDA-R seeks a Nash strategy of the full game, and once latest is a Nash strategy no opponent can beat it, so the correct dual weight is then the known point $\boldsymbol{e}_m$.

Let $c > 0$ denote the dual strength, which is held fixed during training. Given a policy $\boldsymbol{x}$, the dual side solves a maximization problem with a quadratic regularizer, whose solution is exactly the projection above:

$$\boldsymbol{w}(\boldsymbol{x}) := \operatorname*{arg\,max}_{\boldsymbol{w} \in \Delta_{N+1}} \Big[\boldsymbol{w}^\top \tilde{\boldsymbol{\ell}}(\boldsymbol{x}) - \frac{1}{2c}\|\boldsymbol{w} - \boldsymbol{e}_m\|^2\Big] = \Pi_{\Delta_{N+1}}\big(\boldsymbol{e}_m + c\,\tilde{\boldsymbol{\ell}}(\boldsymbol{x})\big), \tag{4}$$

Algorithm 1 One step of CoDA-R (PPO approximations in Appendix C)

Require: latest $\boldsymbol{x}_k$, pool $P = \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N\}$, share λ_k, dual strength c, proximal strength η

1: $\tilde{\boldsymbol{\ell}}(\boldsymbol{x}) \leftarrow \big(\ell_1(\boldsymbol{x}), \ldots, \ell_N(\boldsymbol{x}), \frac{1}{2}\big)$ ▷ loss rates of latest; mirror arm $\frac{1}{2}$

2: $\boldsymbol{w} \leftarrow \Pi_{\Delta_{N+1}}\big(\boldsymbol{e}_m + c\,\tilde{\boldsymbol{\ell}}(\boldsymbol{x})\big)$ ▷ dual side (whom to play)

3: Train latest against itself (share λ_k) and each $\boldsymbol{y}_i$ (share w_i), pulled toward $\boldsymbol{x}_k$ ▷ primal side

4: $\boldsymbol{x}_k \leftarrow \boldsymbol{x}_{k+1}$, $\lambda_{k+1} \leftarrow w_m$ ▷ move the anchor (Rule B, §4.3)

5: Optionally freeze latest, train an exploiter, and add it to P

where the second equality follows from completing the square. The first term in the brackets pushes weight toward opponents with high loss rates, and the second pulls weight back toward pure self-play, with a weaker pull for larger c. Let $\boldsymbol{w}_P$ denote the pool coordinates (the first N components) and w_m the mirror weight. Since the anchor is fixed, $\boldsymbol{w}(\boldsymbol{x})$ is a closed-form continuous function of $\boldsymbol{x}$ with no state to integrate or remember, which is also why new members need no initialization when they enter the pool. Three properties follow directly from the obtuse-angle principle.

Lemma 4.1 (Properties of the dual rule)**.** *(a) $\boldsymbol{w}(\boldsymbol{x}) = \boldsymbol{e}_m$ if and only if $\max_{i\in P} \ell_i(\boldsymbol{x}) \le \frac{1}{2}$.*
(b) A member holding weight must be currently beating latest:

$$w_i(\boldsymbol{x}) > 0 \implies \ell_i(\boldsymbol{x}) - \tfrac{1}{2} \;\ge\; \frac{1 - w_m(\boldsymbol{x}) + w_i(\boldsymbol{x})}{c} \;>\; 0.$$

(c) If the pool has a single member and $\ell_1(\boldsymbol{x}) = \frac{1}{2} + \mu$, then $w_1(\boldsymbol{x}) = \mathrm{clip}(\frac{c\mu}{2}, 0, 1)$.

The proof is in Appendix A.3. (a) says that when no member beats latest, all weight goes to self-play and the step is exactly proximal self-play, the skeleton of R-NaD and MMD; CoDA-R departs from regularized self-play only while some opponent beats latest. (b) says that heavier members lead by more. (c) is the closed form tested directly in §6.3.

4.2 Primal Side: An Implicit Proximal Best Response

Given the anchor $\boldsymbol{x}_k$, the share $\lambda_k \ge 0$, and the proximal strength $\eta > 0$, first temporarily fix the opponent mixture to a self-play opponent $\bar{\boldsymbol{y}}$ with share λ_k together with pool members with weights $\boldsymbol{w}$. The primal side computes a best response with a proximal term, i.e., it minimizes

$$f_k(\boldsymbol{x}) := \lambda_k\, \ell(\boldsymbol{x}; \bar{\boldsymbol{y}}) + \sum_{i\in P} w_i\, \ell_i(\boldsymbol{x}) + \frac{\eta}{2}\|\boldsymbol{x} - \boldsymbol{x}_k\|^2, \qquad \boldsymbol{x} \in X. \tag{5}$$

The first two terms are the weighted loss rate against the mixed opponent, and the entire effect of the mirror arm on the primal side is carried by the first term; the third term pulls the policy toward the anchor. With $A := [S\boldsymbol{y}_1 \cdots S\boldsymbol{y}_N] \in \mathbb{R}^{M\times N}$, the gradient of the first two terms is the constant vector $\lambda_k S\bar{\boldsymbol{y}} + A\boldsymbol{w}_P$, so f_k has the same shape as (4), a linear term plus a quadratic regularizer, and by completing the square its minimizer is $\Pi_X\big(\boldsymbol{x}_k - \frac{1}{\eta}[\lambda_k S\bar{\boldsymbol{y}} + A\boldsymbol{w}_P]\big)$. The exact algorithm requires the opponent mixture to be evaluated at the solution itself, i.e., $\bar{\boldsymbol{y}} = \boldsymbol{x}_{k+1}$ and $\boldsymbol{w} = \boldsymbol{w}(\boldsymbol{x}_{k+1})$, so the primal update is

$$\boldsymbol{x}_{k+1} = \Pi_X\Big(\boldsymbol{x}_k - \tfrac{1}{\eta}\big[\lambda_k S\boldsymbol{x}_{k+1} + A\,\boldsymbol{w}(\boldsymbol{x}_{k+1})_P\big]\Big), \tag{6}$$

i.e., a projected step of size $1/\eta$ from the anchor that lowers the weighted loss rate; since the unknown appears on both sides, the update is implicit. By the obtuse-angle principle, (6) is equivalent to the variational inequality

$$\big\langle \lambda_k S\boldsymbol{x}_{k+1} + A\,\boldsymbol{w}(\boldsymbol{x}_{k+1})_P + \eta(\boldsymbol{x}_{k+1} - \boldsymbol{x}_k),\; \boldsymbol{x}' - \boldsymbol{x}_{k+1}\big\rangle \;\ge\; 0 \qquad \forall \boldsymbol{x}' \in X, \tag{P}$$

whose three terms are, in order, the gradients at $\boldsymbol{x}_{k+1}$ of the three terms of (5) with $\bar{\boldsymbol{y}} = \boldsymbol{x}_{k+1}$ and $\boldsymbol{w} = \boldsymbol{w}(\boldsymbol{x}_{k+1})$. All subsequent proofs plug a suitable test point $\boldsymbol{x}'$ into (P). Denote the combined opponent and its total mass by

$$\boldsymbol{v}_{k+1} := \lambda_k \boldsymbol{x}_{k+1} + \textstyle\sum_i w_i(\boldsymbol{x}_{k+1})\,\boldsymbol{y}_i, \qquad t_k := \lambda_k + 1 - w_m(\boldsymbol{x}_{k+1}).$$

Then $S\boldsymbol{v}_{k+1}$ is exactly the first two terms of (P), and (P) can be read as latest making a proximal best response to the mixed opponent $\boldsymbol{v}_{k+1}$. The update must be implicit. Lemma 5.2 requires the dual weights and the self-play opponent to be evaluated at the same point $\boldsymbol{x}_{k+1}$; if the opponent is taken to be the old snapshot $\boldsymbol{x}_k$, the resulting explicit update drifts away from the Nash point on a skew-symmetric field and no longer converges (middle panel of Appendix Figure 2). For any anchor, any $\lambda_k \ge 0$, and any pool, (P) has exactly one solution (Proposition A.1 in Appendix A.4).

4.3 SELF-PLAY SHARE

λ_k is fixed within a step, and there are two ways to choose it:

$$\text{Rule A: } \lambda_k \equiv 1; \qquad \text{Rule B (default): } \lambda_0 = 1,\ \ \lambda_{k+1} := w_m(\boldsymbol{x}_{k+1}).$$

Under Rule B, $t_k = 1 + \lambda_k - \lambda_{k+1}$, which stays close to 1. Two choices must be ruled out. With $\lambda \equiv 0$, the primal field vanishes once the pool weights reach zero, and every $\boldsymbol{x}$ is a fixed point. With λ set to the current step's $w_m(\boldsymbol{x}_{k+1})$, the monotonicity required by Proposition A.1 no longer holds.

5 THEORETICAL ANALYSIS

The results in this section hold for exact steps, where the pool P_k may change arbitrarily at every step and λ_k follows Rule A or B. Fix a step k, abbreviate $\boldsymbol{x} := \boldsymbol{x}_{k+1}$, $\boldsymbol{w} := \boldsymbol{w}(\boldsymbol{x})$, $\boldsymbol{v} := \boldsymbol{v}_{k+1}$, and $\Delta\boldsymbol{x}_k := \boldsymbol{x}_{k+1} - \boldsymbol{x}_k$, and define the total pool weight a and the pool pressure b:

$$a(\boldsymbol{x}) := 1 - w_m = \textstyle\sum_{i\in P} w_i, \qquad b(\boldsymbol{x}) := \textstyle\sum_{i\in P} w_i\big(\ell_i(\boldsymbol{x}) - \frac{1}{2}\big).$$

By Lemma 4.1(b), every member holding weight has $\ell_i > \frac{1}{2}$, so $b \geq 0$. Direct expansion gives two identities:

$$\boldsymbol{x}^\top S\boldsymbol{v} = \lambda \underbrace{\boldsymbol{x}^\top S\boldsymbol{x}}_{=0} + \textstyle\sum_i w_i\, \boldsymbol{x}^\top S\boldsymbol{y}_i = b(\boldsymbol{x}), \qquad \boldsymbol{w}^\top\tilde{\boldsymbol{\ell}}(\boldsymbol{x}) = \textstyle\sum_i w_i\ell_i(\boldsymbol{x}) + \frac{1}{2}w_m = b(\boldsymbol{x}) + \frac{1}{2}. \quad (7)$$

5.1 SHARED ENERGY BUDGET

Lemma 5.1 (Pool weight must be backed by current gains)**.**

$$b(\boldsymbol{x}) \ \geq\ \frac{1}{c}\,\|\boldsymbol{w}(\boldsymbol{x}) - \boldsymbol{e}_m\|^2.$$

The proof (Appendix A.5) applies the obtuse-angle principle to (4) with the anchor $\boldsymbol{e}_m$ itself as the test point. In words, the pool can hold weight only in proportion to how much it currently beats latest. This is exactly the step where the moving anchor fails. The same derivation with a moving anchor $\boldsymbol{\rho}_k$ yields only

$$b \ \geq\ -\tfrac{1}{c}\,(\boldsymbol{w} - \boldsymbol{\rho}_k)^\top(\boldsymbol{e}_m - \boldsymbol{w}),$$

whose right-hand side can be zero or even negative, so a tied pool ($b = 0$) can still retain weight. The fixed anchor replaces the right-hand side with a squared term, so a tied pool cannot retain any weight, and the entire family of fixed points in Proposition 3.2 is thereby ruled out.

Lemma 5.2 (A Nash strategy as the test point)**.** *For any Nash strategy $\boldsymbol{x}_N$,*

$$\eta\,\Delta\boldsymbol{x}_k^\top(\boldsymbol{x}_N - \boldsymbol{x}_{k+1}) \ \geq\ b(\boldsymbol{x}_{k+1}).$$

The proof (Appendix A.5) takes $\boldsymbol{x}' = \boldsymbol{x}_N$ in (P) and uses that no mixture of opponents, including $\boldsymbol{v}_{k+1}$, can beat a Nash strategy. In words, whatever the pool gains against latest must show up as a step toward $\boldsymbol{x}_N$.

Theorem 5.3 (Shared energy budget)**.** *For any step k, any pool P_k, any $\lambda_k \geq 0$, and any Nash strategy $\boldsymbol{x}_N$,*

$$\|\boldsymbol{x}_{k+1} - \boldsymbol{x}_N\|^2 + \|\boldsymbol{x}_{k+1} - \boldsymbol{x}_k\|^2 + \frac{2}{c\eta}\|\boldsymbol{w}(\boldsymbol{x}_{k+1}) - \boldsymbol{e}_m\|^2 \ \leq\ \|\boldsymbol{x}_k - \boldsymbol{x}_N\|^2. \quad (8)$$

The proof takes two lines (Appendix A.5). It expands $\|\boldsymbol{x}_k - \boldsymbol{x}_N\|^2$ by the polarization identity and bounds the cross term $2\Delta\boldsymbol{x}_k^\top(\boldsymbol{x}_N - \boldsymbol{x}_{k+1})$ from below by $\frac{2}{\eta}b$ with Lemma 5.2 and then by $\frac{2}{c\eta}\|\boldsymbol{w} - \boldsymbol{e}_m\|^2$ with Lemma 5.1. The right-hand side is the remaining budget before this step; the left-hand side holds the budget after it and the two kinds of spending in this step, policy movement and weight held by the pool.

Corollary 5.4 (Summable, vanishing, and robust to pool admissions)**.** *Telescoping over $k < K$,*

$$\textstyle\sum_{k<K}\|\Delta\boldsymbol{x}_k\|^2 \leq \|\boldsymbol{x}_0 - \boldsymbol{x}_N\|^2 \leq 2, \qquad \sum_{k<K}\|\boldsymbol{w}(\boldsymbol{x}_{k+1}) - \boldsymbol{e}_m\|^2 \leq \tfrac{c\eta}{2}\|\boldsymbol{x}_0 - \boldsymbol{x}_N\|^2 \leq c\eta,$$

hence $\Delta\boldsymbol{x}_k \to 0$, $\boldsymbol{w}(\boldsymbol{x}_{k+1}) \to \boldsymbol{e}_m$, $a(\boldsymbol{x}_{k+1}) \leq \|\boldsymbol{w}(\boldsymbol{x}_{k+1}) - \boldsymbol{e}_m\| \to 0$, and $\lambda_k \to 1$ under Rule B. Since (8) *involves only $\boldsymbol{x}_k$, $\boldsymbol{x}_{k+1}$, $\boldsymbol{w}(\boldsymbol{x}_{k+1})$, and the full-game Nash strategy $\boldsymbol{x}_N$, arbitrary additions or removals of pool members between steps (including all oracle admissions) do not change the inequality at any step, and the conclusions above hold over the entire training run.*

5.2 Measurable Certificates, Convergence, and Rates

We call an upper bound on exploitability that involves only observable quantities a *certificate*.

Theorem 5.5 (Two certificates)**.** *Let $D = \sqrt{2}$. (i) Full-game certificate for latest:*

$$\lambda_k \mathrm{Expl}(\boldsymbol{x}_{k+1}) \;\le\; \tfrac{1}{2}a(\boldsymbol{x}_{k+1}) - b(\boldsymbol{x}_{k+1}) + \eta D\|\Delta \boldsymbol{x}_k\| \;\le\; \tfrac{1}{2}a(\boldsymbol{x}_{k+1}) + \eta D\|\Delta \boldsymbol{x}_k\|.$$

(ii) In-pool certificate:

$$\Big[\max_{i\in P} \ell_i(\boldsymbol{x}_{k+1}) - \tfrac{1}{2}\Big]^+ \;\le\; \frac{\sqrt{2}}{c}\,\|\boldsymbol{w}(\boldsymbol{x}_{k+1}) - \boldsymbol{e}_m\| + \eta D\|\Delta \boldsymbol{x}_k\|.$$

The proof is in Appendix A.6. Both right-hand sides involve only the pool weight a, the distance to pure self-play $\|\boldsymbol{w} - \boldsymbol{e}_m\|$, and the primal residual $\|\Delta \boldsymbol{x}\|$, which are all computable during training and valid for any oracle.

Theorem 5.6 (Latest converges to Nash)**.** *With exact steps, Rule A or B, and arbitrary P_k, $\mathrm{Expl}(\boldsymbol{x}_k) \to 0$, and the whole sequence $\boldsymbol{x}_k$ converges to some Nash strategy.*

The proof is in Appendix A.5. The budget drives both the pool weight and the residual to zero, so certificate (i) drives the exploitability to zero, and the energy inequality with any limit point as reference upgrades this to convergence of the whole sequence. Since convergence holds however exploiters enter or leave the pool, any oracle, however cheap or heuristic, can be plugged in safely, and it adds speed and visibility. Self-play only exposes weaknesses that latest can find against itself, so one exploitable only by a distant strategy or hidden on rarely reached branches is repaired slowly; an exploiter targets it directly, and its lead becomes pool weight at once. A reopened hole regains weight immediately, with no sleeper delay, and each exploiter's measured lead certifies a lower bound on the exploitability of latest, which full-tree best responses cannot give in large games (Appendix A.9).

Theorem 5.7 (Best- and last-iterate rates of latest)**.**

$$\textit{(i)}\;\; \min_{k<K} \lambda_k \mathrm{Expl}(\boldsymbol{x}_{k+1}) \le \frac{\sqrt{c\eta}/2 + 2\eta}{\sqrt{K}}, \qquad \textit{(ii)}\;\; \mathrm{Expl}(\boldsymbol{x}_K) \le \sqrt{c\eta/2}\; n^{-1/4} + 2\eta\, n^{-1/2},$$

where (ii) holds under Rule A whenever the pool is unchanged during the n steps before step K.

The proof is in Appendix A.7. Thus the trained policy itself has a last-iterate rate $O(K^{-1/4})$ once the pool stops changing; mixtures of opponents also converge (Proposition A.2).

5.3 Versions with Errors

An implementation introduces two kinds of error. The dual side uses an estimate $\hat{\boldsymbol{\ell}}_k$ in place of $\tilde{\boldsymbol{\ell}}(\hat{\boldsymbol{x}}_{k+1})$, satisfying $\|\hat{\boldsymbol{\ell}}_k - \tilde{\boldsymbol{\ell}}(\hat{\boldsymbol{x}}_{k+1})\|_\infty \le e_k$, and the actual weights are $\boldsymbol{w}_{k+1} = \Pi_\Delta(\boldsymbol{e}_m + c\hat{\boldsymbol{\ell}}_k)$. The primal solution $\hat{\boldsymbol{x}}_{k+1}$ satisfies (P) only up to a slack of $-\delta_k$.

Theorem 5.8 (Energy inequality and certificates with errors)**.** *Let $a = a_{k+1}$ be computed from the actual weights. The energy inequality becomes*

$$\|\hat{\boldsymbol{x}}_{k+1} - \boldsymbol{x}_N\|^2 + \|\Delta \boldsymbol{x}_k\|^2 + \tfrac{2}{c\eta}\|\boldsymbol{w}_{k+1} - \boldsymbol{e}_m\|^2 \;\le\; \|\boldsymbol{x}_k - \boldsymbol{x}_N\|^2 + \tfrac{2}{\eta}(\delta_k + a\, e_k),$$

and the two certificates of Theorem 5.5 become

$$\begin{aligned} \lambda_k \mathrm{Expl}(\hat{\boldsymbol{x}}_{k+1}) \;&\le\; \tfrac{1}{2}a + \eta D\|\Delta \boldsymbol{x}_k\| + \delta_k + a e_k, \\ \Big[\max_{i\in P} \ell_i(\hat{\boldsymbol{x}}_{k+1}) - \tfrac{1}{2}\Big]^+ \;&\le\; \tfrac{\sqrt{2}}{c}\|\boldsymbol{w}_{k+1} - \boldsymbol{e}_m\| + \eta D\|\Delta \boldsymbol{x}_k\| + \delta_k + 2e_k. \end{aligned}$$

The proof is in Appendix A.8. When the errors are summable, all convergence results are preserved; when they are constant, we only obtain an $O(1/K + (\delta + e)/\eta)$ bound on the minimum per-step spending, not on the eventual iterates (Appendix A.8). Estimation errors enter the energy inequality and the latest certificate only through $a\, e_k$, so their effect shrinks as the pool empties near convergence.

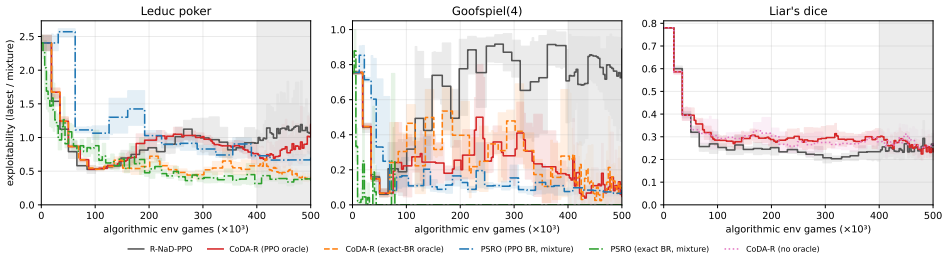


Figure 1: Full-tree exploitability at a matched budget (seed mean ± 95% CI; gray: last 20% of the budget). PSRO deploys a multi-network mixture; all other methods deploy the single latest network.

6 Experiments

6.1 Finite-Dimensional Verification

We run the exact algorithm on random skew-symmetric matrix games (up to 64 pure strategies, 1024 steps per cell) and check every inequality of Section 5 at every step. Nine groups comprise 3830 cells, 2880 with injected estimation and solver errors. Among about 3.7 million per-step records in the main cells there is not a single violation (Appendix Figure 3); the final $\mathrm{Expl}(\boldsymbol{x}_K)$ lies between 10^{-12} and 10^{-5}, and all pool weights go to zero. Breaking any of three mechanisms (a moving dual anchor, a stale mirror snapshot, or frozen pool coefficients) causes thousands of energy-inequality violations, so the checks do detect wrong algorithms. From the counterexample of Proposition 3.2, the fixed anchor converges within a few steps, while the moving anchor stays at $\mathrm{Expl} = 0.5$ and neither the explicit mirror nor $\lambda \equiv 0$ converges (Appendix Figure 2).

6.2 Neural Experiments at a Matched Budget

We implement CoDA-R on PPO (Schulman et al., 2017), with each exact object of §4 replaced by a runnable approximation whose error enters e_k or δ_k (details in Appendix C), and compare it with R-NaD-PPO and PSRO on Leduc poker (Southey et al., 2005), Goofspiel(4) (Ross, 1971), and Liar's Dice (one die each) (Lisý et al., 2015) in OpenSpiel (Lanctot et al., 2019). All PPO methods share the network, the PPO hyperparameters, and η (a mechanism comparison, not one of individually tuned methods), and the same budget of $B = 5 \times 10^5$ games, counting all games consumed, oracle games included. We report full-tree exact exploitability (in chips for Leduc) through its mean `tail_mean` and maximum `tail_worst` over the last 20% of the budget, as the IQM and bootstrap 95% CI over 4 seeds (Agarwal et al., 2021). R-NaD-PPO and CoDA-R deploy the single latest network, while PSRO deploys a whole-game mixture of networks, which are not directly interchangeable.

Table 1 and Figure 1 give the main results. **Goofspiel(4).** R-NaD-PPO reaches its lowest point at about 10^5 games and then degrades to 0.8–0.9; its policy entropy falls below 10^{-4} after 2×10^5 games (Appendix Figure 5), so the nearly deterministic policy is fully countered. This is exactly the phase in which the pool of CoDA-R gains weight, and the average gain of oracle admissions against latest is 0.13. The tail mean of CoDA-R is 0.083, better than R-NaD-PPO on all 4 seeds, although its learner plays about 14% fewer games. The PSRO (PPO BR) mixture reaches 0.072, on par with CoDA-R, indicating that what decides this game is whether the vulnerabilities revealed by the oracle are exploited (exact-BR PSRO uses no games; Appendix Table 4). The `tail_worst` of CoDA-R is 0.49, because weight estimates lag behind jumps in the true loss rates (Appendix C.1). **Liar's Dice.** The tails of all three methods lie within 0.25–0.27 with fully overlapping CIs, and the median oracle admission gain is 0.002.

Leduc and the dual strength. At $c = 6$, the payoff scale leaves the leads of the PPO exploiter too small to earn pool weight (tail 1.235 vs. 0.980). With scores normalized over a 26-chip span, 0.8 chips of exploitability is a loss-rate shift of only 0.03, so the closed-form weight $c\mu/2$ is below 0.1. In a pre-specified sweep that changes only c ($c \in \{12, 24, 48\}$, both oracles; Appendix Table 5), the IQM of `tail_mean` decreases in c under both oracles. At $c = 48$, the exact-BR version reaches 0.492, better than R-NaD-PPO on all 4 seeds; the PPO-oracle version reaches 0.746 and is better on 3 seeds, but its CIs overlap and its `at_budget` and AUC match R-NaD-PPO; its oracle

Table 1: Matched-budget comparison (IQM [95% CI], 4 seeds). L: single latest network; M: multi-network mixture. Δ: per-seed paired difference (method − R-NaD-PPO); wins: seeds on which the method is better. Bold: best single-network value per game.

Method	Deploy	tail_mean	tail_worst	Δtail_mean [CI] (wins)
Leduc				
R-NaD-PPO	L	0.980 [0.736, 1.684]	1.224 [1.038, 2.163]	—
CoDA-R PPO oracle, c=48	L	0.746 [0.646, 1.206]	1.217 [0.913, 1.712]	-0.284 [-0.938, $+0.470$] (3/4)
CoDA-R exact-BR oracle, c=48	L	**0.492** [0.400, 0.543]	**0.708** [0.569, 0.845]	$\mathbf{-0.508}$ [-1.163, -0.273] (4/4)
PSRO, PPO BR	M	0.656 [0.609, 0.783]	0.735 [0.647, 0.877]	-0.359 [-0.901, -0.057] (4/4)
Goofspiel(4)				
R-NaD-PPO	L	0.797 [0.561, 0.953]	0.901 [0.848, 0.994]	—
CoDA-R PPO oracle	L	**0.083** [0.037, 0.227]	0.488 [0.320, 0.595]	-0.659 [-0.916, -0.446] (4/4)
CoDA-R exact-BR oracle	L	0.115 [0.090, 0.186]	**0.483** [0.424, 0.638]	$\mathbf{-0.692}$ [-0.841, -0.375] (4/4)
PSRO, PPO BR	M	0.072 [0.055, 0.126]	0.095 [0.061, 0.181]	-0.705 [-0.866, -0.506] (4/4)
Liar's Dice				
R-NaD-PPO	L	0.259 [0.207, 0.274]	**0.309** [0.263, 0.406]	—
CoDA-R PPO oracle	L	0.265 [0.231, 0.280]	0.333 [0.286, 0.350]	$\mathbf{-0.001}$ [-0.028, $+0.073$] (2/4)
CoDA-R no oracle	L	**0.256** [0.227, 0.348]	0.342 [0.276, 0.382]	$+0.023$ [-0.031, $+0.074$] (2/4)

still finds almost no vulnerabilities, and we cannot tell how much of the extra weight at larger c comes from estimation noise. The $c = 48$ results in Table 1 use the evaluation seeds and count as a hyperparameter sweep.

6.3 Controlled Vulnerability Injection and Ablations

Controlled vulnerability injection. To test whether pool weight exits exactly when the real gain disappears, we inject two pre-trained attackers A and B in turn into a common R-NaD-PPO base on Goofspiel(4), and compare two CoDA-R branches (EMA or exact loss rates) against R-NaD-PPO and fixed-mix, which mixes in the attackers at fixed proportions (setup in Appendix C). In both CoDA-R branches, w_A and $\ell(\boldsymbol{x}; A) - \frac{1}{2}$ drop to 0 in sync, the true lead of A is at most 0.011 once $w_A < 0.01$, and the scatter hugs the closed form of Lemma 4.1(c). Fixed proportions, by contrast, are harmful. After B is injected, fixed-mix overfits the attackers and its tail exploitability rises to 0.45, versus 0.23 for R-NaD-PPO and 0.17 and 0.27 for the CoDA-R branches, and these three CIs overlap (the attackers lead by only about 0.05, so repair speed is not compared).

Ablations. In shorter runs, η is the most sensitive hyperparameter, and all pool-side ablations differ within the CIs. The moving-anchor CoDA is worse than CoDA-R on Kuhn poker (Kuhn, 1950), Leduc, and Goofspiel(4) (0.29/0.86/0.96 vs. 0.06/0.64/0.04; Appendix C). Because of errors, the certificates are unreliable in the neural setting, where `latest_cert` underestimates the truth in 63 of 72 oracle evaluations and does not beat last-checkpoint selection, since the KL proxy does not bound the theoretical term and the exploiter gap τ is unknown; in PPO they are only diagnostic.

7 Conclusion

By placing the moving anchor on the primal side and fixing the dual anchor at pure self-play, CoDA-R makes policy movement and pool weight share one budget, which yields single-network last-iterate convergence however opponents enter or leave the pool, with certificates computable during training. It thus combines the openness of population methods with the single-network guarantee of regularized self-play. Black-box opponents can enter or leave the pool at any time, convergence is always carried by the one network being trained, and the method is merely a wrapper around PPO. Empirically, with suitable hyperparameters, CoDA-R achieves a significant advantage on Goofspiel(4), where regularized self-play collapses and gets countered, matching a multi-network PSRO mixture with one network, and a considerable advantage on Leduc, while tying on Liar's Dice. Future work includes a Bregman (KL) version of the current Euclidean, normal-form theory, an end-to-end bound on δ_k, a last-iterate rate under Rule B, and larger games.

Reproducibility Statement

§3–5 give all definitions of the algorithm, and Appendix A gives all proofs. Anonymized source code is submitted as a zip file in the supplementary material, including the finite-dimensional solver, the PPO implementation, all baselines, the experiment entry points and all hyperparameter configurations, and the scripts that generate all figures and tables.

AI Use Statement

In this work, we used generative AI tools to assist with drafting and polishing the text of the paper, writing and refactoring experiment code, formatting references, and aggregating experiment logs to generate figures and tables. Regarding research content, AI tools were used to refine the formulation of research hypotheses, to assist with derivations and the checking of proof steps, and to assist with interpreting experimental results. The core algorithm, theorem statements, and research design were proposed by the authors. We did not use generative AI to generate synthetic data or to fabricate experimental results. All AI-assisted code was tested and type-checked by the authors, and all AI-assisted proofs, interpretations, and text were manually reviewed item by item by the authors. The authors take responsibility for the entire content of this paper.

References

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron Courville, and Marc G. Bellemare. Deep reinforcement learning at the edge of the statistical precipice. In *Advances in Neural Information Processing Systems 34 (NeurIPS)*, pp. 29304–29320, 2021.

James P. Bailey and Georgios Piliouras. Multiplicative weights update in zero-sum games. In *Proceedings of the 2018 ACM Conference on Economics and Computation (EC)*, pp. 321–338, 2018.

David Balduzzi, Sébastien Racanière, James Martens, Jakob Foerster, Karl Tuyls, and Thore Graepel. The mechanics of n-player differentiable games. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, volume 80 of *Proceedings of Machine Learning Research*, pp. 354–363. PMLR, 2018.

David Balduzzi, Marta Garnelo, Yoram Bachrach, Wojciech M. Czarnecki, Julien Pérolat, Max Jaderberg, and Thore Graepel. Open-ended learning in symmetric zero-sum games. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, volume 97 of *Proceedings of Machine Learning Research*, pp. 434–443. PMLR, 2019.

Trapit Bansal, Jakub Pachocki, Szymon Sidor, Ilya Sutskever, and Igor Mordatch. Emergent complexity via multi-agent competition. In *International Conference on Learning Representations (ICLR)*, 2018.

Heinz H. Bauschke and Patrick L. Combettes. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. Springer, New York, 2011.

Noam Brown and Tuomas Sandholm. Superhuman AI for heads-up no-limit poker: Libratus beats top professionals. *Science*, 359(6374):418–424, 2018.

Shicong Cen, Yuting Wei, and Yuejie Chi. Fast policy extragradient methods for competitive games with entropy regularization. In *Advances in Neural Information Processing Systems 34 (NeurIPS)*, pp. 27952–27964, 2021.

Wojciech M. Czarnecki, Gauthier Gidel, Brendan Tracey, Karl Tuyls, Shayegan Omidshafiei, David Balduzzi, and Max Jaderberg. Real world games look like spinning tops. In *Advances in Neural Information Processing Systems 33 (NeurIPS)*, pp. 17443–17454, 2020.

Constantinos Daskalakis, Andrew Ilyas, Vasilis Syrgkanis, and Haoyang Zeng. Training GANs with optimism. In *International Conference on Learning Representations (ICLR)*, 2018.

Le Cong Dinh, Stephen McAleer, Zheng Tian, Nicolas Perez-Nieves, Oliver Slumbers, David Henry Mguni, Jun Wang, Haitham Bou Ammar, and Yaodong Yang. Online Double Oracle. *Transactions on Machine Learning Research*, 2022.

John Duchi, Shai Shalev-Shwartz, Yoram Singer, and Tushar Chandra. Efficient projections onto the ℓ_1-ball for learning in high dimensions. In *Proceedings of the 25th International Conference on Machine Learning (ICML)*, pp. 272–279, 2008.

Francisco Facchinei and Jong-Shi Pang. *Finite-Dimensional Variational Inequalities and Complementarity Problems*. Springer, New York, 2003.

Yoav Freund and Robert E. Schapire. Adaptive game playing using multiplicative weights. *Games and Economic Behavior*, 29(1–2):79–103, 1999.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems 27 (NIPS)*, 2014.

Philip Hartman and Guido Stampacchia. On some non-linear elliptic differential-functional equations. *Acta Mathematica*, 115:271–310, 1966.

Johannes Heinrich and David Silver. Deep reinforcement learning from self-play in imperfect-information games. *arXiv preprint arXiv:1603.01121*, 2016.

Johannes Heinrich, Marc Lanctot, and David Silver. Fictitious self-play in extensive-form games. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, volume 37 of *Proceedings of Machine Learning Research*, pp. 805–813. PMLR, 2015.

Mark Herbster and Manfred K. Warmuth. Tracking the best expert. *Machine Learning*, 32(2): 151–178, 1998.

Harold W. Kuhn. A simplified two-person poker. In Harold W. Kuhn and Albert W. Tucker (eds.), *Contributions to the Theory of Games*, volume 1, pp. 97–103. Princeton University Press, 1950.

Harold W. Kuhn. Extensive games and the problem of information. In Harold W. Kuhn and Albert W. Tucker (eds.), *Contributions to the Theory of Games*, volume 2, pp. 193–216. Princeton University Press, 1953.

Marc Lanctot, Vinicius Zambaldi, Audrunas Gruslys, Angeliki Lazaridou, Karl Tuyls, Julien Pérolat, David Silver, and Thore Graepel. A unified game-theoretic approach to multiagent reinforcement learning. In *Advances in Neural Information Processing Systems 30 (NIPS)*, 2017.

Marc Lanctot, Edward Lockhart, Jean-Baptiste Lespiau, Vinicius Zambaldi, Satyaki Upadhyay, Julien Pérolat, Sriram Srinivasan, Finbarr Timbers, Karl Tuyls, Shayegan Omidshafiei, Daniel Hennes, Dustin Morrill, Paul Muller, Timo Ewalds, Ryan Faulkner, János Kramár, Bart De Vylder, Brennan Saeta, James Bradbury, David Ding, Sebastian Borgeaud, Matthew Lai, Julian Schrittwieser, Thomas Anthony, Edward Hughes, Ivo Danihelka, and Jonah Ryan-Davis. OpenSpiel: A framework for reinforcement learning in games. *arXiv preprint arXiv:1908.09453*, 2019.

Viliam Lisý, Marc Lanctot, and Michael Bowling. Online Monte Carlo counterfactual regret minimization for search in imperfect information games. In *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pp. 27–36, 2015.

Stephen McAleer, John Lanier, Roy Fox, and Pierre Baldi. Pipeline PSRO: A scalable approach for finding approximate Nash equilibria in large games. In *Advances in Neural Information Processing Systems 33 (NeurIPS)*, pp. 20238–20248, 2020.

Stephen McAleer, Kevin Wang, John Lanier, Marc Lanctot, Pierre Baldi, Tuomas Sandholm, and Roy Fox. Anytime PSRO for two-player zero-sum games. In *AAAI Workshop on Reinforcement Learning in Games*, 2022. arXiv:2201.07700.

H. Brendan McMahan, Geoffrey J. Gordon, and Avrim Blum. Planning in the presence of cost functions controlled by an adversary. In *Proceedings of the 20th International Conference on Machine Learning (ICML)*, pp. 536–543, 2003.

Panayotis Mertikopoulos, Bruno Lecouat, Houssam Zenati, Chuan-Sheng Foo, Vijay Chandrasekhar, and Georgios Piliouras. Optimistic mirror descent in saddle-point problems: Going the extra (gradient) mile. In *International Conference on Learning Representations (ICLR)*, 2019.

Aryan Mokhtari, Asuman Ozdaglar, and Sarath Pattathil. A unified analysis of extra-gradient and optimistic gradient methods for saddle point problems: Proximal point approach. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 108 of *Proceedings of Machine Learning Research*, pp. 1497–1507. PMLR, 2020.

Paul Muller, Shayegan Omidshafiei, Mark Rowland, Karl Tuyls, Julien Pérolat, Siqi Liu, Daniel Hennes, Luke Marris, Marc Lanctot, Edward Hughes, Zhe Wang, Guy Lever, Nicolas Heess, Thore Graepel, and Rémi Munos. A generalized training approach for multiagent learning. In *International Conference on Learning Representations (ICLR)*, 2020.

Arkadi Nemirovski. Prox-method with rate of convergence $O(1/t)$ for variational inequalities with Lipschitz continuous monotone operators and smooth convex-concave saddle point problems. *SIAM Journal on Optimization*, 15(1):229–251, 2004.

OpenAI, Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Dębiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, Rafal Józefowicz, Scott Gray, Catherine Olsson, Jakub Pachocki, Michael Petrov, Henrique Pondé de Oliveira Pinto, Jonathan Raiman, Tim Salimans, Jeremy Schlatter, Jonas Schneider, Szymon Sidor, Ilya Sutskever, Jie Tang, Filip Wolski, and Susan Zhang. Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680*, 2019.

Julien Pérolat, Rémi Munos, Jean-Baptiste Lespiau, Shayegan Omidshafiei, Mark Rowland, Pedro Ortega, Neil Burch, Thomas Anthony, David Balduzzi, Bart De Vylder, Georgios Piliouras, Marc Lanctot, and Karl Tuyls. From Poincaré recurrence to convergence in imperfect information games: Finding equilibrium via regularization. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8525–8535. PMLR, 2021.

Julien Pérolat, Bart De Vylder, Daniel Hennes, Eugene Tarassov, Florian Strub, Vincent de Boer, Paul Muller, Jerome T. Connor, Neil Burch, Thomas Anthony, Stephen McAleer, Romuald Elie, Sarah H. Cen, Zhe Wang, Audrunas Gruslys, Aleksandra Malysheva, Mina Khan, Sherjil Ozair, Finbarr Timbers, Toby Pohlen, Tom Eccles, Mark Rowland, Marc Lanctot, Jean-Baptiste Lespiau, Bilal Piot, Shayegan Omidshafiei, Edward Lockhart, Laurent Sifre, Nathalie Beauguerlange, Rémi Munos, David Silver, Satinder Singh, Demis Hassabis, and Karl Tuyls. Mastering the game of Stratego with model-free multiagent reinforcement learning. *Science*, 378(6623):990–996, 2022.

R. Tyrrell Rockafellar. Monotone operators and the proximal point algorithm. *SIAM Journal on Control and Optimization*, 14(5):877–898, 1976.

Sheldon M. Ross. Goofspiel—the game of pure strategy. *Journal of Applied Probability*, 8(3): 621–625, 1971.

John Schulman, Philipp Moritz, Sergey Levine, Michael I. Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. In *International Conference on Learning Representations (ICLR)*, 2016.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, Timothy Lillicrap, Karen Simonyan, and Demis Hassabis. A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. *Science*, 362(6419):1140–1144, 2018.

Maurice Sion. On general minimax theorems. *Pacific Journal of Mathematics*, 8(1):171–176, 1958.

Samuel Sokota, Ryan D'Orazio, J. Zico Kolter, Nicolas Loizou, Marc Lanctot, Ioannis Mitliagkas, Noam Brown, and Christian Kroer. A unified approach to reinforcement learning, quantal response equilibria, and two-player zero-sum games. In *International Conference on Learning Representations (ICLR)*, 2023.

Finnegan Southey, Michael Bowling, Bryce Larson, Carmelo Piccione, Neil Burch, Darse Billings, and Chris Rayner. Bayes' bluff: Opponent modelling in poker. In *Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence (UAI)*, pp. 550–558, 2005.

Oriol Vinyals, Igor Babuschkin, Wojciech M. Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H. Choi, Richard Powell, Timo Ewalds, Petko Georgiev, Junhyuk Oh, Dan Horgan, Manuel Kroiss, Ivo Danihelka, Aja Huang, Laurent Sifre, Trevor Cai, John P. Agapiou, Max Jaderberg, Alexander S. Vezhnevets, Rémi Leblond, Tobias Pohlen, Valentin Dalibard, David Budden, Yury Sulsky, James Molloy, Tom L. Paine, Caglar Gulcehre, Ziyu Wang, Tobias Pfaff, Yuhuai Wu, Roman Ring, Dani Yogatama, Dario Wünsch, Katrina McKinney, Oliver Smith, Tom Schaul, Timothy Lillicrap, Koray Kavukcuoglu, Demis Hassabis, Chris Apps, and David Silver. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature*, 575(7782): 350–354, 2019.

John von Neumann. Zur Theorie der Gesellschaftsspiele. *Mathematische Annalen*, 100(1):295–320, 1928.

Michael P. Wellman. Methods for empirical game-theoretic analysis (extended abstract). In *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI)*, pp. 1552–1555, 2006.

A PROOFS

Apart from the minimax theorem and the Hartman–Stampacchia theorem, all proofs use only three tools, namely the polarization identity $\|\boldsymbol{u}+\boldsymbol{v}\|^2 = \|\boldsymbol{u}\|^2+\|\boldsymbol{v}\|^2+2\boldsymbol{u}^\top\boldsymbol{v}$, completing the square, and the obtuse-angle principle $\boldsymbol{w} = \Pi_C(\boldsymbol{v}) \iff (\boldsymbol{v}-\boldsymbol{w})^\top(\boldsymbol{u}-\boldsymbol{w}) \le 0 \ \forall \boldsymbol{u} \in C$. The obtuse-angle principle also yields the monotonicity and nonexpansiveness of the projection. For $\boldsymbol{w} = \Pi_C(\boldsymbol{z})$ and $\boldsymbol{w}' = \Pi_C(\boldsymbol{z}')$, taking $\boldsymbol{u} = \boldsymbol{w}'$ and $\boldsymbol{u} = \boldsymbol{w}$ respectively and adding gives

$$(\boldsymbol{z}-\boldsymbol{z}')^\top(\boldsymbol{w}-\boldsymbol{w}') \ \ge \ \|\boldsymbol{w}-\boldsymbol{w}'\|^2 \ \ge \ 0. \tag{9}$$

A.1 PROPOSITION 3.1: CODA CONVERGES IN THE RESTRICTED GAME

Let $L_k(\boldsymbol{x},\boldsymbol{\rho}) := \boldsymbol{\rho}^\top\boldsymbol{\ell}(\boldsymbol{x}) - \frac{1}{2c}\|\boldsymbol{\rho}-\boldsymbol{\rho}_k\|^2$. It is affine in $\boldsymbol{x}$ and strongly concave in $\boldsymbol{\rho}$, and X and Δ_N are convex and compact, so by the minimax theorem of Sion (1958) a saddle point exists, and since the regularizer does not depend on $\boldsymbol{x}$ it can be pulled outside $\min_{\boldsymbol{x}}$:

$$\min_{\boldsymbol{x}}\max_{\boldsymbol{\rho}} L_k(\boldsymbol{x},\boldsymbol{\rho}) = \max_{\boldsymbol{\rho}}\Big[\min_{\boldsymbol{x}}\boldsymbol{\rho}^\top\boldsymbol{\ell}(\boldsymbol{x}) - \tfrac{1}{2c}\|\boldsymbol{\rho}-\boldsymbol{\rho}_k\|^2\Big] = \max_{\boldsymbol{\rho}\in\Delta_N}\Big[\varphi(\boldsymbol{\rho}) - \tfrac{1}{2c}\|\boldsymbol{\rho}-\boldsymbol{\rho}_k\|^2\Big].$$

Hence $\boldsymbol{\rho}_{k+1} = \boldsymbol{w}_{k+1}$ is a proximal point step on the concave function φ, and the saddle-point property gives $\boldsymbol{x}_{k+1} \in \arg\min_{\boldsymbol{x}} \boldsymbol{w}_{k+1}^\top\boldsymbol{\ell}(\boldsymbol{x})$.

Monotone ascent. $\boldsymbol{\rho}_{k+1}$ is an optimal solution of the problem above and $\boldsymbol{\rho}_k$ is feasible, so

$$\varphi(\boldsymbol{\rho}_{k+1}) - \tfrac{1}{2c}\|\boldsymbol{\rho}_{k+1}-\boldsymbol{\rho}_k\|^2 \ \ge \ \varphi(\boldsymbol{\rho}_k) - 0,$$

which is the first inequality of the proposition.

Finite energy. Weak duality gives $0 \le \varphi(\boldsymbol{\rho}) \le v^* \le 1$. Telescoping the first inequality,

$$\tfrac{1}{2c}\sum_{k<K}\|\boldsymbol{\rho}_{k+1}-\boldsymbol{\rho}_k\|^2 \ \le \ \varphi(\boldsymbol{\rho}_K) - \varphi(\boldsymbol{\rho}_0) \ \le \ 1,$$

so $\sum_k \|\boldsymbol{\rho}_{k+1}-\boldsymbol{\rho}_k\|^2 \le 2c$; the series converges and its terms tend to zero.

In-pool certificate. Given $\boldsymbol{x}_{k+1}$, completing the square gives $\boldsymbol{\rho}_{k+1} = \Pi_{\Delta_N}(\boldsymbol{\rho}_k + c\,\boldsymbol{\ell}(\boldsymbol{x}_{k+1}))$. Apply the obtuse-angle principle with $\boldsymbol{u} = \boldsymbol{e}_j$, $j \in \arg\max_i \ell_i(\boldsymbol{x}_{k+1})$:

$$\begin{aligned}&\big(\boldsymbol{\rho}_k + c\,\boldsymbol{\ell}(\boldsymbol{x}_{k+1}) - \boldsymbol{\rho}_{k+1}\big)^\top(\boldsymbol{e}_j - \boldsymbol{\rho}_{k+1}) \le 0\\ \iff\ & c\,\boldsymbol{\ell}(\boldsymbol{x}_{k+1})^\top(\boldsymbol{e}_j - \boldsymbol{\rho}_{k+1}) \ \le \ (\boldsymbol{\rho}_{k+1}-\boldsymbol{\rho}_k)^\top(\boldsymbol{e}_j - \boldsymbol{\rho}_{k+1}).\end{aligned}$$

By Cauchy–Schwarz and the diameter of the simplex, the right-hand side is at most $\sqrt{2}\,\|\boldsymbol{\rho}_{k+1}-\boldsymbol{\rho}_k\|$; the left-hand side equals $c[\max_i \ell_i(\boldsymbol{x}_{k+1})-\boldsymbol{\rho}_{k+1}^\top \boldsymbol{\ell}(\boldsymbol{x}_{k+1})]$. The best response gives $\boldsymbol{\rho}_{k+1}^\top \boldsymbol{\ell}(\boldsymbol{x}_{k+1}) = \min_{\boldsymbol{x}} \boldsymbol{\rho}_{k+1}^\top \boldsymbol{\ell}(\boldsymbol{x}) = \varphi(\boldsymbol{\rho}_{k+1}) \le v^*$, so

$$\max_i \ell_i(\boldsymbol{x}_{k+1}) - v^* \;\le\; \max_i \ell_i(\boldsymbol{x}_{k+1}) - \varphi(\boldsymbol{\rho}_{k+1}) \;\le\; \tfrac{\sqrt{2}}{c}\|\boldsymbol{\rho}_{k+1}-\boldsymbol{\rho}_k\|.$$

Rate. For any $\boldsymbol{u} \in \Delta_N$ and $t \in (0,1]$, $\boldsymbol{u}_t := \boldsymbol{\rho}_{k+1} + t(\boldsymbol{u}-\boldsymbol{\rho}_{k+1})$ is feasible, and the optimality of $\boldsymbol{\rho}_{k+1}$ gives

$$\varphi(\boldsymbol{u}_t) - \tfrac{1}{2c}\|\boldsymbol{u}_t-\boldsymbol{\rho}_k\|^2 \;\le\; \varphi(\boldsymbol{\rho}_{k+1}) - \tfrac{1}{2c}\|\boldsymbol{\rho}_{k+1}-\boldsymbol{\rho}_k\|^2.$$

Using the concavity of φ, $\varphi(\boldsymbol{u}_t) \ge (1-t)\varphi(\boldsymbol{\rho}_{k+1}) + t\varphi(\boldsymbol{u})$, expanding the quadratic term, dividing by t, and letting $t \to 0^+$, we obtain

$$\varphi(\boldsymbol{u}) - \varphi(\boldsymbol{\rho}_{k+1}) \;\le\; \tfrac{1}{c}(\boldsymbol{\rho}_{k+1}-\boldsymbol{\rho}_k)^\top(\boldsymbol{u}-\boldsymbol{\rho}_{k+1}).$$

Taking $\boldsymbol{u} = \boldsymbol{\rho}^* \in \arg\max \varphi$ and using the polarization identity $2(\boldsymbol{\rho}_{k+1}-\boldsymbol{\rho}_k)^\top(\boldsymbol{\rho}^*-\boldsymbol{\rho}_{k+1}) = \|\boldsymbol{\rho}^*-\boldsymbol{\rho}_k\|^2 - \|\boldsymbol{\rho}^*-\boldsymbol{\rho}_{k+1}\|^2 - \|\boldsymbol{\rho}_{k+1}-\boldsymbol{\rho}_k\|^2$,

$$v^* - \varphi(\boldsymbol{\rho}_{k+1}) \;\le\; \tfrac{1}{2c}\big(\|\boldsymbol{\rho}^*-\boldsymbol{\rho}_k\|^2 - \|\boldsymbol{\rho}^*-\boldsymbol{\rho}_{k+1}\|^2\big),$$

and telescoping over $k = 0, \dots, K-1$ gives $\sum_{k=1}^{K}[v^* - \varphi(\boldsymbol{\rho}_k)] \le \|\boldsymbol{\rho}^*-\boldsymbol{\rho}_0\|^2/(2c) \le 1/c$. By monotone ascent, $v^* - \varphi(\boldsymbol{\rho}_k)$ is nonincreasing, and the smallest term is at most the average, so $v^* - \varphi(\boldsymbol{\rho}_K) \le 1/(cK)$. This bound is independent of N, whereas the corresponding bound for Hedge contains a $\log N$ factor. ■

A.2 Verification of Proposition 3.2

The pool member ties with latest:

$$\boldsymbol{e}_1^\top S \boldsymbol{y} = S_{12}\cdot\tfrac{1}{2} + S_{13}\cdot\tfrac{1}{2} = \tfrac{1}{2}\cdot(-1)\cdot\tfrac{1}{2} + \tfrac{1}{2}\cdot 1\cdot\tfrac{1}{2} = 0,$$

so $\ell_1(\boldsymbol{e}_1) = \frac{1}{2}$, the mirror coordinate is also $\frac{1}{2}$, and $c\tilde{\boldsymbol{\ell}} = c(\frac{1}{2},\frac{1}{2})$ is a constant vector. Adding a constant vector to a point of the simplex and then projecting returns the point itself (the projection threshold rises uniformly by the same amount), so $\Pi_\Delta(\boldsymbol{\rho} + c\tilde{\boldsymbol{\ell}}) = \boldsymbol{\rho}$ and the dual side stays put. The primal field is

$$S\big(\lambda \boldsymbol{e}_1 + \tfrac{2}{3}\boldsymbol{y}\big) = S\big(\tfrac{1}{3},\tfrac{1}{3},\tfrac{1}{3}\big)^\top = 0,$$

because the uniform distribution is the Nash strategy of RPS and S maps it to zero; with anchor $\boldsymbol{x}_k = \boldsymbol{x}$ the proximal term is also zero, so the primal residual is zero. Yet $\mathrm{Expl}(\boldsymbol{e}_1) = \max_j S_{1j} = \frac{1}{2}$: always playing rock is fully countered by paper. ■

A.3 Lemma 4.1: Properties of the Dual Rule

Let $\boldsymbol{z} := \boldsymbol{e}_m + c\tilde{\boldsymbol{\ell}}(\boldsymbol{x})$ and $\boldsymbol{w} = \Pi_\Delta(\boldsymbol{z})$.

(a) The obtuse-angle principle says $\boldsymbol{e}_m = \Pi_\Delta(\boldsymbol{z})$ if and only if $(\boldsymbol{z}-\boldsymbol{e}_m)^\top(\boldsymbol{u}-\boldsymbol{e}_m) \le 0$ for all $\boldsymbol{u} \in \Delta$. Since $\boldsymbol{z}-\boldsymbol{e}_m = c\tilde{\boldsymbol{\ell}}$, the condition becomes $\tilde{\boldsymbol{\ell}}^\top \boldsymbol{u} \le \tilde{\boldsymbol{\ell}}^\top \boldsymbol{e}_m = \frac{1}{2}$ for all $\boldsymbol{u}$; a linear function attains its maximum over the simplex at a vertex, so this means $\max_j \tilde{\ell}_j \le \frac{1}{2}$. The mirror coordinate is exactly $\frac{1}{2}$, so this is equivalent to $\max_{i\in P} \ell_i \le \frac{1}{2}$.

(b) Move a little weight from member i to the mirror by taking $\boldsymbol{u} = \boldsymbol{w} + \varepsilon(\boldsymbol{e}_m - \boldsymbol{e}_i)$ with $0 < \varepsilon \le w_i$, which remains in Δ. The obtuse-angle principle gives

$$(\boldsymbol{z}-\boldsymbol{w})^\top(\boldsymbol{u}-\boldsymbol{w}) = \varepsilon\big[(z_m-w_m)-(z_i-w_i)\big] \le 0 \iff c\,\ell_i - w_i \;\ge\; z_m - w_m = 1 + \tfrac{c}{2} - w_m,$$

i.e., $c(\ell_i - \frac{1}{2}) \ge 1 - w_m + w_i \ge w_i > 0$.

(c) The points of Δ_2 are $(s, 1-s)$ with $s \in [0,1]$, and their squared distance to $\boldsymbol{z} = (\frac{c}{2} + c\mu,\ 1 + \frac{c}{2})$,

$$(\tfrac{c}{2} + c\mu - s)^2 + (\tfrac{c}{2} + s)^2,$$

is a convex quadratic function of s with unconstrained minimizer $s = \frac{c\mu}{2}$; the minimizer over $[0,1]$ is obtained by clipping it to the interval. ■

With multiple members, the weights are determined by a common projection threshold, and there is no per-member closed form. For example, with $c = 4$ and both members at loss rate 0.9, $\boldsymbol{z} = (3.6, 3.6, 3)$ and the candidate is $\boldsymbol{w} = (\frac{1}{2}, \frac{1}{2}, 0)$; for any $\boldsymbol{u} \in \Delta_3$, $(\boldsymbol{z} - \boldsymbol{w})^\top(\boldsymbol{u} - \boldsymbol{w}) = 3.1(u_1 + u_2 - 1) + 3u_3 = -0.1u_3 \le 0$, so by the obtuse-angle principle it is indeed the projection, and the mirror weight is zero.

A.4 Unique Solution at Each Step

Proposition A.1 (Unique solution at each step)**.** *For any anchor $\boldsymbol{x}_k$, any $\lambda_k \ge 0$, and any pool,* (P) *has one and only one solution.*

Uniqueness comes from three facts, namely that the self-play term is skew-symmetric, the pool term is monotone (by the monotonicity of the projection), and the proximal term is η-strongly monotone; existence follows from the Hartman–Stampacchia theorem. Denote the field $G_k(\boldsymbol{x}) := \lambda_k S\boldsymbol{x} + A\boldsymbol{w}(\boldsymbol{x})_P$.

G_k is monotone.

Self-play term: $(\boldsymbol{x} - \boldsymbol{x}')^\top S(\boldsymbol{x} - \boldsymbol{x}') = 0$, so $\langle \lambda_k S(\boldsymbol{x} - \boldsymbol{x}'), \boldsymbol{x} - \boldsymbol{x}' \rangle = 0$.

Pool term: for $\boldsymbol{z} = \boldsymbol{e}_m + c\tilde{\boldsymbol{\ell}}(\boldsymbol{x})$ and $\boldsymbol{z}' = \boldsymbol{e}_m + c\tilde{\boldsymbol{\ell}}(\boldsymbol{x}')$, the mirror coordinates coincide and the pool coordinates differ by $c(\ell_i(\boldsymbol{x}) - \ell_i(\boldsymbol{x}')) = c\,(S\boldsymbol{y}_i)^\top(\boldsymbol{x} - \boldsymbol{x}')$, i.e., $(\boldsymbol{z} - \boldsymbol{z}')_P = c\,A^\top(\boldsymbol{x} - \boldsymbol{x}')$ and $(\boldsymbol{z} - \boldsymbol{z}')_m = 0$. By (9),

$$0 \le (\boldsymbol{z} - \boldsymbol{z}')^\top \big(\boldsymbol{w}(\boldsymbol{x}) - \boldsymbol{w}(\boldsymbol{x}')\big) = (\boldsymbol{z} - \boldsymbol{z}')_P^\top \big(\boldsymbol{w}(\boldsymbol{x}) - \boldsymbol{w}(\boldsymbol{x}')\big)_P = c\,(\boldsymbol{x} - \boldsymbol{x}')^\top A\big(\boldsymbol{w}(\boldsymbol{x}) - \boldsymbol{w}(\boldsymbol{x}')\big)_P,$$

i.e., $\langle A\boldsymbol{w}(\boldsymbol{x})_P - A\boldsymbol{w}(\boldsymbol{x}')_P, \boldsymbol{x} - \boldsymbol{x}' \rangle \ge 0$. After adding $\eta(\boldsymbol{x} - \boldsymbol{x}_k)$, the whole field satisfies $\langle \cdot, \boldsymbol{x} - \boldsymbol{x}' \rangle \ge \eta\|\boldsymbol{x} - \boldsymbol{x}'\|^2$, i.e., it is η-strongly monotone. This uses the fact that λ_k is constant within the step; otherwise the self-play term would no longer be a skew-symmetric quadratic form.

Uniqueness. Suppose $\boldsymbol{x}$ and $\boldsymbol{x}'$ both solve (P). Take the test point $\boldsymbol{x}'$ in the inequality for $\boldsymbol{x}$, and the test point $\boldsymbol{x}$ in the inequality for $\boldsymbol{x}'$:

$$\langle G_k(\boldsymbol{x}) + \eta(\boldsymbol{x} - \boldsymbol{x}_k),\, \boldsymbol{x}' - \boldsymbol{x} \rangle \ge 0, \qquad \langle G_k(\boldsymbol{x}') + \eta(\boldsymbol{x}' - \boldsymbol{x}_k),\, \boldsymbol{x} - \boldsymbol{x}' \rangle \ge 0.$$

Adding them gives $\eta\|\boldsymbol{x} - \boldsymbol{x}'\|^2 \le -\langle G_k(\boldsymbol{x}) - G_k(\boldsymbol{x}'), \boldsymbol{x} - \boldsymbol{x}' \rangle \le 0$, so $\boldsymbol{x} = \boldsymbol{x}'$.

Existence. $\tilde{\boldsymbol{\ell}}$ is affine and $\boldsymbol{w}(\boldsymbol{x})$ is continuous by the nonexpansiveness in (9), so G_k is continuous; X is compact and convex. A variational inequality with a continuous field on a compact convex set has a solution, which is the Hartman–Stampacchia theorem (Hartman & Stampacchia, 1966; Facchinei & Pang, 2003). ■

A.5 Lemmas 5.1 and 5.2, Theorems 5.3 and 5.6

Proof of Lemma 5.1. $\boldsymbol{w} = \Pi_\Delta(\boldsymbol{e}_m + c\tilde{\boldsymbol{\ell}})$. Apply the obtuse-angle principle with $\boldsymbol{u} = \boldsymbol{e}_m \in \Delta$:

$$\big(\boldsymbol{e}_m + c\tilde{\boldsymbol{\ell}} - \boldsymbol{w}\big)^\top(\boldsymbol{e}_m - \boldsymbol{w}) \le 0 \iff \|\boldsymbol{e}_m - \boldsymbol{w}\|^2 + c\,\tilde{\boldsymbol{\ell}}^\top(\boldsymbol{e}_m - \boldsymbol{w}) \le 0.$$

By (7), $\tilde{\boldsymbol{\ell}}^\top \boldsymbol{e}_m = \frac{1}{2}$ and $\tilde{\boldsymbol{\ell}}^\top \boldsymbol{w} = b + \frac{1}{2}$, so $\tilde{\boldsymbol{\ell}}^\top(\boldsymbol{e}_m - \boldsymbol{w}) = -b$. Hence $\|\boldsymbol{w} - \boldsymbol{e}_m\|^2 \le c\,b$. ■

Proof of Lemma 5.2. Take $\boldsymbol{x}' = \boldsymbol{x}_N$ in (P) and combine the first two terms into $S\boldsymbol{v}$:

$$\langle S\boldsymbol{v},\, \boldsymbol{x}_N - \boldsymbol{x} \rangle + \eta\,\Delta\boldsymbol{x}_k^\top(\boldsymbol{x}_N - \boldsymbol{x}) \ge 0, \qquad \langle S\boldsymbol{v},\, \boldsymbol{x}_N - \boldsymbol{x} \rangle = \boldsymbol{x}_N^\top S\boldsymbol{v} - \boldsymbol{x}^\top S\boldsymbol{v} = \boldsymbol{x}_N^\top S\boldsymbol{v} - b.$$

$\boldsymbol{v} = \lambda\boldsymbol{x} + \sum_i w_i \boldsymbol{y}_i$ is a nonnegative combination of points in X, and by (1) each term satisfies $\boldsymbol{x}_N^\top S\boldsymbol{z} \le 0$, so $\boldsymbol{x}_N^\top S\boldsymbol{v} \le 0$. Hence $\eta\,\Delta\boldsymbol{x}_k^\top(\boldsymbol{x}_N - \boldsymbol{x}) \ge b - \boldsymbol{x}_N^\top S\boldsymbol{v} \ge b$. ■

Proof of Theorem 5.3. The polarization identity gives

$$\|\boldsymbol{x}_k - \boldsymbol{x}_N\|^2 = \|\Delta\boldsymbol{x}_k\|^2 + \|\boldsymbol{x}_{k+1} - \boldsymbol{x}_N\|^2 + 2\,\Delta\boldsymbol{x}_k^\top(\boldsymbol{x}_N - \boldsymbol{x}_{k+1}),$$

and applying Lemma 5.2 and then Lemma 5.1 to the cross term,

$$2\,\Delta\boldsymbol{x}_k^\top(\boldsymbol{x}_N - \boldsymbol{x}_{k+1}) \;\ge\; \tfrac{2}{\eta}\,b(\boldsymbol{x}_{k+1}) \;\ge\; \tfrac{2}{c\eta}\|\boldsymbol{w}(\boldsymbol{x}_{k+1}) - \boldsymbol{e}_m\|^2.$$ ■

Proof of Theorem 5.6. Corollary 5.4 gives $a \to 0$ and $\|\Delta \boldsymbol{x}\| \to 0$; moreover, $\lambda_k \to 1$ under Rule B and $\lambda_k \equiv 1$ under Rule A, so $\lambda_k \geq \frac{1}{2}$ for large k, and substituting into Theorem 5.5(i) gives $\mathrm{Expl}(\boldsymbol{x}_{k+1}) \leq \frac{1}{\lambda_k}[\frac{1}{2}a + \eta D\|\Delta \boldsymbol{x}_k\|] \to 0$. Since X is compact and Expl is continuous, the limit $\bar{\boldsymbol{x}}$ of any convergent subsequence is Nash; Theorem 5.3 holds with the reference point $\bar{\boldsymbol{x}}$, so $\|\boldsymbol{x}_k - \bar{\boldsymbol{x}}\|$ is nonincreasing, and since it tends to zero along the subsequence, it tends to zero along the whole sequence. ■

A.6 Theorem 5.5: Two Certificates

(i) Take $\boldsymbol{y}_e \in \arg\max_{\boldsymbol{y}} \ell(\boldsymbol{x}; \boldsymbol{y})$; then $\boldsymbol{x}^\top S \boldsymbol{y}_e = \mathrm{Expl}(\boldsymbol{x})$, and skew-symmetry gives $\boldsymbol{y}_e^\top S \boldsymbol{x} = -\mathrm{Expl}(\boldsymbol{x})$. Take $\boldsymbol{x}' = \boldsymbol{y}_e$ in (P), let $\boldsymbol{d} := \boldsymbol{y}_e - \boldsymbol{x}$, and separate the three terms:

$$\underbrace{\lambda\,(S\boldsymbol{x})^\top \boldsymbol{d}}_{\text{self-play}} + \underbrace{(A\boldsymbol{w}_P)^\top \boldsymbol{d}}_{\text{pool}} + \underbrace{\eta\,\Delta \boldsymbol{x}^\top \boldsymbol{d}}_{\text{proximal}} \;\geq\; 0.$$

Self-play term: $(S\boldsymbol{x})^\top \boldsymbol{d} = \boldsymbol{y}_e^\top S\boldsymbol{x} - \boldsymbol{x}^\top S\boldsymbol{x} = -\mathrm{Expl}(\boldsymbol{x})$. Pool term: using $(S\boldsymbol{y}_i)^\top \boldsymbol{z} = \ell_i(\boldsymbol{z}) - \frac{1}{2}$,

$$(A\boldsymbol{w}_P)^\top \boldsymbol{d} = \sum_i w_i\big(\ell_i(\boldsymbol{y}_e) - \ell_i(\boldsymbol{x})\big) = \sum_i w_i\big(\ell_i(\boldsymbol{y}_e) - \tfrac{1}{2}\big) - b \;\leq\; \tfrac{1}{2}a - b,$$

where the last step uses that loss rates are at most 1. Proximal term: by Cauchy–Schwarz and the diameter, $\eta\,\Delta \boldsymbol{x}^\top \boldsymbol{d} \leq \eta D\|\Delta \boldsymbol{x}\|$. Combining the three terms gives $-\lambda \mathrm{Expl}(\boldsymbol{x}) + \frac{1}{2}a - b + \eta D\|\Delta \boldsymbol{x}\| \geq 0$, and rearranging yields the first inequality; the second inequality uses $b \geq 0$.

(ii) Apply the obtuse-angle principle to $\boldsymbol{w} = \Pi_\Delta(\boldsymbol{e}_m + c\tilde{\boldsymbol{\ell}})$ with $\boldsymbol{u} = \boldsymbol{e}_j$ (j any pool member):

$$\big(\boldsymbol{e}_m + c\tilde{\boldsymbol{\ell}} - \boldsymbol{w}\big)^\top (\boldsymbol{e}_j - \boldsymbol{w}) \leq 0 \iff c\,\tilde{\boldsymbol{\ell}}^\top (\boldsymbol{e}_j - \boldsymbol{w}) \leq (\boldsymbol{w} - \boldsymbol{e}_m)^\top (\boldsymbol{e}_j - \boldsymbol{w}) \leq \sqrt{2}\,\|\boldsymbol{w} - \boldsymbol{e}_m\|.$$

By (7), the left-hand side equals $c(\ell_j - \frac{1}{2} - b)$, so $\ell_j - \frac{1}{2} \leq \frac{\sqrt{2}}{c}\|\boldsymbol{w} - \boldsymbol{e}_m\| + b$; moreover, by Lemma 5.2 and Cauchy–Schwarz, $b \leq \eta \Delta \boldsymbol{x}^\top (\boldsymbol{x}_N - \boldsymbol{x}) \leq \eta D\|\Delta \boldsymbol{x}\|$. Taking j to be the largest component, the right-hand side is nonnegative, so taking the positive part does not affect the bound. ■

A.7 Theorem 5.7 and Mixed Strategies

Proof of (i). Let $\alpha_k = \|\boldsymbol{w}(\boldsymbol{x}_{k+1}) - \boldsymbol{e}_m\|$ and $\beta_k = \|\Delta \boldsymbol{x}_k\|$. By Theorem 5.5(i) and $a \leq \alpha_k$, $\lambda_k \mathrm{Expl}(\boldsymbol{x}_{k+1}) \leq \frac{1}{2}\alpha_k + \eta D \beta_k$. The minimum is at most the average; applying Cauchy–Schwarz to each of the two sums ($\sum_{k<K} \alpha_k \leq \sqrt{K}\sqrt{\sum \alpha_k^2}$) and substituting Corollary 5.4,

$$\begin{aligned}\min_{k<K} \lambda_k \mathrm{Expl}(\boldsymbol{x}_{k+1}) \;&\leq\; \frac{1}{K}\sum_{k<K}\Big(\tfrac{1}{2}\alpha_k + \eta D \beta_k\Big) \\ &\leq\; \frac{1}{\sqrt{K}}\Big(\tfrac{1}{2}\sqrt{\tfrac{c\eta}{2}} + \eta D\Big)\|\boldsymbol{x}_0 - \boldsymbol{x}_N\| \;\leq\; \frac{\sqrt{c\eta}/2 + 2\eta}{\sqrt{K}},\end{aligned}$$

where the last step uses $\|\boldsymbol{x}_0 - \boldsymbol{x}_N\| \leq \sqrt{2}$ and $D = \sqrt{2}$.

Monotone residual. With a fixed pool and a fixed λ, two consecutive steps share the same field G. Take the test point $\boldsymbol{x}_{k+2}$ in (P) at step k and $\boldsymbol{x}_{k+1}$ at step $k+1$:

$$\langle G(\boldsymbol{x}_{k+1}) + \eta \Delta \boldsymbol{x}_k,\; \boldsymbol{x}_{k+2} - \boldsymbol{x}_{k+1}\rangle \geq 0, \qquad \langle G(\boldsymbol{x}_{k+2}) + \eta \Delta \boldsymbol{x}_{k+1},\; \boldsymbol{x}_{k+1} - \boldsymbol{x}_{k+2}\rangle \geq 0.$$

Adding them, the G part is ≤ 0 by monotonicity, leaving $\eta\langle \Delta \boldsymbol{x}_k - \Delta \boldsymbol{x}_{k+1}, \Delta \boldsymbol{x}_{k+1}\rangle \geq 0$, i.e., $\|\Delta \boldsymbol{x}_{k+1}\|^2 \leq \Delta \boldsymbol{x}_k^\top \Delta \boldsymbol{x}_{k+1} \leq \|\Delta \boldsymbol{x}_k\|\|\Delta \boldsymbol{x}_{k+1}\|$, so $\|\Delta \boldsymbol{x}_{k+1}\| \leq \|\Delta \boldsymbol{x}_k\|$.

Proof of (ii). Under Rule A with the pool unchanged over steps $K-n, \ldots, K-1$, the field is the same at all these steps, so $\|\Delta \boldsymbol{x}_k\|$ is nonincreasing there by the monotone residual. Summing the energy inequality (8) from $\boldsymbol{x}_{K-n}$ gives $n\|\Delta \boldsymbol{x}_{K-1}\|^2 \leq \sum_{k=K-n}^{K-1} \|\Delta \boldsymbol{x}_k\|^2 \leq \|\boldsymbol{x}_{K-n} - \boldsymbol{x}_N\|^2 \leq 2$, i.e., $\beta := \|\Delta \boldsymbol{x}_{K-1}\| \leq \sqrt{2/n}$. At $\boldsymbol{x} = \boldsymbol{x}_K$, Lemma 5.2 and Cauchy–Schwarz give $b(\boldsymbol{x}) \leq \eta D \beta$, and Lemma 5.1 with $a^2 \leq \|\boldsymbol{w} - \boldsymbol{e}_m\|^2$ gives $a(\boldsymbol{x}) \leq \sqrt{c\,b(\boldsymbol{x})} \leq \sqrt{c\eta D\beta}$. Substituting into Theorem 5.5(i) with $\lambda \equiv 1$,

$$\mathrm{Expl}(\boldsymbol{x}_K) \;\leq\; \tfrac{1}{2}\sqrt{c\eta D\beta} + \eta D\beta \;\leq\; \sqrt{c\eta/2}\; n^{-1/4} + 2\eta\, n^{-1/2},$$

using $D = \sqrt{2}$. The residual itself decays at rate $n^{-1/2}$, but the pool weight only at rate $n^{-1/4}$, which dominates. ■

Besides latest $\boldsymbol{x}_{k+1}$ (a single network), each step yields two deployable mixtures, the current mixture $\boldsymbol{q}_{k+1} := \boldsymbol{v}_{k+1}/t_k$ and the time average

$$\bar{\boldsymbol{q}}_K := \frac{1}{T_K} \sum_{k<K} \boldsymbol{v}_{k+1}, \qquad T_K := \sum_{k<K} t_k.$$

In sequential games, a mixture becomes a behavior strategy via reach probabilities (Kuhn, 1953). Both mixtures also converge.

Proposition A.2 (Mixed strategies). *(i) The per-step mixture satisfies* $t_k \mathrm{Expl}(\boldsymbol{q}_{k+1}) \le \eta D \|\Delta \boldsymbol{x}_k\|$. *Under Rule B,* $t_k \to 1$, *so* $\mathrm{Expl}(\boldsymbol{q}_{k+1}) \to 0$, *and this conclusion does not require* $a \to 0$.
(ii) The time average satisfies

$$\mathrm{Expl}(\bar{\boldsymbol{q}}_K) \;\le\; \frac{\eta \max_{\boldsymbol{y}} \|\boldsymbol{x}_0 - \boldsymbol{y}\|^2}{2T_K} \;\le\; \frac{\eta}{T_K},$$

and $T_K \ge K$ *under both rules.*
(iii) With the errors of §5.3, these bounds become

$$t_k \mathrm{Expl}(\boldsymbol{q}_{k+1}) \;\le\; \eta D \|\Delta \boldsymbol{x}_k\| + \delta_k + a e_k, \qquad \mathrm{Expl}(\bar{\boldsymbol{q}}_K) \;\le\; \Big[\eta + \textstyle\sum_{k<K} (\delta_k + a_{k+1} e_k)\Big] \big/ T_K.$$

Unlike Theorem 5.5(i), (i) has no $\frac{1}{2}a$ term, because that term comes from the crude bound that the loss rate of the exploiter against a pool member is at most 1, which is not needed once the deployed object is the mixture formed from the coefficients of the algorithm. (ii) is the standard order for averaged iterates of the proximal point method (Rockafellar, 1976; Nemirovski, 2004), except that under the augmented structure the pool cross terms cancel exactly and pool admissions do not break the telescoping sum. (iii) is proved in Appendix A.8.

Proof of (i). $\boldsymbol{q}_{k+1} = \boldsymbol{v}_{k+1}/t_k \in X$, because $\boldsymbol{v}$ is a nonnegative combination of points in X with total mass t_k. For any $\boldsymbol{y} \in X$, take $\boldsymbol{x}' = \boldsymbol{y}$ in (P):

$$\boldsymbol{y}^\top S \boldsymbol{v} - \underbrace{\boldsymbol{x}^\top S \boldsymbol{v}}_{=b} + \eta\, \Delta \boldsymbol{x}^\top (\boldsymbol{y} - \boldsymbol{x}) \ge 0 \implies -\boldsymbol{y}^\top S \boldsymbol{v} \;\le\; -b + \eta\, \Delta \boldsymbol{x}^\top (\boldsymbol{y} - \boldsymbol{x}) \;\le\; \eta D \|\Delta \boldsymbol{x}\|.$$

By skew-symmetry, the left-hand side equals $\boldsymbol{v}^\top S \boldsymbol{y} = t\, \boldsymbol{q}^\top S \boldsymbol{y}$, and maximizing over $\boldsymbol{y}$ gives $t_k \mathrm{Expl}(\boldsymbol{q}_{k+1}) \le \eta D \|\Delta \boldsymbol{x}_k\|$. Under Rule B, $t_k = 1 + \lambda_k - \lambda_{k+1} \to 1$.

Proof of (ii). The intermediate step above, $-\boldsymbol{y}^\top S \boldsymbol{v}_{k+1} \le \eta \Delta \boldsymbol{x}_k^\top (\boldsymbol{y} - \boldsymbol{x}_{k+1})$, holds at every step. Using the polarization identity (with reference point $\boldsymbol{y}$),

$$2\, \Delta \boldsymbol{x}_k^\top (\boldsymbol{y} - \boldsymbol{x}_{k+1}) = \|\boldsymbol{x}_k - \boldsymbol{y}\|^2 - \|\boldsymbol{x}_{k+1} - \boldsymbol{y}\|^2 - \|\Delta \boldsymbol{x}_k\|^2 \;\le\; \|\boldsymbol{x}_k - \boldsymbol{y}\|^2 - \|\boldsymbol{x}_{k+1} - \boldsymbol{y}\|^2,$$

and telescoping over $k < K$,

$$T_K\, \bar{\boldsymbol{q}}_K^\top S \boldsymbol{y} = -\boldsymbol{y}^\top S \sum_{k<K} \boldsymbol{v}_{k+1} \;\le\; \tfrac{\eta}{2} \big(\|\boldsymbol{x}_0 - \boldsymbol{y}\|^2 - \|\boldsymbol{x}_K - \boldsymbol{y}\|^2\big) \;\le\; \tfrac{\eta}{2} \|\boldsymbol{x}_0 - \boldsymbol{y}\|^2 \;\le\; \eta.$$

Divide both sides by T_K and maximize over $\boldsymbol{y}$. Under Rule B, $t_k = \lambda_k + 1 - \lambda_{k+1}$, and the sum telescopes to $T_K = K + \lambda_0 - \lambda_K \ge K$ (since $\lambda_0 = 1$); under Rule A, $t_k \ge 1$. ■

A.8 Theorem 5.8 and the Two Convergence Results

Let $\hat{b} := \sum_i w_{k+1,i} (\hat{\ell}_{k,i} - \frac{1}{2})$ denote the pool pressure computed from the estimated loss rates, while a and b are defined with the actual weights and the true loss rates.

Energy inequality. Lemma 5.1 holds as usual for the vector that is actually projected (the proof is the same, with $\tilde{\boldsymbol{\ell}}$ replaced by $\hat{\boldsymbol{\ell}}$), giving $\|\boldsymbol{w}_{k+1} - \boldsymbol{e}_m\|^2 \le c\hat{b}$. Switching back to the true values,

$$\hat{b} - b = \sum_i w_{k+1,i} \big(\hat{\ell}_{k,i} - \ell_i(\hat{\boldsymbol{x}}_{k+1})\big) \le \sum_i w_{k+1,i}\, e_k = a\, e_k,$$

since only the pool coordinates carry error (the mirror coordinate is always $\frac{1}{2}$), which makes this tighter than the generic $2e$ bound. Lemma 5.2 with $-\delta_k$: taking $\boldsymbol{x}' = \boldsymbol{x}_N$ in the relaxed (P), we again have $\boldsymbol{x}_N^\top S\hat{\boldsymbol{v}} \le 0$, which gives $\eta\Delta\boldsymbol{x}^\top(\boldsymbol{x}_N - \hat{\boldsymbol{x}}) \ge b - \delta_k$. Chaining these,

$$\eta\,\Delta\boldsymbol{x}^\top(\boldsymbol{x}_N - \hat{\boldsymbol{x}}) \;\ge\; b - \delta_k \;\ge\; \hat{b} - ae_k - \delta_k \;\ge\; \tfrac{1}{c}\|\boldsymbol{w}_{k+1} - \boldsymbol{e}_m\|^2 - ae_k - \delta_k,$$

and substituting into the polarization identity gives the energy inequality.

Certificates. Repeat the proofs of Theorem 5.5 and Proposition A.2(i)(ii) with two substitutions; this also proves Proposition A.2(iii). The right-hand side of (P) changes from 0 to $-\delta_k$, which adds δ_k to every certificate. Wherever $-b \le 0$ was used, we now only have $\hat{b} \ge 0$, so we use $-b \le -\hat{b} + ae_k \le ae_k$, which adds ae_k to the latest certificate and to each of the two mixture certificates. In-pool certificate: the obtuse-angle principle applied to $\boldsymbol{e}_m + c\hat{\boldsymbol{\ell}}$ gives $\hat{\ell}_j - \frac{1}{2} \le \frac{\sqrt{2}}{c}\|\boldsymbol{w} - \boldsymbol{e}_m\| + \hat{b}$; switching back to the true values uses $\ell_j \le \hat{\ell}_j + e_k$ and $\hat{b} \le b + ae_k \le b + e_k$, for a total of $2e_k$, and finally $b \le \eta D\|\Delta\boldsymbol{x}\| + \delta_k$.

Summable errors ⇒ convergence. If $\sum_k(\delta_k + a_{k+1}e_k) < \infty$, summing the energy inequality gives $\sum\|\Delta\boldsymbol{x}_k\|^2 < \infty$ and $\sum\|\boldsymbol{w}_{k+1} - \boldsymbol{e}_m\|^2 < \infty$, so $a \to 0$ and $\Delta\boldsymbol{x} \to 0$, and substituting into the latest certificate gives $\mathrm{Expl}(\hat{\boldsymbol{x}}_{k+1}) \to 0$. For convergence of the whole sequence, define

$$E_k := \|\boldsymbol{x}_k - \boldsymbol{x}_N\|^2 + \tfrac{2}{\eta}\sum_{j\ge k}\big(\delta_j + a_{j+1}e_j\big);$$

the energy inequality gives $E_{k+1} \le E_k$, and since the tail sum tends to zero, E_k and $\|\boldsymbol{x}_k - \boldsymbol{x}_N\|^2$ have the same limit; then take the limit along a subsequence converging to a Nash strategy. Summable errors do not restore the monotonicity of the residuals.

Constant errors ⇒ neighborhood. If $\delta_k \le \delta$ and $e_k \le e$, summing the energy inequality and dividing by K gives

$$\min_{k<K}\Big[\|\Delta\boldsymbol{x}_k\|^2 + \tfrac{2}{c\eta}\|\boldsymbol{w}_{k+1} - \boldsymbol{e}_m\|^2\Big] \le \tfrac{2}{K} + \tfrac{2}{\eta}(\delta + e).$$

This is a statement about $\min_{k<K}$. Re-summing over an arbitrary suffix yields a $\liminf$-type statement, but it does not imply that the iterates stay in the neighborhood from some point onward. ■

A.9 The Role of the Oracle

Theorem 5.6 shows that exploiters can enter or leave the pool freely without breaking convergence. On top of this, the oracle does two things. It provides directed pressure (with a single member, the admission weight is $\mathrm{clip}(c\mu/2, 0, 1)$), and it provides measurable evidence, namely the admission gain $g := [\ell(\hat{\boldsymbol{x}}; \hat{\boldsymbol{y}}) - \max_{i\in P}\ell_i(\hat{\boldsymbol{x}})]^+$. If the oracle falls short of the worst-case opponent by only τ, i.e., $\max_{\boldsymbol{y}}\ell(\hat{\boldsymbol{x}}; \boldsymbol{y}) \le \ell(\hat{\boldsymbol{x}}; \hat{\boldsymbol{y}}) + \tau$, then a one-line decomposition gives

$$\mathrm{Expl}(\hat{\boldsymbol{x}}) \le \Big[\max_{i\in P}\ell_i(\hat{\boldsymbol{x}}) - \tfrac{1}{2}\Big]^+ + g + \tau;$$

τ is not measurable for a neural exploiter, and this is the source of the bridge proxy underestimating the true value in §6.3.

B Comparison with Mainstream Methods

Table 2: Comparison with mainstream methods.

	Deployed object	Equilibrium carrier	Modifies RL core	Nontransitive memory	Actively finds holes	Online certificate
PSRO / DO	Mixture	External meta-solver	No	Yes	Yes	Within meta-game
Hedge/MWU pool weights†	Mixture or time average	Accumulated weights (moving dual anchor)	No	Yes	Yes	Time-average regret
PFSP / League	Single network	None	No	Yes	Yes	None
NFSP	Average network	Average network	Yes	Implicit	No	None
R-NaD / MMD	Single network	Moving anchor	Yes	No	No	Primal residual
CoDA (moving dual anchor)	Single network	Moving anchor on sampling simplex	No	Yes	Yes	In-pool
CoDA-R	Single network or mixture	Primal moving anchor + fixed dual anchor $\boldsymbol{e}_m$	Wrapper	Yes	Yes	In-pool and full-game

†Including the Hedge/PRD meta-solvers of PSRO (Lanctot et al., 2017), the opponent sampler of OpenAI Five (OpenAI et al., 2019), AP-SRO (McAleer et al., 2022), and ODO (Dinh et al., 2022).

C Additional Experimental Results

Implementation and evaluation protocol. The PPO implementation (Schulman et al., 2017) (with GAE (Schulman et al., 2016) for advantage estimation) follows the procedure of §4, replacing each exact object with a runnable approximation whose error is absorbed into e_k or δ_k. One outer step contains 32 rollouts. After each rollout, the game outcomes are used to update the loss-rate estimates $\hat{\boldsymbol{\ell}}$ and to re-project, and the next rollout samples opponents accordingly and applies importance weighting. The opponent of the mirror arm is a frozen copy of the current parameters, the proximal term takes the form of R-NaD-style reward shaping, and the policy anchor and λ are updated at the end of each outer step. The oracle trains exploiters with PPO, and the upper-bound control uses full-tree exact best responses instead. In the implementation, a and $\|\boldsymbol{w} - \boldsymbol{e}_m\|$ can be read off directly, but the primal residual has only a KL proxy and δ_k has no end-to-end bound, as is also the case for neural R-NaD-PPO and PSRO. The budget of $B = 5 \times 10^5$ games counts the games the algorithm actually consumes, including learner games, exploiter training, admission evaluation, and the meta-matrix games of PSRO; CoDA-R spends about 14% of its budget on the oracle, and its learner plays correspondingly fewer games. Exploitability is measured in chips for Leduc; the payoffs of the other games are ± 1.

Short and long runs. The long-run comparison of §6.2, the dual-strength sweep on Leduc, and the vulnerability injection of §6.3 use the same setup; the ablations, the comparison with CoDA, and the neural certificate proxies come from earlier short runs (96 or 192 rollouts). The main difference between the two lies in the dual-side loss-rate estimator. Short runs use an EMA with a fixed step size ($\beta = 0.1$) and a pool capacity of 16; long runs instead use an EMA whose step size adapts to the number of games, reset the history when the estimate changes abruptly, reserve a small fraction of games to probe zero-weight members, and use a pool capacity of 8. In addition, the horizontal axis of short runs counts pool-evaluation games that are used only for logging.

Controlled vulnerability injection. Results are shown in Figure 6, Figure 7, and Table 6. For each seed, we first train R-NaD-PPO for 64 rollouts to obtain a common base, and train attacker A with PPO so that the loss rate of the base rises to 0.55–0.57; four branches then start from the same base with newly created optimizers and each train for 96 rollouts, with A injected at rollout 0 and B (an attacker trained against the policy of the reference branch at rollout 48) injected at rollout 48. The proportions of fixed-mix (fixmix in the table) are fixed before the experiment, namely $(w_m, w_A) = (0.65, 0.35)$ after A is injected and $(w_m, w_A, w_B) = (0.50, 0.25, 0.25)$ after B is injected. The value 0.35 is the closed-form initial weight computed from an expected lead of 0.12 and $c = 6$; the actual attackers lead by only 0.05–0.07, so fixed-mix gives A about twice the initial weight of CoDA-R (0.12–0.20). coda_r_exact overwrites the estimates with the exact ℓ_i at every rollout ($e_k = 0$). B finds no vulnerability on 3 of 4 seeds, and the T_{fix} of all branches is close to immediate repair. The exact-loss-rate version is not better than the EMA version, and the difference is within the CIs.

Ablations, comparison with CoDA, and certificate proxies. The ablations are shown in Figure 11 (96 rollouts), the comparison with the moving-anchor CoDA in Figure 8, and the neural certificate proxies in Figure 9. CoDA-R and R-NaD-PPO show similar trends as η varies. Of the two forms of the proximal term, loss KL is better on Leduc and reward shaping is significantly better on Goofspiel(4); the main experiments use reward shaping for both methods. Freezing the mirror for a whole period is worse on Leduc (1.36 vs. 1.14); the differences for freezing the pool coefficients and for switching the oracle on or off are all within the CIs. Among the negative results, the bridge proxy $[\max_P \ell_i - \frac{1}{2}]^+ + g$ falls below the true value in 55 of 72 oracle evaluations; selecting the

checkpoint at the minimum of `latest_cert` is worse on Leduc than taking the last one ($+0.145$ [0.101, 0.678], Figure 10).

C.1 Estimator and Oracle Diagnostics

Table 3 explains two phenomena in the main results. On Goofspiel(4), the oracle does find vulnerabilities (mean g of 0.13), but in 54 of the 61 cases where the true loss rate against a member jumped above 0.52, the weight of that member was below 0.01 at the time; the lag of the estimator is the source of the occasional spikes in `tail_worst`, and it is also the measured form of e_k in Theorem 5.8. On Leduc, g is always 0 for the PPO oracle, so the pool barely takes part in training; the true loss rates against exact-BR members do exceed 0.52 (96 times), but the closed-form weight is too small at $c = 6$, and after scaling up to $c = 48$ the holes drop to 5 and the pool mass rises to 0.29.

Table 3: Pool diagnostics of each CoDA-R cell in the long-run comparison. g is the gain against latest at oracle admission; holes is the number of member occurrences with true loss rate > 0.52 at evaluation ticks, with the number of those whose weight was < 0.01 at the time in parentheses; dual regret $= \max_j \ell_j - \sum_j w_j \ell_j$ ($\boldsymbol{w}$ normalized within the pool).

Environment	Method	#oracle	mean g	$g > 0.02$	pool_mass	holes (missed)	dual regret mean / p90
Goofspiel(4)	PPO oracle	24	0.127	21	0.040	61 (54)	0.044 / 0.146
Goofspiel(4)	exact-BR oracle	28	0.116	18	0.039	64 (50)	0.035 / 0.120
Leduc	PPO oracle, c=6	24	0.000	0	0.023	0	0.004 / 0.011
Leduc	exact-BR oracle, c=6	28	0.019	14	0.065	96 (30)	0.008 / 0.020
Leduc	exact-BR oracle, c=48	28	0.010	2	0.291	5 (2)	0.006 / 0.014
Liar's Dice	PPO oracle	28	0.018	7	0.066	n/a	n/a

C.2 Figures and Tables

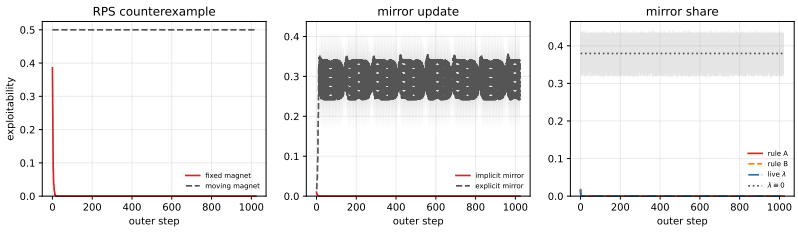


Figure 2: Finite-dimensional mechanism comparison (seed mean $\pm$ CI; vertical axis $\mathrm{Expl}(\boldsymbol{x}_k)$). Left: fixed versus moving anchor, starting from the RPS counterexample point. Middle: implicit versus explicit mirror. Right: share Rules A and B, real-time λ, and $\lambda \equiv 0$.

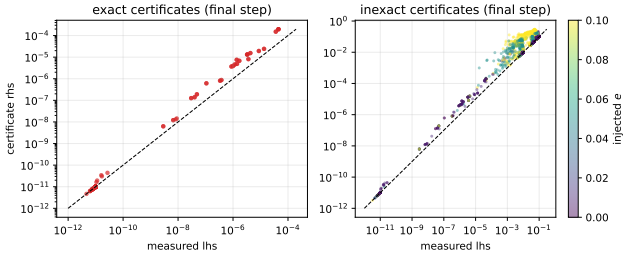


Figure 3: Certificates at the last step of the finite-dimensional experiments. The horizontal axis is the measured left-hand side, the vertical axis is the right-hand side of the certificate, and the dashed line is the diagonal $y = x$. Left: exact cells. Right: cells with errors, colored by the injected e.

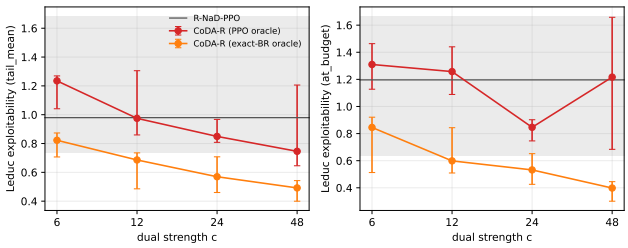


Figure 4: Dual-strength sweep on Leduc (IQM ± 95% CI). The gray line and shading are R-NaD-PPO. Left: `tail_mean`. Right: `at_budget`.

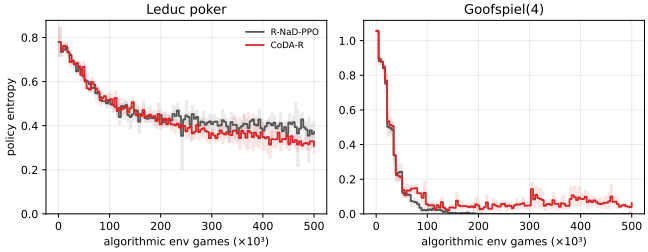


Figure 5: Policy entropy in the matched-budget long-run comparison (seed mean ± CI). On Goofspiel(4), the entropy of R-NaD-PPO falls below 10^{-4} after 2×10^5 games, in step with the degradation of its exploitability; CoDA-R stays at 0.05–0.1.

Table 4: Full metrics of the matched-budget long-run comparison ($B = 5 \times 10^5$, IQM [95% CI], $n = 4$). AUC is the integral over $[0, B]$ with previous-value hold, divided by B. Bold marks the best value among single-network methods in each game.

Method	Deploy	at_budget	tail_mean	AUC
Leduc				
R-NaD-PPO	L	1.196 [0.637, 1.666]	0.980 [0.736, 1.684]	0.937 [0.860, 1.081]
CoDA-R PPO oracle, c=48	L	1.217 [0.684, 1.658]	0.746 [0.646, 1.206]	0.930 [0.852, 0.986]
CoDA-R exact-BR oracle, c=48	L	**0.398** [0.301, 0.445]	**0.492** [0.400, 0.543]	**0.681** [0.637, 0.706]
PSRO, PPO BR	M	0.650 [0.637, 0.877]	0.656 [0.609, 0.783]	1.123 [1.114, 1.148]
PSRO, exact BR	M	0.353 [0.304, 0.419]	0.374 [0.369, 0.394]	0.616 [0.558, 0.634]
Goofspiel(4)				
R-NaD-PPO	L	0.869 [0.827, 0.990]	0.797 [0.561, 0.953]	0.653 [0.579, 0.707]
CoDA-R PPO oracle	L	**0.014** [0.000, 0.165]	**0.083** [0.037, 0.227]	**0.244** [0.211, 0.283]
CoDA-R exact-BR oracle	L	0.079 [0.000, 0.323]	0.115 [0.090, 0.186]	0.290 [0.239, 0.330]
PSRO, PPO BR	M	0.088 [0.054, 0.115]	0.072 [0.055, 0.126]	0.173 [0.156, 0.191]
PSRO, exact BR	M	$< 10^{-15}$	$< 10^{-15}$	0.018 [0.011, 0.023]
Liar's Dice				
R-NaD-PPO	L	0.272 [0.216, 0.291]	0.259 [0.207, 0.274]	**0.279** [0.268, 0.282]
CoDA-R PPO oracle	L	**0.251** [0.179, 0.257]	0.265 [0.231, 0.280]	0.317 [0.295, 0.345]
CoDA-R no oracle	L	0.253 [0.230, 0.353]	**0.256** [0.227, 0.348]	0.310 [0.304, 0.337]

Table 5: Dual-strength sweep on Leduc ($B = 5 \times 10^5$, IQM [95% CI], $n = 4$; $c = 6$ is taken from the main-experiment runs). pool_mass is the mean of a over the whole run, and est err is the mean of $|\hat{\ell}_i - \ell_i|$ at evaluation ticks. Bold marks the best `at_budget` and `tail_mean` under each oracle.

oracle	c	at_budget	tail_mean	pool_mass	est err
PPO	6	1.310 [1.127, 1.464]	1.235 [1.041, 1.269]	0.023	0.010
PPO	12	1.257 [1.089, 1.440]	0.975 [0.860, 1.305]	0.039	0.010
PPO	24	**0.846** [0.746, 0.902]	0.849 [0.808, 0.967]	0.070	0.010
PPO	48	1.217 [0.684, 1.658]	**0.746** [0.646, 1.206]	0.123	0.010
exact BR	6	0.845 [0.513, 0.921]	0.823 [0.707, 0.873]	0.065	0.011
exact BR	12	0.599 [0.509, 0.843]	0.686 [0.486, 0.735]	0.107	0.010
exact BR	24	0.532 [0.426, 0.652]	0.570 [0.460, 0.708]	0.175	0.009
exact BR	48	**0.398** [0.301, 0.445]	**0.492** [0.400, 0.543]	0.291	0.010

Table 6: Controlled vulnerability injection (Goofspiel(4), $n = 4$, IQM [95% CI]). T_{fix} is the rollout index at which $\ell \leq 0.52$ first holds for three consecutive rollouts; recur_A is the maximum of $\ell(\boldsymbol{x}; A) - \frac{1}{2}$ over $r \geq 56$; gap is the maximum true gain of A while $w_A < 0.01$; est err is the mean of $|\hat{\ell}_A - \ell(\boldsymbol{x}; A)|$. Lower is better for the first four columns, and bold marks the best value in each column; gap and est err are diagnostics of CoDA-R and are not compared.

Branch	$T_{\text{fix,A}}$	recur_A	expl_tail	expl_worst_tail	gap	est err
rnad	3.5	0.0073	0.232 [0.186, 0.326]	0.387 [0.221, 0.569]	—	—
fixmix	**2.0**	0.0138	0.450 [0.201, 0.574]	0.521 [0.289, 0.614]	—	—
coda_r_ema	6.0	0.0080	**0.167** [0.072, 0.272]	**0.272** [0.099, 0.514]	0.0115	0.0062
coda_r_exact	3.5	**0.0056**	0.267 [0.185, 0.339]	0.404 [0.393, 0.450]	0.0048	0

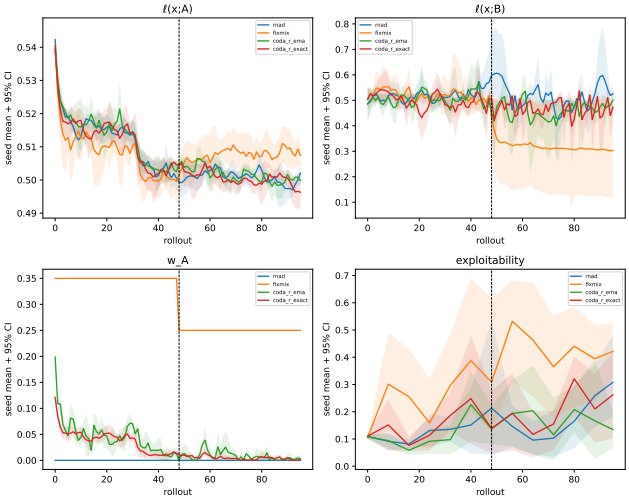


Figure 6: Controlled vulnerability injection (Goofspiel(4), seed mean ± 95% CI; the dashed line marks rollout 48, where B is injected). Top left: $\ell(\boldsymbol{x}; A)$. Top right: $\ell(\boldsymbol{x}; B)$. Bottom left: w_A (fixed proportion for fixmix). Bottom right: full-tree exploitability.

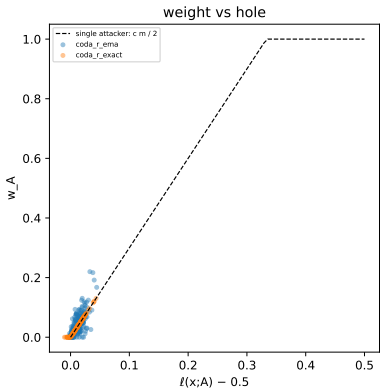


Figure 7: Scatter of w_A against $\ell(\boldsymbol{x}; A) - \frac{1}{2}$ in the controlled vulnerability injection (only rollouts before B is injected). The dashed line is the single-member closed form $\text{clip}(c\mu/2, 0, 1)$ with $c = 6$.

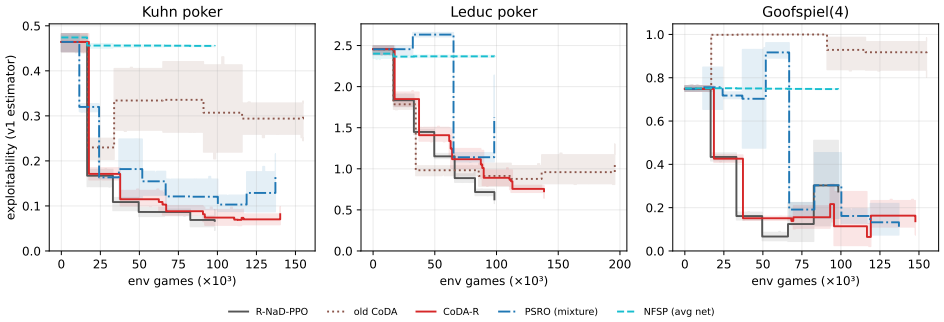


Figure 8: Short runs (192 rollouts; the horizontal axis counts all environment games, including diagnostic games; 4 seeds), comparing CoDA-R, R-NaD-PPO, CoDA with a moving dual anchor, the PSRO mixture, and the NFSP average network. Kuhn, CoDA, and NFSP are reported only here.

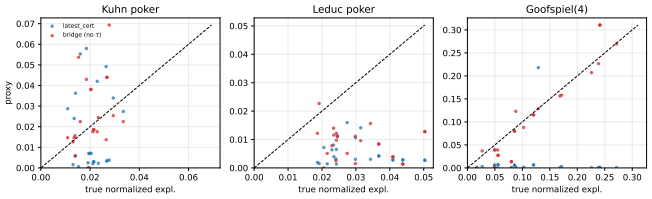


Figure 9: Scatter of two neural proxies against the true normalized exploitability of the same policy at oracle admission times (short runs). `latest_cert` falls below the true value 63/72 times, and the bridge proxy 55/72 times.

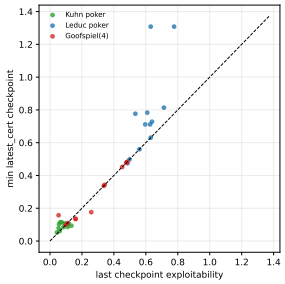


Figure 10: Checkpoint selection. The horizontal axis is the exploitability of the last checkpoint, and the vertical axis is the exploitability at the minimum of `latest_cert`.

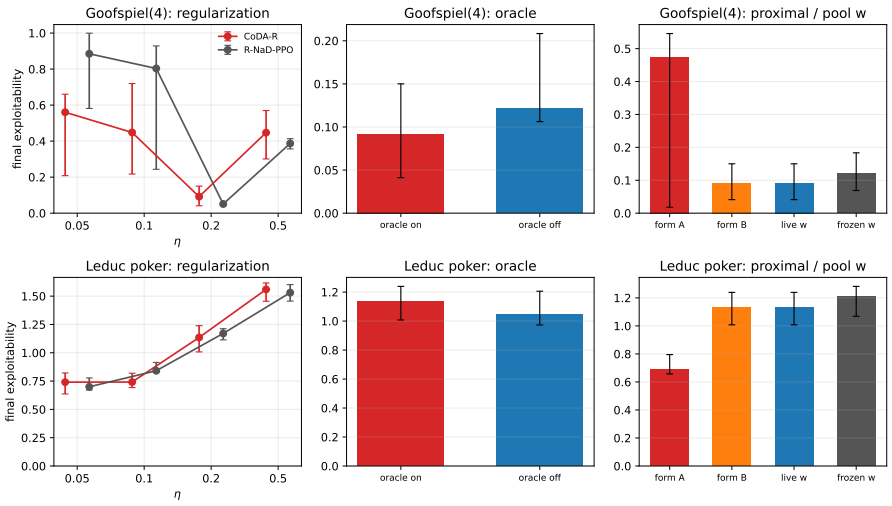


Figure 11: Ablations (96 rollouts; final exploitability, IQM ± CI). Left: η grid, shared by CoDA-R and R-NaD-PPO. Middle: oracle on/off. Right: proximal-term form A (loss KL) / B (reward shaping), and pool coefficients re-projected every rollout / frozen for a whole period.