WIN RATE IS NOT SECURITY: REDUCING LLM EXPLOITABILITY

Anonymous authors
Paper under double-blind review

ABSTRACT

As large language models (LLMs) evolve, an increasing body of research evaluates them on strategic games such as poker to study their behavior in environments with hidden information. We argue that there is a major gap in current evaluations that center on win rate, where LLMs with high win rates can fail against a well-constructed adversary. In a controlled Kuhn poker audit, half of the eight audited LLM systems earn above-equilibrium payoffs against a fixed opponent pool but fall below equilibrium against opponents that maximize their loss. Reducing this shortfall from equilibrium, known as exploitability, remains scarce in research on LLM game-playing agents. We propose an exploitability bound in terms of the training opponent, the requested policy updates, and the mismatch between the requested and learned policies. We further show how distillation error bounds the increase in exploitability relative to the target policy. We propose a new training method that uses exploitability to constrain and guide training. Across 18 poker runs, 94.4% have exploitability below the absolute game value at the selected checkpoint, at the final checkpoint, and throughout the final 20 rounds. A capability-preserving extension also maintains performance close to the unmodified model on general capability benchmarks.

1 INTRODUCTION

Strategic games provide controlled settings for training and evaluating the reasoning of large language models (LLMs). SPIRAL and MARSHAL use multi-agent self-play to develop reasoning capabilities (Liu et al., 2025; Yuan et al., 2025), while PokerBench evaluates poker decisions against expert actions (Zhuang et al., 2025). Such evaluations measure progress against sampled opponents or reference actions. Yet success against these opponents does not establish whether a policy can protect itself when an opponent adapts to its weaknesses. In a two-player zero-sum game, this question is captured by exploitability, the shortfall from the equilibrium payoff against the opponent that minimizes its payoff. A high win rate against an opponent pool does not establish a bound on this shortfall.

In this paper, we study how to carry a worst-case guarantee from a low-exploitability target into a trained LLM policy. Neural training changes action probabilities through shared parameters, so the model may not express the policy the game solver requests. For example, the target policy may vary its actions with the private cards, while the trained model chooses similar actions regardless of its cards. We study this *neural realization* problem, bound the additional exploitability caused by the mismatch, and check the learned policy after updates. We focus on finite games with exact best responses, which makes each step directly testable.

This problem builds on established approaches to equilibrium computation and policy learning. Counterfactual regret minimization (CFR) computes approximate equilibria (Zinkevich et al., 2007); Deep CFR uses neural function approximation (Brown et al., 2018), and policy-space response oracles construct mixtures of learned policies (Lanctot et al., 2017). Policy distillation trains a model to match another policy's action probabilities (Rusu et al., 2015). Our focus is whether the target's security survives that transfer into a pretrained LLM; related work is discussed in Appendix A. We address this question through three contributions: exact evaluation, a bound on the exploitability added by neural realization error, and a training procedure that audits the resulting policy to ensure meaningful learning.

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.

First, **we evaluate complete LLM policies against their exact best responses** (Section 2) to show the presence of the problem. Half of the eight sampled policy pairs earn above-equilibrium mean payoffs against a common opponent pool in canonical Kuhn, yet fall below equilibrium against their exact best responses. The experiment evaluates sequential policies with private information and separates the accuracy of policy extraction from exact evaluation of the resulting action distributions.

Second, **we bound the increase in exploitability caused by neural realization error** (Section 3). Our analysis separates adversary fidelity, update geometry, and neural realization. They account for weaknesses missed by the training opponent, the cumulative payoff shortfall of the requested updates relative to the best fixed policy, and the mismatch between requested and expressed policies. The bound states when classical regret guarantees survive neural training. For a fixed target, we further bound the increase in exploitability by the square root of a weighted divergence between target and model action probabilities. This connects the distillation objective to a worst-case guarantee. We also compute the minimum exploitability achievable by policies that ignore private information, quantifying the loss imposed by that restriction (Appendix D).

Third, **we train the model to match a fixed target and verify its exploitability** (Section 4; Figure 4). Training against exact best responses produces a collection of model policies. We choose a mixture of these policies that minimizes exploitability, allowing policies with different weaknesses to complement one another. This mixture becomes the target when its exploitability is below the absolute game value, the magnitude of the equilibrium payoff. Distillation starts from the saved checkpoint whose action probabilities most closely match the target. Since the bound does not guarantee optimization success, each proposed update is checked against an exact best response. Once an exploitability ceiling is activated, only updates that remain within it are accepted. We then reload the saved model and check that it reproduces the evaluated policy.

The selected policy, final policy, and final 20 training rounds remain below the absolute game value in 94.4% of the 18 Qwen3-4B runs across both seats in Kuhn, Leduc, and a fixed-board no-limit river endgame. In a constructed finite negotiation game with private participant types, the mixture target yields stable sub-value policies in five of six runs. A separate extension that constrains changes to general capabilities produces one model with sub-value exploitability on Kuhn with scores close to the unmodified model on the complete MMLU, HellaSwag, and GSM8K benchmarks.

2 Problem statement

We evaluate eight existing LLM systems on canonical Kuhn poker to test whether success against fixed opponents implies resistance to exploitation.

2.1 Evaluation protocol

Kuhn poker is a two-player zero-sum game with private cards and a game tree small enough to compute exact best responses. In sequence form, a realization plan records the probabilities of a player's action sequences.

Definition 1 (Exploitability)**.** *For a two-player zero-sum extensive-form game with payoff matrix A, let v^* denote the equilibrium payoff for seat 0: the highest expected payoff it can guarantee against any opponent. The exploitability of a seat-0 realization plan x is*

$$\mathrm{Expl}(x) = v^* - \min_{y \in \mathcal{Y}} x^\top A y,$$

where $\mathcal{Y}$ contains all feasible opponent realization plans; seat 1 is defined symmetrically. We report $\mathrm{Expl}(x)/|v^*|$, writing a ratio k as $k \times |v|$ (von Stengel, 1996; Koller et al., 1996). A policy is sub-value *when this ratio is below one.*

The audit covers released SPIRAL and MARSHAL checkpoints (Liu et al., 2025; Yuan et al., 2025), our reproduction of SEPO's supervised fine-tuning (SFT) pipeline (Arumugam et al., 2026), and prompted GPT-5.5, Claude Opus 4.8, and Qwen3-8B models. We estimate action probabilities at all information sets from valid responses, and compute their best-response values. Appendix B.1 gives the extraction settings, parse rates, and unparsed-output sensitivity.

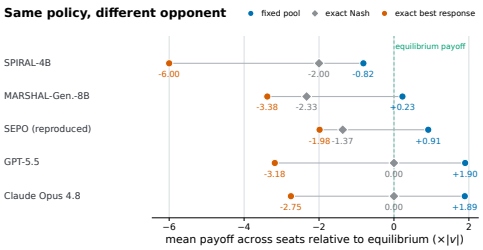

Figure 1: **Changing the opponent reverses apparent success.**

We construct a shared opponent pool for uniform comparison: each policy is played against a common fixed pool, an exact Nash strategy, and its own exact best response. The opponent pool uniformly mixes all eight opposite-seat policies.

2.2 Results

Half of the eight systems—MARSHAL-Generalist-8B, the SEPO reproduction, GPT-5.5, and Claude Opus 4.8—earn above-equilibrium mean payoffs against the common pool but fall below equilibrium against their exact best responses (Figure 1). All eight audited systems also have exploitability above $1 \times |v|$ at both seats (Table 1). SPIRAL was reported to be at or near equilibrium in its training Kuhn variant, but its policies exhibit exploitabilities of $5.000 \times |v|$ and $7.000 \times |v|$ under canonical Kuhn payoffs. Thus, success against the fixed pool does not avoid substantial worst-case loss. Appendix B.9 reports the complete opponent comparison.

3 Theory

Classical regret analysis assumes that the policy an update produces is the policy that gets played; LLM post-training breaks that assumption, because the trainer may request one policy while the model executes another. This section formalizes the gap and proves that our method bounds the added worst-case loss during fixed-target distillation. Full proofs are in Appendices C and G.

3.1 Requested, Expressed, and Deployed Policies

Let $\mathcal{X}$ and $\mathcal{Y}$ be the sets of legal realization plans for the two players in a finite perfect-recall game. Let A be the sequence-form payoff matrix of a two-player zero-sum extensive-form game with value v^*. For a sequence of T training rounds, at round $t \in \{1, \ldots, T\}$ the trainer requests the *requested* realization plan $m_t \in \mathcal{X}$, the model *expresses* the plan $x_t \in \mathcal{X}$ extracted under the declared interface, and $y_t \in \mathcal{Y}$ is the training adversary. When $m_t \neq x_t$, a guarantee for the requested policy does not automatically transfer to the expressed policy.

Table 1: **All audited policy tables remain exploitable on canonical Kuhn.** Values are exact exploitabilities of the sampled, parsed-output policies, reported in multiples of $|v^*|$. Details in Appendix B.1.

System	Policy source	Seat 0	Seat 1
SPIRAL-4B	Released checkpoint	5.000	7.000
MARSHAL-Kuhn-4B	Released checkpoint	4.395	5.470
MARSHAL-Generalist-4B	Released checkpoint	4.005	4.540
MARSHAL-Generalist-8B	Released checkpoint	3.281	3.480
SEPO	Our SFT reproduction	1.459	2.510
GPT-5.5	Prompted model	3.500	2.861
Claude Opus 4.8	Prompted model	2.700	2.800
Qwen3-8B	Prompted model	8.000	7.000

We use the exploitability definition in Section 2.

The denominator $|v^*|$ is the equilibrium edge granted by the game's information structure; for the favored player, a value below one means no opponent can erase it.

3.2 Formalizing the gap

Our first result proposes to separate the solver-to-neural gap into three distinct terms.

Theorem 1 (Exploitability bound). *For $T \geq 1$, suppose y_t is an approximate best response to x_t with gap γ_t, i.e., $x_t^\top A y_t \leq \min_{y \in \mathcal{Y}} x_t^\top A y + \gamma_t$. Define*

$$R_T^m = \max_{x \in \mathcal{X}} \sum_{t=1}^{T} x^\top A y_t - \sum_{t=1}^{T} m_t^\top A y_t.$$

If $\|x_t - m_t\|_1 \leq \varepsilon_t$ and $u_A = \max_{y \in \mathcal{Y}} \|Ay\|_\infty$, define $\bar{\gamma}_T = T^{-1} \sum_t \gamma_t$ and $\bar{\varepsilon}_T = T^{-1} \sum_t \varepsilon_t$. Then

$$\frac{1}{T} \sum_{t=1}^{T} \mathrm{Expl}(x_t) \;\leq\; \underbrace{\frac{R_T^m}{T}}_{\text{update geometry}} + \underbrace{\bar{\gamma}_T}_{\text{adversary fidelity}} + \underbrace{\frac{1}{T} \sum_t (m_t - x_t)^\top A y_t}_{\text{neural realization}} \;\leq\; \frac{R_T^m}{T} + \bar{\gamma}_T + u_A \bar{\varepsilon}_T.$$

Consequently, the best played iterate is no worse than the same bound.

The three terms isolate distinct failure modes. *Adversary fidelity* $\bar{\gamma}_T$ grows when the training opponent leaves weaknesses unexplored; *update geometry* R_T^m/T is the regret accumulated along the requested path; *neural realization* is the strategic loss incurred when training turns the requested plan m_t into the expressed plan x_t. For exact best responses, the fidelity term is zero; the remaining terms bound the average exploitability of the expressed policies. The bound does not require the final policy to be the least exploitable iterate.

Corollary 1 (No-regret convergence for LLMs). *Suppose the responses are exact, so $\bar{\gamma}_T = 0$, and suppose the planned policy gives $R_T^m \leq C\sqrt{T}$ for a constant C independent of T. If the average realization error also vanishes, $\bar{\varepsilon}_T \to 0$, then*

$$\min_{1 \leq t \leq T} \mathrm{Expl}(x_t) \;\leq\; \frac{C}{\sqrt{T}} + u_A \bar{\varepsilon}_T \;\to\; 0.$$

Classical regret matching$^+$ supplies a no-regret bound for its own update sequence. Applying that rate here requires the stated regret condition for the requested policies, together with vanishing average realization error.

3.3 Bounding the loss from distillation

For a fixed target, distillation error bounds the additional exploitability of the learned policy. The target can be an archive mixture or an exact equilibrium; its own exploitability contributes to the resulting bound.

Let Z be the finite set of pure opponent plans, let $\ell_z(x)$ be the loss against z, and write $E(x) = \max_{z \in Z} \ell_z(x)$ for exploitability. The set $S_\tau = \{x : E(x) \leq \tau\}$ of policies that are at most τ-exploitable is convex. For a move from a target q to a neural policy p, define the one-sided added loss

$$d(q \to p) = \max\left\{0, \max_{z \in Z} [\ell_z(p) - \ell_z(q)]\right\}.$$

For the target and student action distributions q_I, p_I at information set I, let $r_q(I)$ denote the target's own reach probability, and fix weights $w_I \geq r_q(I)$. The excess reach-weighted cross-entropy is

$$\Phi_q(p) = \sum_I w_I D_{\mathrm{KL}}(q_I \| p_I) = \sum_I w_I [\mathrm{CE}(q_I, p_I) - H(q_I)].$$

Theorem 2 (Distillation bounds exploitability)**.** *Let $w_I \geq r_q(I)$ for every information set I, and define*

$$\Phi_q(p) = \sum_I w_I D_{\mathrm{KL}}(q_I \| p_I).$$

Let $x_1, \ldots, x_T$ be observed played policies whose training trajectory has exploitability bound B_T, and let q minimize E over their convex hull. In a finite perfect-recall game, suppose student action probabilities are positive and terminal utility has width Δ. Then

$$E(q) \leq B_T, \qquad d(q \to p) \leq \Delta\sqrt{\Phi_q(p)}, \qquad E(p) \leq B_T + \Delta\sqrt{\Phi_q(p)}.$$

where

$$B_T = \frac{R_T^m}{T} + \bar{\gamma}_T + u_A \bar{\varepsilon}_T.$$

For an exact-equilibrium target, $B_T = 0$.

Hence, excess reach-weighted distillation loss bounds the added worst-case loss against every opponent. If the target is an exact equilibrium, the student's exploitability is controlled entirely by the distillation error.

Note that we have introduced weights inside Φ_q with $w_I \geq r_q(I)$. We now show that this is necessary: write $c_z(I) = \Pr_{q,z}[I]$ for the probability that the target q reaches I while the opponent plays z.

Proposition 1 (Valid reach weights)**.**

- *$\sum_I \widetilde{w}_I \kappa_I$ dominates the reached discrepancy $\sum_I c_z(I)\kappa_I$ for every opponent z and every $\kappa_I \geq 0$ if and only if $\widetilde{w}_I \geq c_z(I)$ for every z, I.*
- *If we only have $c_z(I) \leq r_q(I)$ (unknown opponent), then $r_q(I)$ is the pointwise least weight that is safe for all potential players.*
- *No finite universal constant can replace the square-root dependence of Theorem 2 by a bounded-payoff guarantee that is linear in $\Phi_q(p)$.*

So the target's own reach used to compute distillation loss is the weakest assumption the trainer can make while still covering every opponent.

3.4 Checking exploitability during training

Proposition 1 justifies the weighting used in the distillation loss. It does not ensure that optimization makes $\Phi_q(p)$ sufficiently small. We therefore evaluate proposed updates directly: after each update, we extract the complete model policy under the declared interface and score it against a freshly computed exact best response. Once an exploitability ceiling is activated, we reject updates exceeding it and restore the preceding model state. We call the exact evaluation of each candidate an *audit*; Section 4 gives the activation threshold and ceiling used in training.

Under the acceptance condition below, every accepted policy after entry remains within the specified exploitability threshold.

Proposition 2 (Preservation of an exploitability threshold)**.** *The set $S_\tau = \{x : E(x) \leq \tau\}$ is closed and convex. The one-sided loss d is nonnegative, positively homogeneous, and subadditive, and satisfies $E(x) \leq E(m) + d(m \to x)$ for all feasible m, x. Suppose every proposed update is exactly audited, every accepted update after first entering S_τ remains in S_τ, and rejection restores the last accepted state. Then a procedure that returns only such an accepted state returns a policy in S_τ.*

4 Methodology

Our method has three stages (Figure 4 in Appendix E): search for policies the model can express, construct a fixed target, and distill that target while checking exploitability. The target provides a consistent training objective, while exact best-response checks determine whether the model's learned policy satisfies the required exploitability threshold.

Searching for candidate policies. We use regret matching$^+$ (RM$^+$) to generate training targets against the current model policy's exact best response (Zinkevich et al., 2007; Tammelin, 2014). At information set I, let $\pi_t(a \mid I)$ be the model's action distribution at round t, and let $Q_t(I, a)$ be the counterfactual value of action a against that response. With $V_t(I) = \sum_a \pi_t(a \mid I) Q_t(I, a)$, we update the accumulated nonnegative regrets as

$$R_t^+(I, a) = \max\{R_{t-1}^+(I, a) + Q_t(I, a) - V_t(I), 0\}.$$

Normalizing these regrets gives the next requested action distribution. If all regrets are zero, we use a uniform distribution. A small uniform component keeps every legal action's probability positive. We then train the model to match this requested policy, extract its actual action probabilities, and compute its exploitability. Repeating this process yields an observed policy set containing the policies the model actually learned. We use this set to construct the target.

Constructing a fixed target. Different observed policies can have different weaknesses. We choose their least-exploitable mixture. Let $p_1, \ldots, p_n$ be the observed realization plans, which encode probabilities of the player's action sequences, and write $P = [p_1, \ldots, p_n]$. Let $w = (w_1, \ldots, w_n)$ contain the mixture weights, where w_i is the weight assigned to policy p_i. The weights $w \in \Delta_n$ form a simplex, and the mixed realization plan is $Pw = \sum_{i=1}^n w_i p_i$. We choose the optimal weights w^* by solving

$$q = Pw^*, \qquad w^* \in \arg\min_{w \in \Delta_n} E(Pw).$$

This is a linear program because exploitability is a maximum of affine losses. The resulting mixture is converted into action probabilities at each information set. We use it as the target if its exploitability is below the absolute game value. Otherwise, we use an exact equilibrium. In either case, we hold the chosen target q fixed during distillation.

Initializing the model and distilling the target. We start from the saved checkpoint whose action probabilities most closely match the target. Closeness is measured by total variation distance averaged over information sets, with deterministic tie-breaking. This selects the initial model from the existing observed set before distillation begins.

Training minimizes cross-entropy between the fixed target q and the model policy. At each information set I, the loss has weight $w_I = \max\{r_q(I), 10^{-6}\}$, where $r_q(I)$ is the product of the target's own action probabilities along the sequence leading to I. The weights satisfy $w_I \geq r_q(I)$, as required by the distillation bound in Theorem 2; the small positive floor also gives a training signal to information sets the target never reaches. The bound relates excess loss above the target's entropy to the additional exploitability of the model policy. Proposition 1 characterizes which weights provide this guarantee. The bound does not ensure that optimization attains a sufficiently small loss, so we also check exploitability directly.

Checking updates and evaluating the saved policy. After each proposed update, we extract the complete model policy and compute its exact exploitability. We also measure its distance from the target and its change from the preceding policy, using mean total variation distance. Initially, updates are constrained by progress toward the target and a limit on policy change; once exploitability reaches $0.95 \times |v|$, a ceiling of $0.99 \times |v|$ activates and later updates that exceed it are rejected, restoring the preceding model state. The reported sub-value threshold remains $1 \times |v|$.

To measure the saved policy, we reload each candidate checkpoint with the base model and repeat policy extraction and exact best-response evaluation under the same numerical configuration. The resulting exploitability describes that checkpoint under the declared game and model interface. We retain the extracted policy, target, loss weights, and checkpoint hashes so the reported values can be reproduced. Section 5 reports selected and final exploitability and whether all measurements in the final training window remain sub-value.

Preserving general capabilities. A separate extension also preserves general capabilities: a quadratic program projects each update onto half-spaces defined by five protected quantities (chat divergence, reference loss, instruction following, knowledge, and reasoning), and the largest step satisfying them and the exploitability threshold is accepted (Appendix F).

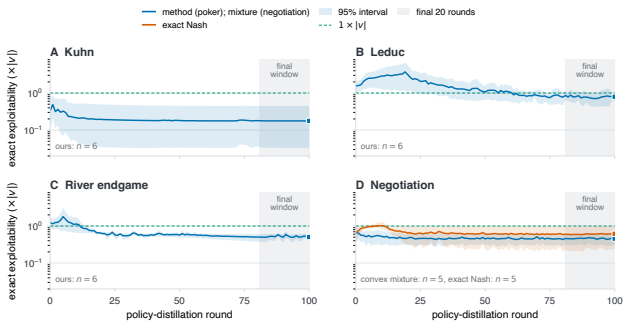

Figure 2: **The complete procedure produces stable sub-value policies.**

Table 2: **The learned policies achieve stable sub-value exploitability.**

game	method	init.	selected	final	P90	stable/all
Kuhn	ours	0.337	0.177	0.177	0.177	6/6
Leduc	ours	1.577	0.696	0.793	0.907	5/6
River endgame	ours	1.228	0.493	0.505	0.565	6/6
Negotiation	convex mixture	0.667	0.423	0.452	0.492	5/6
Negotiation	exact Nash	0.667	0.611	0.615	0.624	4/6

5 EXPERIMENTS

We evaluate Qwen3-4B in both seats of Kuhn, Leduc, and one fixed-board no-limit river endgame, with three seeds. Matched three-seed controls test opponent feedback and update strategy. The capability extension evaluates both exploitability and general tasks on the same adapter.

We also evaluate a finite negotiation game in which two players divide a $10 budget through an opening offer and at most one counteroffer. Each player privately observes a low, medium, or high strength. Walking away ends the negotiation and determines the allocation from both players' strengths, with a $1 transfer from the departing player to the other player. This setting tests agreement and withdrawal decisions under private information; Appendix E.5 gives the full rules.

The main poker cohort and matched ablations use 32-bit floating-point arithmetic (fp32) throughout. Candidates are reloaded and evaluated at rounds 25, 50, 75, and 100. *Selected exploitability* is the lowest value on this grid, *final exploitability* the value at round 100, and P90 the 90th percentile over the final 20 rounds. A *stable sub-value outcome* requires selected and final exploitability and every final-window measurement to be below the absolute game value; runs without a final policy remain in outcome denominators (Appendices B, E, and E.2).

5.1 COMPLETE-METHOD PERFORMANCE

Across the 18 poker runs, 94.4% achieve stable sub-value outcomes (Table 2). Kuhn and river meet this condition in all six runs each, and Leduc in five of six. Relative to their initializers, mean exploitability decreases by 52.2% at selection, 49.9% at the final policy, and 46.7% at P90. These runs evaluate fixed-target distillation from saved search checkpoints. Figure 2 shows the complete distillation trajectories.

In negotiation, archive and equilibrium targets yield stable sub-value outcomes in five of six and four of six runs, respectively.

The capability-constrained Kuhn extension ends at $0.2585 \times |v|$ after 20 accepted updates. On 25,403 examples, that adapter scores $0.6836/0.6861/0.8506$ on Massive Multitask Language Understanding (MMLU), HellaSwag, and Grade School Math 8K (GSM8K), compared with $0.683/0.685/0.850$ for the unmodified model (Hendrycks et al., 2021; Zellers et al., 2019; Cobbe et al., 2021).

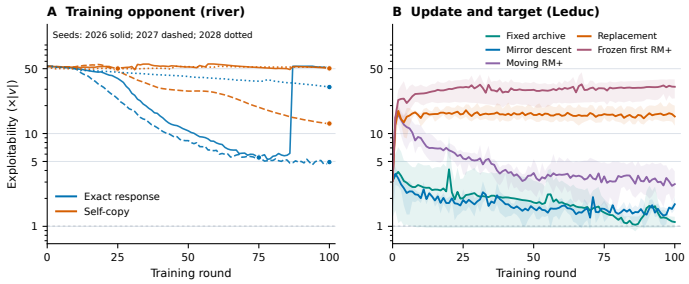

Figure 3: **Better intermediate policies need not yield stable final behavior.**

Table 3: **The archive target gives the lowest final exploitability in all three seeds.** P90 summarizes the final 20 rounds; stable requires every measurement in that window, selected and final exploitability to be sub-value.

Strategy	Selected	Final	P90	Stable
Fixed archive target	1.113	1.113	1.562	1/3
Mirror descent	1.409	1.729	1.713	0/3
Moving RM$^+$	2.745	2.852	3.660	0/3
Direct replacement	14.626	15.271	16.396	0/3
Frozen first RM$^+$ target	28.752	31.911	32.470	0/3

5.2 Opponent feedback and update strategy

Training opponent. Six river seat-0 runs compare exact best-response feedback with *self-copy feedback*, the model's opposite-seat policy re-extracted each round. Both arms start from the unmodified model and use proximal mirror descent: action probabilities are updated multiplicatively using counterfactual action values. Each run receives 100 rounds with 25 inner training passes per round, without acceptance checks. Both arms are evaluated against exact best responses.

Exact feedback gives lower selected exploitability in all three seeds: means are $14.046 \times |v|$ for exact feedback and $37.795 \times |v|$ for self-copy. Final means are $29.288 \times |v|$ and $38.384 \times |v|$, with exact feedback better in two seeds. In the remaining seed, its selected checkpoint is $5.518 \times |v|$ but its final policy reaches $51.244 \times |v|$ (Figure 3A). These results support exact best-response feedback as a means of finding less exploitable policies during search. The subsequent stages of our method construct a fixed target and constrain further updates to preserve low exploitability.

Update and fixed-target comparisons. Fifteen Leduc seat-0 runs compare five strategies across three seeds. Within each seed, all arms start from the same terminal search checkpoint, use exact opponent feedback, and receive 100 rounds with five inner passes per round at learning rate 10^{-4}. They use the same current-policy reach weights and disable acceptance checks. *Direct replacement* sets the target to a softmax of counterfactual values. The other arms use mirror descent, moving RM$^+$ targets, the *frozen first RM$^+$ target* held fixed after the first requested update, or a fixed archive-mixture target.

The fixed archive target gives the lowest final exploitability in every seed (Table 3), but only one run is stable sub-value. Its P90 is also worse than mirror descent in one seed. Freezing the first RM$^+$ target performs poorly; that target itself has exploitability $28.787 \times |v|$ in seed 2026. Fixing a target therefore does not ensure low exploitability: the target's quality and the model's fit both matter.

5.3 Target construction and initialization

Combining archived policies. We compare two targets constructed from each of 13 policy archives: the optimized mixture and the least-exploitable individual policy. Within each comparison, initialization and distillation settings are identical. In 11 archives, both targets are sub-value and both yield stable sub-value learned policies. The mixture gives lower selected, final, and P90 exploitability

Table 4: **The combined rule has a modest advantage in stable sub-value outcomes.** Stable requires sub-value selected and final policies and all final-window measurements.

Target rule	Stable / cases	Rate
Mixture-only	7/12	58.3%
Nash-only	10/12	83.3%
Combined	11/12	91.7%

in 10 of these comparisons. In the remaining two archives, only the mixture is sub-value, so the individual-policy control is not trained. These results show that aggregation can improve both target exploitability and the policy learned through distillation.

Mixture and equilibrium targets. The combined rule uses the mixture when its exact exploitability is sub-value and an equilibrium otherwise. In the matched comparison on Leduc and river over 12 game–seat–seed settings, it achieves stable sub-value outcomes in 91.7% of cases, compared with 83.3% for Nash-only and 58.3% for mixture-only (Table 4). Combined therefore yields one additional stable outcome over Nash-only. The result supports a modest advantage in this comparison, rather than a uniform advantage for archive targets. Nash-only retains the shared saved initializer and acceptance checks; it is distinct from distillation directly from the unmodified model. Appendix B.7 specifies the matched settings and how the combined outcomes are obtained.

Initialization. Across nine seat-1 runs, the checkpoint nearest to the target yields eight stable sub-value outcomes, compared with seven for the terminal search checkpoint. Both yield sub-value final policies in eight runs; the difference is persistence through the final window. A separate direct-from-base study produces no final policy in all nine runs after updates are rejected. That comparison changes both the target and initialization and does not establish that equilibrium distillation generally fails (Appendix B.6).

6 Conclusion and Limitations

High payoffs against fixed opponents do not establish low exploitability. We evaluate complete LLM policies against exact best responses, relate exploitability to adversary fidelity, update geometry, and neural realization, and bound the additional loss caused by imperfect distillation of a fixed target. Our method searches for model policies, constructs a fixed target from their mixture or an exact equilibrium, and distills that target while checking the expressed policy's exploitability: across both seats in Kuhn, Leduc, and a fixed-board river endgame, 94.4% of the 18 Qwen3-4B runs achieve stable sub-value outcomes. The matched comparisons show that target construction affects final exploitability, while gains at selected checkpoints need not persist through later training.

Limitations. Each exploitability evaluation applies to a specified finite game and the policy extracted under the declared model interface. In the external-policy audit, action probabilities are estimated from valid sampled responses. Exact best-response evaluation of these estimates does not remove sampling uncertainty or provide confidence bounds for the underlying models. The training evaluation covers Qwen3-4B, three seeds, and a single river board; its conclusions have yet to be established for other models and larger games. Evidence for preserving general capabilities is limited to one Kuhn adapter. The additional verification in Appendix C is specific to the saved action probabilities.

In the separate numerical audit, the distillation bound holds at every evaluated checkpoint, but falls below the sub-value threshold only on Kuhn. The theory also does not ensure that training reaches a sub-value policy from an arbitrary initializer, and one Leduc run remains above the threshold. The comparison with distillation from the unmodified model changes both the target and initialization; the matched ablations examine these choices separately. The opponent and update comparisons use three seeds and fixed training budgets. The fixed-target results measure refinement from saved search checkpoints, and the target comparison conditions on a shared initializer and acceptance rule. A modest observed difference in stable-outcome rates does not establish that one target source is preferable in every setting.

USE OF AI

We used large language models to assist with experimental code implementation, analysis and visualization of results, theoretical exposition, and manuscript polish and revision. The authors reviewed the AI-assisted contributions and take full responsibility for the methods, results, claims, and final manuscript.

ETHICS STATEMENT

This work studies how to identify and reduce worst-case strategic weaknesses in language-model policies. Exact auditing can reveal exploitable behavior and is therefore dual use, but our experiments concern bounded, synthetic games rather than deployed systems or human participants. Each exploitability measurement states its game and model interface. General capability is evaluated separately on the same model. The artifacts support reproducible evaluation and defensive training. The study uses no private data or human-subject data.

REPRODUCIBILITY STATEMENT

The supplementary code provides finite game definitions, policy extraction, exact best-response evaluation, target construction, and distillation. Retained policy tables allow the checkpoint exploitabilities in the opponent and update comparisons to be recomputed without model training. The package also includes experiment settings, outcome measurements, rational witness verification, and conditional Lean formalizations. The appendix specifies the evaluation interface and training procedures; the supplementary README states the available records and the limits of their coverage.

REFERENCES

Mohammed Alshiekh, Roderick Bloem, Ruediger Ehlers, Bettina Könighofer, Scott Niekum, and Ufuk Topcu. Safe reinforcement learning via shielding. *arXiv preprint arXiv:1708.08611*, 2017. AAAI 2018.

Karthika Arumugam, Kiran Kumar Manku, and Amit Dhanda. Safe equilibrium policy optimization for strategic agent policies. *arXiv preprint arXiv:2605.30854*, 2026.

Noam Brown and Tuomas Sandholm. Solving imperfect-information games via discounted regret minimization. *arXiv preprint arXiv:1809.04040*, 2018. AAAI 2019.

Noam Brown, Adam Lerer, Sam Gross, and Tuomas Sandholm. Deep counterfactual regret minimization. *arXiv preprint arXiv:1811.00164*, 2018. ICML 2019.

Neil Burch, Matej Moravcik, and Martin Schmid. Revisiting CFR+ and alternating updates. *Journal of Artificial Intelligence Research*, 64:429–443, 2019. doi: 10.1613/jair.1.11370.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Jeremy M. Cohen, Elan Rosenfeld, and J. Zico Kolter. Certified adversarial robustness via randomized smoothing. *arXiv preprint arXiv:1902.02918*, 2019. ICML 2019.

Will Drevo and Isaac Hendley. treys: A pure-Python poker hand evaluation library. `https://github.com/ihendley/treys`, 2023.

Gabriele Farina, Christian Kroer, and Tuomas Sandholm. Faster game solving via predictive blackwell approachability: Connecting regret matching and mirror descent. *arXiv preprint arXiv:2007.14358*, 2020. AAAI 2021.

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

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021. URL https://arxiv.org/abs/2009.03300.

Daniel Hennes, Dustin Morrill, Shayegan Omidshafiei, et al. Neural replicator dynamics. *arXiv preprint arXiv:1906.00190*, 2019. AAMAS 2020.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. *arXiv preprint arXiv:2212.04089*, 2022.

Michael Johanson, Nolan Bard, Neil Burch, and Michael Bowling. Finding optimal abstract strategies in extensive-form games. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence*, pp. 1371–1379, 2012.

Guy Katz, Clark Barrett, David Dill, Kyle Julian, and Mykel Kochenderfer. Reluplex: An efficient SMT solver for verifying deep neural networks. *arXiv preprint arXiv:1702.01135*, 2017. CAV 2017.

Daphne Koller, Nimrod Megiddo, and Bernhard von Stengel. Efficient computation of equilibria for extensive two-person games. *Games and Economic Behavior*, 14(2):247–259, 1996.

Marc Lanctot, Vinicius Zambaldi, Audrunas Gruslys, et al. A unified game-theoretic approach to multiagent reinforcement learning. *arXiv preprint arXiv:1711.00832*, 2017. NeurIPS 2017.

Boning Li, Baoxiang Wang, and Longbo Huang. PokerSkill: LLMs can play expert-level poker without training or solvers. *arXiv preprint arXiv:2605.30094*, 2026.

Minhua Lin, Enyan Dai, Hui Liu, et al. How far are LLMs from professional poker players? revisiting game-theoretic reasoning with agentic tool use. *arXiv preprint arXiv:2602.00528*, 2026.

Bo Liu, Leon Guertler, Simon Yu, et al. SPIRAL: Self-play on zero-sum games incentivizes reasoning via multi-agent multi-turn reinforcement learning. *arXiv preprint arXiv:2506.24119*, 2025.

Edward Lockhart, Marc Lanctot, Julien Pérolat, Jean-Baptiste Lespiau, Dustin Morrill, Finbarr Timbers, and Karl Tuyls. Computing approximate equilibria in sequential adversarial games by exploitability descent. *arXiv preprint arXiv:1903.05614*, 2019. IJCAI 2019.

Carlos Martin and Tuomas Sandholm. ApproxED: Approximate exploitability descent via learned best responses. *arXiv preprint arXiv:2301.08830*, 2023.

Stephen McAleer, Gabriele Farina, Marc Lanctot, and Tuomas Sandholm. ESCHER: Eschewing importance sampling in games by computing a history value function to estimate regret. *arXiv preprint arXiv:2206.04122*, 2022. ICLR 2023.

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, Christos Papadimitriou, and Georgios Piliouras. Cycles in adversarial regularized learning. *arXiv preprint arXiv:1709.02738*, 2017. SODA 2018.

Vlad Murgoci, Matthijs Spaan, and Yaniv Oren. AlphaExploitem: Going beyond the Nash equilibrium in poker by learning to exploit suboptimal play. *arXiv preprint arXiv:2605.09150*, 2026.

Michael Noukhovitch, Samuel Lavoie, Florian Strub, and Aaron Courville. Language model alignment with elastic reset. *arXiv preprint arXiv:2312.07551*, 2023.

Julien Perolat, Bart De Vylder, Daniel Hennes, et al. Mastering the game of Stratego with model-free multiagent reinforcement learning. *Science*, 378(6623):990–996, 2022.

Marc-Antoine Provost, Nejc Ilenic, Christopher Solinas, and Philippe Beardsell. GTO wizard benchmark. *arXiv preprint arXiv:2603.23660*, 2026.

Andrei A. Rusu, Sergio Gomez Colmenarejo, Caglar Gulcehre, Guillaume Desjardins, James Kirkpatrick, Razvan Pascanu, Volodymyr Mnih, Koray Kavukcuoglu, and Raia Hadsell. Policy distillation. *arXiv preprint arXiv:1511.06295*, 2015. ICLR 2016.

Can Savcı. GARIP: A running-average moving reference for last-iterate self-play in two-player zero-sum games. *arXiv preprint arXiv:2606.22688*, 2026.

Leyang Shen, Yang Zhang, Xiaoyan Zhao, Chun Kai Ling, and Tat-Seng Chua. Enhancing decision-making with large language models through multi-agent fictitious play. *arXiv preprint arXiv:2606.19308*, 2026.

Pratham Singla, Shivank Garg, and Vihan Singh. Poker arena: Multi-axis profiling of strategic reasoning and memory in LLMs. *arXiv preprint arXiv:2606.13815*, 2026.

Samuel Sokota, Ryan D'Orazio, J. Zico Kolter, et al. A unified approach to reinforcement learning, quantal response equilibria, and two-player zero-sum games. *arXiv preprint arXiv:2206.05825*, 2022. ICLR 2023.

Eric Steinberger, Adam Lerer, and Noam Brown. DREAM: Deep regret minimization with advantage baselines and model-free learning. *arXiv preprint arXiv:2006.10410*, 2020.

Oskari Tammelin. Solving large imperfect information games using CFR+. *arXiv preprint arXiv:1407.5042*, 2014.

Bernhard von Stengel. Efficient computation of behavior strategies. *Games and Economic Behavior*, 14(2):220–246, 1996.

Mingzhi Wang, Chengdong Ma, Qizhi Chen, et al. Magnetic preference optimization: Achieving last-iterate convergence for language model alignment. *arXiv preprint arXiv:2410.16714*, 2024.

Eric Wong and J. Zico Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. *arXiv preprint arXiv:1711.00851*, 2017. ICML 2018.

Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo-Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, and Ludwig Schmidt. Robust fine-tuning of zero-shot models. *arXiv preprint arXiv:2109.01903*, 2021.

Hang Xu, Kai Li, Haobo Fu, Qiang Fu, Junliang Xing, and Jian Cheng. Dynamic discounted counterfactual regret minimization. In *International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=6PbvbLyqT6`.

Hang Xu, Kai Li, Haobo Fu, Qiang Fu, Junliang Xing, and Jian Cheng. Deep (predictive) discounted counterfactual regret minimization. *arXiv preprint arXiv:2511.08174*, 2025.

Huining Yuan, Zelai Xu, Zheyue Tan, et al. MARSHAL: Incentivizing multi-agent reasoning via self-play with strategic LLMs. *arXiv preprint arXiv:2510.15414*, 2025. Earlier versions titled "MARS: Reinforcing Multi-Agent Reasoning of LLMs through Self-Play in Strategic Games".

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4791–4800, 2019. doi: 10.18653/v1/P19-1472.

Richard Zhuang, Akshat Gupta, Richard Yang, Aniket Rahane, Zhengyu Li, and Gopala Anumanchipalli. PokerBench: Training large language models to become professional poker players. *arXiv preprint arXiv:2501.08328*, 2025. AAAI 2025.

Martin Zinkevich, Michael Johanson, Michael Bowling, and Carmelo Piccione. Regret minimization in games with incomplete information. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 1729–1736, 2007.

A Related Work

This appendix collects the related-work discussion: strategic-reasoning benchmarks, additional equilibrium methods, and model-preservation techniques beyond the summary in the main text. Also see Table 6.

A.1 Strategic-reasoning benchmarks

PokerBench measures action agreement with expert labels (Zhuang et al., 2025). PokerSkill, the GTO Wizard benchmark, and Poker Arena measure approximate game-theoretic-optimal action quality, tournament performance, memory, or belief tracking (Li et al., 2026; Provost et al., 2026; Singla et al., 2026). Solver-assisted agents can be highly effective when tool use is allowed (Lin et al., 2026). These evaluations ask whether an agent takes strong actions under their sampling protocol. Exact exploitability instead asks whether any opponent strategy can systematically profit from the complete fixed policy. Action agreement measures sampled decisions; exact evaluation supplies a worst-case exploitability bound.

Multi-agent fictitious-play prompting and explicit exploitation methods study adaptation against selected opponents (Shen et al., 2026; Murgoci et al., 2026). Our opponent-pool control shows that exploitation skill and resistance to exploitation are distinct objectives.

A.2 Additional equilibrium methods

Counterfactual regret minimization and its discounted and predictive variants provide the main tabular foundation for imperfect-information game solving (Zinkevich et al., 2007; Tammelin, 2014; Burch et al., 2019; Brown & Sandholm, 2018; Xu et al., 2024). Magnetic mirror descent and related anchored dynamics target stable last-iterate behavior (Sokota et al., 2022; Perolat et al., 2022; Wang et al., 2024; Savcı, 2026). Our search loop inherits the CFR-BR pattern: one seat follows regret updates while the other supplies an exact best response (Johanson et al., 2012). Double-oracle and PSRO methods expand a policy population against best responses and recompute a meta-strategy in the restricted game (McMahan et al., 2003; Lanctot et al., 2017). Our archive program shares the restricted-master simplex but fixes the population to realized checkpoints from one LLM. Its output is a teacher for the same model, and its primal–dual witness establishes the target's exploitability before neural policy distillation.

Function-approximation solvers replace tabular values, regrets, or policies with neural networks (Brown et al., 2018; Steinberger et al., 2020; Xu et al., 2025; Heinrich & Silver, 2016; McAleer et al., 2022; Hennes et al., 2019; Martin & Sandholm, 2023). Their approximation error is analyzed as part of the solver. Our decomposition instead treats the realized LLM policy as an independently evaluated object and measures the target-to-policy discrepancy at every round.

A.3 Worst-case verification and model preservation

Formal robustness analysis verifies behavior under a declared perturbation set (Katz et al., 2017; Wong & Kolter, 2017; Cohen et al., 2019). Our perturbation set is strategic rather than geometric: the opponent may choose any feasible realization plan. Sequence form makes the resulting worst case a linear program, and the rational-witness layer makes the serialized target independently checkable.

Safety shields monitor or correct actions that violate a formal specification (Alshiekh et al., 2017). Our update-acceptance rule applies the same accept–reject principle at the level of a complete model update. Its predicate is membership in the exact safe-policy polytope, and restoration operates on the adapter before final-window and saved-policy evaluation. Model merging, elastic constraints, and data anchors provide complementary mechanisms for preserving general behavior (Wortsman et al., 2021; Ilharco et al., 2022; Noukhovitch et al., 2023); the functional projection evaluates that behavior separately on the same adapter.

B Detailed Experiment Results

Table 5: **Exploitability across all evaluation runs.** Means and two-sided 95% percentile intervals. The lower block retains the endpoint and sub-value outcome counts for all runs.

Game / method	Measurement	Mean	Lower	Upper
Kuhn / ours	selected	.177	.033	.448
	final	.177	.033	.448
	P90	.177	.033	.449
Leduc / ours	selected	.696	.431	.943
	final	.793	.585	1.033
	P90	.907	.643	1.286
River / ours	selected	.493	.369	.606
	final	.505	.391	.609
	P90	.565	.506	.619
Negotiation / mixture	selected	.423	.266	.580
	final	.452	.300	.603
	P90	.492	.333	.651
Negotiation / Nash	selected	.611	.212	1.010
	final	.615	.215	1.010
	P90	.624	.263	.985

Game	Method	Final / all	Sub-value / all	Stable / all
Kuhn	ours	6/6	6/6	6/6
Leduc	ours	6/6	5/6	5/6
River	ours	6/6	6/6	6/6
Negotiation	mixture	5/6	5/6	5/6
Negotiation	Nash	5/6	4/6	4/6

B.1 External-policy audit: extraction and scope

The audit in Section 2 contains eight systems, two policies per system (for different seats), and six information sets per seat. Table 7 reports the sampling budget per information set and the overall fraction of outputs parsed as legal actions. Unparsed responses are excluded from the denominator. All draws use temperature 1. SPIRAL and MARSHAL use the prompts and parsers from their original papers; SEPO uses the reproduction's prompt format. GPT-5.5 and Claude Opus 4.8 use system instructions with reasoning enabled; Qwen3-8B uses neither.

Table 7: **Exact policy evaluation follows finite-sample extraction.** Parse rates are percentages over all six information sets in each seat.

System	Samples per information set	Parse rate, seat 0	Parse rate, seat 1
SPIRAL-4B	100	100.0	100.0
MARSHAL-Kuhn-4B	100	100.0	100.0
MARSHAL-Generalist-4B	100	100.0	100.0
MARSHAL-Generalist-8B	100	100.0	99.7
SEPO reproduction	200	99.4	99.6
GPT-5.5	100	60.5	64.5
Claude Opus 4.8	60	100.0	100.0
Qwen3-8B prompting	200	100.0	100.0

Scope of the exact evaluation. The sequence-form solver computes worst-case expected payoff for a specified policy table. It introduces no match-sampling error, but it does not remove uncertainty in the estimated action probabilities. The audit values are therefore exact for the retained tables and

Table 6: **Training and evaluation choices in related methods.** Equilibrium rows name update profiles. Adversary levels are S0 (none), S1 (self-play), S2 (scripted), S3 (population best response), and S4 (exact best response). Marks denote exact (✓), estimated ((✓)), or absent (✗) measurements.

method	adversary	update	policy	realization measured?	own Expl reported?	our eval. ($\times\lvert v\rvert$)
Equilibrium computation (no language model in the loop)						
CFR / CFR+ / DCFR	profile[1]	regret	—	—	✓	—
MMD / R-NaD	profile[1]	proximal	—	—	✓	—
neural CFR family	sampled profile[1]	regret	net	✗	(✓)	—
NFSP	S1+S3	PG+average	net	✗	(✓)	—
PSRO / double oracle	S3	meta	net	✗	(✓)	—
LLM game players (current literature)						
Qwen3-8B prompting	S0	none	LLM	✗	✗	8.0
frontier prompting[2]	S0	none	LLM	✗	✗	2.8–3.5
SPIRAL	S1	PG	LLM	✗	≈ 0[3]	5–7
MARSHAL / MARS	S1	PG	LLM	✗	✗	3.3–5.5
GOE-LLM	pool best response[4]	PG	LLM	✗	✗	—[4]
SEPO	S2	PG	LLM	✗	0[5]	2.5
VERGE	S4 scalar[6]	GRPO	LLM	(✓)	✓	—[6]
ISO-LLM	?[7]	PG	LLM	✗	5.2 mBB[7]	—
ours	**S4**	**mixture/distill.**	**LLM**	✓	✓	**<1**

[1]Current-profile feedback (sampled in Deep CFR/DREAM) [2]GPT-5.5 $3.5 \times |v|$ (parse 60.5%; $6.90 \times |v|$ if count parse failure), Opus $2.8 \times |v|$ (parse 100%). [3]Only zero on SPIRAL's Kuhn variant [4]Used to exploit others, did not measure exploitability [5]For their fixed adversary only; our reproduction showed $1.46/2.51 \times |v|$. [6]Scalar residual on one-shot matrix games; no sequential policy is available to evaluate. [7]Adversary unspecified.

point estimates for the underlying sampled models. They are not finite-sample confidence bounds on those models. Historical extraction-time code revisions were not recorded for every system, and provider checkpoint revisions are unavailable for the API models. The precise additional strategy-instruction string for the two API configurations was also not retained. The said parameters were not reproduced.

Source-game and reproduction distinctions. SPIRAL's original modified Kuhn variant had absolute exploitability zero. Our results only measure its exploitability on canonical Kuhn. For SEPO, we reproduced the described pipeline on Qwen3-4B ourselves, including source data generation and LoRA fine tuning.

B.2 Opponent and update comparisons

Tables 8 and 9 report every seed in the opponent and update comparisons. All runs reach the 100-round endpoint. The selected policy is chosen only from the four saved checkpoints; the trajectory minimum is not substituted for a saved policy.

Table 8: **Exact feedback improves selected checkpoints, but later updates can lose that improvement.** All six runs use river seat 0, mirror descent, fresh model initialization, 100 rounds and 25 inner passes per round. Values are exploitability in multiples of $|v|$. No run has a stable sub-value outcome. Selection uses rounds 25, 50, 75 and 100.

Seed	Feedback	Initial	Selected	Final	P90	Tail max
2026	Exact	53.021	5.518	51.244	53.078	53.126
2026	Self-copy	53.021	50.361	50.361	52.549	52.745
2027	Exact	53.021	4.925	4.925	5.244	5.345

Seed	Feedback	Initial	Selected	Final	P90	Tail max
2027	Self-copy	53.021	12.818	12.818	16.136	16.731
2028	Exact	53.021	31.696	31.696	35.885	36.610
2028	Self-copy	53.021	50.205	51.974	52.017	52.019

Table 9: **Complete three-seed update and target comparison.** Values are exploitability in multiples of $|v|$ for Leduc seat 0. The starting checkpoints have exploitability 4.835, 1.247 and 3.229 for seeds 2026, 2027 and 2028, respectively. A check denotes a stable sub-value outcome. All arms retain their final evaluated policy.

Seed	Strategy	Selected	Final	P90	Tail max	Stable
2026	Fixed archive target	1.112	1.112	2.413	2.750	✗
2026	Mirror descent	1.698	2.088	1.954	2.088	✗
2026	Moving RM$^+$	3.914	3.914	4.877	5.048	✗
2026	Direct replacement	18.031	19.967	20.585	21.798	✗
2026	Frozen first RM$^+$ target	19.701	23.923	24.317	26.717	✗
2027	Fixed archive target	0.958	0.958	0.958	0.958	✓
2027	Mirror descent	1.113	1.584	1.584	1.619	✗
2027	Moving RM$^+$	1.792	2.114	3.207	3.567	✗
2027	Direct replacement	12.131	12.131	14.201	15.474	✗
2027	Frozen first RM$^+$ target	32.990	38.244	38.283	38.432	✗
2028	Fixed archive target	1.268	1.268	1.315	1.389	✗
2028	Mirror descent	1.417	1.516	1.599	1.686	✗
2028	Moving RM$^+$	2.529	2.529	2.895	3.357	✗
2028	Direct replacement	13.715	13.715	14.403	15.069	✗
2028	Frozen first RM$^+$ target	33.566	33.566	34.811	35.419	✗

The frozen-first target is the initial RM$^+$ requested update, held fixed throughout distillation. Fixing the first RM$^+$ target does not confer a sub-value guarantee; its exploitability is $28.787 \times |v|$ for seed 2026. The archive control also trains its target when the mixture is above the threshold, without applying the equilibrium fallback.

B.3 Aggregate convex-target comparison

Table 10: **Convex policy aggregation usually survives neural realization.** The 11 trained pairs use the same archive, retained initializer, optimizer, controller, numerical configuration, and horizon; only the fixed target changes. Entries are mean exploitability in multiples of $|v|$. Lower is better. †Best-policy minus convex-mixture exploitability; positive values favor convex aggregation.

target	selected	final	window P90	lower P90	stable/trained
best archived policy	.421	.426	.429	1/11	11/11
convex archive policy	**.368**	**.369**	**.374**	**10/11**	11/11
paired difference†	+.0528	+.0566	+.0546	median +.0339	—

B.4 Complete fixed-evaluation outcomes

Table 11: **All 18 poker runs have measured endpoints.** Values are exploitability in multiples of $|v|$. Tail max is the maximum over the final 20 rounds. Stable requires selected, final, and tail max below one; a low P90 alone is insufficient. Individual rows are shown for auditability; the main text reports aggregate means, intervals, and stable sub-value rates.

game	seat	seed	init.	selected	final	P90	tail max	stable
Kuhn	0	2026	0.059	0.024	0.024	0.024	0.024	✓
Kuhn	1	2026	1.387	0.845	0.845	0.846	0.846	✓
Leduc	0	2026	0.886	0.776	0.776	0.807	0.810	✓
Leduc	1	2026	0.804	0.688	0.688	0.693	0.694	✓
River endgame	0	2026	1.325	0.337	0.358	0.556	0.861	✓
River endgame	1	2026	0.825	0.524	0.524	0.531	0.541	✓
Kuhn	0	2027	0.160	0.032	0.032	0.032	0.032	✓
Kuhn	1	2027	0.188	0.064	0.064	0.064	0.064	✓
Leduc	0	2027	1.247	0.958	0.958	0.959	0.961	✓
Leduc	1	2027	2.238	1.139	1.318	1.804	1.955	✗
River endgame	0	2027	2.318	0.248	0.285	0.437	0.564	✓
River endgame	1	2027	0.866	0.632	0.635	0.636	0.637	✓
Kuhn	0	2028	0.067	0.029	0.029	0.031	0.041	✓
Kuhn	1	2028	0.160	0.066	0.066	0.066	0.066	✓
Leduc	0	2028	1.391	0.144	0.547	0.547	0.700	✓
Leduc	1	2028	2.896	0.469	0.469	0.631	0.664	✓
River endgame	0	2028	1.104	0.644	0.653	0.655	0.662	✓
River endgame	1	2028	0.929	0.572	0.572	0.574	0.575	✓

B.5 Convex policy aggregation

Table 12: **Convex policy aggregation over six poker archives.** The linear program minimizes exploitability over the full policy hull. Support counts give nonzero mixture weights. The predefined target-selection rule uses exact Nash only when the convex mixture is not sub-value. The archive is the complete six-cell construction preceding distillation; it separates a secure target from a secure individual checkpoint, and the exact-Nash row is the equilibrium fallback.

game	seat	best policy	convex mixture	support	selected target
Kuhn	0	0.037	0.024	3	convex mixture
Kuhn	1	1.356	0.845	3	convex mixture
Leduc	0	0.719	0.589	7	convex mixture
Leduc	1	0.814	0.662	7	convex mixture
river	0	1.261	1.215	3	exact Nash
river	1	0.631	0.524	4	convex mixture

B.6 Initialization and direct-from-base comparisons

The nearest and terminal initializers share the selected target and the remaining distillation settings. The direct-from-base comparison additionally changes the target to an exact equilibrium. All nine direct-from-base runs stop after a rejected update before a final policy is available.

Table 13: **Matched initializer and direct-from-base comparisons.** Triples are selected/final/P90 exploitability. Checkmarks denote stable sub-value outcomes, including all final 20 rounds. The nearest and terminal initializers share the selected target and training settings. Direct-from-base additionally changes the target to an exact Nash policy; it produces no final policy in any row. Parentheses give the last evaluated prefix before stopping, not a final outcome.

game	seed	nearest checkpoint	terminal checkpoint	Nash from base
Kuhn	2026	0.845/0.845/0.846 ✓	0.845/0.845/0.846 ✓	— (6.296 at r3)
Leduc	2026	0.688/0.688/0.693 ✓	0.692/0.692/0.695 ✓	— (29.481 at r2)
River endgame	2026	0.524/0.524/0.531 ✓	0.525/0.525/0.527 ✓	— (58.325 at r9)
Kuhn	2027	0.064/0.064/0.064 ✓	0.064/0.064/0.064 ✓	— (6.273 at r3)
Leduc	2027	1.139/1.318/1.804 ✗	0.990/0.990/1.729 ✗	— (29.477 at r2)
River endgame	2027	0.632/0.635/0.636 ✓	0.636/0.636/0.693 ✓	— (56.796 at r9)
Kuhn	2028	0.066/0.066/0.066 ✓	0.066/0.066/0.066 ✓	— (6.253 at r3)
Leduc	2028	0.469/0.469/0.631 ✓	0.484/0.644/0.579 ✓	— (29.488 at r2)
River endgame	2028	0.572/0.572/0.574 ✓	12.196/12.196/21.792 ✗	— (58.155 at r9)
stable/all		8/9	7/9	0/9

B.7 Target-source comparisons

Table 4 reports the matched comparison: Leduc and river, both seats, and seeds 2026–2028 give 12 matched settings. Mixture-only and Nash-only train one policy each per setting, for 24 training runs. Mixture-only gives seven stable sub-value outcomes, Nash-only ten, and the combined rule eleven. The combined rule chooses the target from its exact exploitability before distillation. It does not choose whichever trained arm performs better; its outcome reuses the arm selected for each setting. Nash-only changes the target while retaining the shared initializer and acceptance rule; this comparison therefore conditions on that initialization procedure.

B.8 Convex aggregation versus the best archived policy

Table 14: **Complete convex-mixture versus best-policy target comparison.** Values are final-window P90 exploitability in multiples of $|v|$. Both policies in each trained pair have sub-value selected and final checkpoints and all final-20-round measurements below one. A dash means no final policy: the best archived target is not sub-value, so no neural training is performed. The 13 registered pairs test whether convex aggregation remains useful after neural distillation; eleven have two sub-value targets and receive matched training, and the other two stop before training. †The number is the exact target exploitability, not a neural-policy P90. Convex aggregation is lower in 10 of 11 trained pairs; the Leduc-seat-0/2026 pair is the sole reversal.

game	seat	seed	convex mixture	best policy	best-policy outcome
Kuhn	0	2026	**.024**	.037	stable
Kuhn	1	2026	.846	1.356†	—
Kuhn	0	2027	**.032**	.044	stable
Kuhn	1	2027	**.064**	.098	stable
Kuhn	0	2028	**.031**	.033	stable
Kuhn	1	2028	**.066**	.086	stable
Leduc	0	2026	.807	**.722**	stable
Leduc	1	2026	**.693**	.812	stable
Leduc	0	2027	.959	1.101†	—
river	1	2026	**.531**	.642	stable

game	seat	seed	convex mixture	best policy	best-policy outcome
river	1	2027	**.636**	.750	stable
river	0	2028	**.655**	.728	stable
river	1	2028	**.574**	.762	stable

B.9 Opponent-dependent policy evaluation

Table 15: **The evaluation conclusion changes with the opponent.** Exact mean payoff advantage across both seats, relative to equilibrium and normalized by $|v|$, for the same frozen canonical-Kuhn policies. Positive values beat the equilibrium payoff. The fixed pool is identical for every system; the exact best response is policy specific. The pool is the uniform mixture of the eight opposite-seat policies, exact Nash is a shared reference, and values are exact for the frozen extracted policy tables.

system	fixed pool	exact Nash	exact best response
SPIRAL-4B	-0.816	-2.000	-6.000
MARSHAL-Kuhn-4B	-0.774	-2.537	-4.933
MARSHAL-Generalist-4B	-0.644	-2.331	-4.273
MARSHAL-Generalist-8B	$+0.227$	-2.332	-3.380
SEPO reproduction	$+0.915$	-1.367	-1.985
GPT-5.5	$+1.901$	0.000	-3.181
Claude Opus 4.8	$+1.891$	0.000	-2.750
Qwen3-8B prompting	-2.700	-1.500	-7.500

C Full proofs

This appendix proves the two parts of Proposition 2 and Theorem 2 together with Corollary 2 (Threat representation), then Proposition 1, Propositions 3 and 4, and Proposition 5 with the rational exploitability bound. It also gives the archive optimization program and the rational proof record checked by the accompanying verifier.

C.1 Threat representation

Fix the audited seat and refine the sets $\mathcal{X}$ and $\mathcal{Y}$ of Section 3 to the sequence-form realization-plan polytopes of the audited seat and its opponent. By the perfect-recall sequence-form realization theorem, $\mathcal{Y}$ is the convex hull of the finite set Z of pure opponent realization plans. For fixed x, if $y = \sum_{z\in Z} \lambda_z z$ with $\lambda_z \geq 0$ and $\sum_z \lambda_z = 1$, then linearity gives $x^\top Ay = \sum_z \lambda_z x^\top Az$. This weighted average is no smaller than its least pure-plan term, while every pure plan itself belongs to $\mathcal{Y}$. Hence a linear best response is attained by some $z \in Z$. Writing the audited loss ℓ_z against z of Section 3 in the affine form $\ell_z(x) = a_z^\top x + \beta_z$ expresses the exploitability $E(x) = \max_{z\in Z} \ell_z(x)$ there as a maximum of affine functions. For seat 0, $\ell_y(x) = v^* - x^\top Ay$; the seat-1 expression is symmetric.

Proof of Proposition 2, safe-set and one-sided-loss part. The condition $E(x) \leq \tau$ holds exactly when $a_z^\top x + \beta_z \leq \tau$ for every $z \in Z$. Hence

$$S_\tau = \mathcal{X} \cap \bigcap_{z\in Z} \{x : a_z^\top x + \beta_z \leq \tau\}.$$

The sequence-form set $\mathcal{X}$ is a bounded polyhedron, and each additional constraint is a closed halfspace. Their finite intersection is a closed convex polytope. Equivalently, for $x_1, x_2 \in S_\tau$ and $\alpha \in [0, 1]$,

$$E(\alpha x_1 + (1-\alpha)x_2) \leq \alpha E(x_1) + (1-\alpha)E(x_2) \leq \tau.$$

Define $D(u) = \max\{0, \max_z a_z^\top u\}$. Then $d(m \to x) = D(x - m)$. The function D is the support function of $\mathrm{conv}(\{0\} \cup \{a_z : z \in Z\})$. It is therefore nonnegative, positively homogeneous, and subadditive. In particular,

$$d(m \to x) \le d(m \to q') + d(q' \to x).$$

For every z,

$$\ell_z(x) = \ell_z(m) + a_z^\top (x - m) \le E(m) + D(x - m).$$

Taking the maximum over z proves $E(x) \le E(m) + d(m \to x)$. A support function need not be symmetric or strictly positive away from zero, so d is a one-sided gauge. □

For seat 0, the gauge is the sequence-form linear program (LP)

$$d(m \to x) = \max\{0, \max_{y \in \mathcal{Y}} (m - x)^\top A y\};$$

for seat 1 it is $\max\{0, \max_{r \in \mathcal{X}} r^\top A(x - m)\}$. The gauge ignores displacements invisible to all opponent threats.

Proof of Theorem 2. Because $T \ge 1$, the archive hull is a nonempty compact polytope. The function E is continuous as the maximum of finitely many affine functions, so the archive minimum is attained and q exists. Every played plan belongs to the archive hull, so

$$E(q) \le \min_{t \le T} E(x_t) \le \frac{1}{T} \sum_{t=1}^{T} E(x_t) \le B_T,$$

where the final inequality is Theorem 1.

Fix a pure opponent plan z. Let P_q^z and P_p^z be the distributions over terminal histories induced by target q and student p. The finite-tree KL chain rule and perfect recall give

$$D_{\mathrm{KL}}(P_q^z \| P_p^z) = \sum_I \Pr_{q,z}[I] D_{\mathrm{KL}}(q_I \| p_I), \qquad \Pr_{q,z}[I] \le r_q(I) \le w_I.$$

Nonnegativity of KL therefore gives $D_{\mathrm{KL}}(P_q^z \| P_p^z) \le \Phi_q(p)$. Strictly positive student action probabilities ensure that P_q^z is absolutely continuous with respect to P_p^z.

For finite distributions, the elementary Hellinger–KL and Hellinger–total-variation inequalities imply

$$D_{\mathrm{TV}}(P_q^z, P_p^z) \le \sqrt{D_{\mathrm{KL}}(P_q^z \| P_p^z)} \le \sqrt{\Phi_q(p)}.$$

Because terminal utility has width Δ, the corresponding payoff change is at most $\Delta D_{\mathrm{TV}}(P_q^z, P_p^z)$. Hence

$$\ell_z(p) - \ell_z(q) \le \Delta \sqrt{\Phi_q(p)} \quad \text{for every } z.$$

The right-hand side is nonnegative, so maximizing over threats and zero gives $d(q \to p) \le \Delta \sqrt{\Phi_q(p)}$. Proposition 2 and the archive bound now yield

$$E(p) \le E(q) + d(q \to p) \le B_T + \Delta \sqrt{\Phi_q(p)}.$$

The safe-set conclusion follows by comparison with τ. For an exact Nash target, the same fixed-target argument begins from $E(q) = 0$. □

Corollary 2 (Same-archive averaging comparison)**.** *Let $a_t \ge 0$ with $\sum_t a_t = 1$, let $\bar{x}_a = \sum_t a_t x_t$, and define $G(a) = E(\bar{x}_a) - E(q)$. Then $G(a) \ge 0$ and*

$$E(p) \le E(\bar{x}_a) - [G(a) - \Delta \sqrt{\Phi_q(p)}].$$

Thus p is strictly less exploitable than $\bar{x}_a$ whenever the archive gain exceeds the neural realization bound.

Proof of Corollary 2. The nonnegative weights sum to one, so the scheduled average $\bar{x}_a$ belongs to the archive hull. Optimality of q gives $E(q) \le E(\bar{x}_a)$ and therefore $G(a) \ge 0$. Theorem 2 gives $E(p) \le E(q) + \Delta \sqrt{\Phi_q(p)}$. Substituting $E(q) = E(\bar{x}_a) - G(a)$ proves the displayed bound. The strict conclusion follows when the bracketed difference is positive. □

Table 16: The reach-weighted distillation bound covers every logged checkpoint. "Final $< |v|$" reports the percentage of trajectories whose analytic upper bound, not only observed exploitability, is below one game value.

game	trajectories	checkpoints	coverage	final $< \|v\|$
Kuhn	21	1,830	100%	61.9%
Leduc	23	2,000	100%	0%
river endgame	29	2,156	100%	0%
negotiation	11	1,017	100%	None
all	84	7,003	100%	15.5%

C.2 Optimality and KL-rate tightness of the distillation bound

Proof of Proposition 1. Fix nonnegative local discrepancies $\kappa = (\kappa_I)_I$ and consider any separable majorant $\Psi_{\widetilde{w}}(\kappa) = \sum_I \widetilde{w}_I \kappa_I$. If $c_z(I) \le \widetilde{w}_I$ for all z, I, multiplying each coordinate by $\kappa_I \ge 0$ and summing gives

$$\sum_I c_z(I)\kappa_I \le \Psi_{\widetilde{w}}(\kappa) \quad \text{for every } z, \kappa \ge 0.$$

Conversely, suppose this inequality holds for every nonnegative κ. For any fixed z and J, choose the one-hot discrepancy $\kappa_J = 1$ and $\kappa_I = 0$ for $I \ne J$. The inequality reduces to $c_z(J) \le \widetilde{w}_J$. Thus uniform separable domination is equivalent, not merely implied by, coordinatewise reach domination. For the fixed game, the least such weight is therefore $\sup_z c_z(I)$. If only the perfect-recall envelope $c_z(I) \le r_q(I)$ is retained, robustness to every reach table consistent with that envelope holds exactly when $\widetilde{w}_I \ge r_q(I)$: sufficiency is the preceding argument, while necessity uses the admissible saturated table $c_z(I) = r_q(I)$ and the same one-hot discrepancy. Hence the target's own reach is pointwise least in this robust bound class, and the implementation's positive floor can only enlarge the majorant.

Finally, the KL exponent cannot be improved to a universal linear dependence. Take the binary target $q = (1/2, 1/2)$ and student $p_\delta = (1/2 + \delta, 1/2 - \delta)$ with $0 < \delta \le 1/4$. Direct calculation gives

$$D_{\mathrm{KL}}(q\|p_\delta) = -\tfrac{1}{2}\log(1 - 4\delta^2) \le 4\delta^2, \qquad D_{\mathrm{TV}}(q, p_\delta) = \delta.$$

The unit-range payoff $u(0) = 1, u(1) = 0$ also has $\mathbb{E}_{p_\delta} u - \mathbb{E}_q u = \delta$. Given any finite $C \ge 0$, choose δ small enough that $4C\delta < 1$; then

$$C D_{\mathrm{KL}}(q\|p_\delta) \le 4C\delta^2 < \delta.$$

Therefore no game-independent bound controlling every bounded threat payoff can replace the square-root-order conversion by $C D_{\mathrm{KL}}$ without an additional margin, determinism, or other structural premise. □

C.3 Tracking and safe-state preservation

Proposition 3 (Conditional finite-time tracking)**.** *Let $c_t = d(m_t \to x_t)$. If $c_{t+1} \le \rho c_t + L v_t + b_t$ for constants $0 \le \rho < 1$ and $L \ge 0$, and nonnegative target variation v_t and residual b_t, then*

$$E(x_T) \le E(m_T) + \rho^T c_0 + \sum_{t=0}^{T-1} \rho^{T-1-t}(L v_t + b_t).$$

Proof of Proposition 3. Repeated substitution in $c_{t+1} \le \rho c_t + L v_t + b_t$ gives

$$c_T \le \rho^T c_0 + \sum_{t=0}^{T-1} \rho^{T-1-t}(L v_t + b_t).$$

Formally, the claim follows by induction: it is immediate at $T = 0$; multiply the bound at T by ρ, add $L v_T + b_T$, and collect the geometric powers to obtain the bound at $T + 1$. Proposition 2 gives $E(x_T) \le E(m_T) + c_T$, which proves the stated result. □

For a fixed target, $v_t = 0$. If $b_t \le \bar{b}$, then

$$E(x_T) \le E(m) + \rho^T c_0 + \bar{b}\frac{1-\rho^T}{1-\rho}.$$

Whenever this envelope lies below the security threshold, it bounds the terminal policy's exploitability.

Proof of Proposition 2, preservation part. The first accepted secure state lies in S_τ. Inductively, acceptance keeps the state in S_τ, while rejection restores the previous member of S_τ. The returned state is therefore in S_τ. □

C.4 Representation floors persist through convex archives

The card-blind policy classes in Appendix D are convex in realization-plan space. The following proposition states the general consequence of any proved convex representation floor.

Proposition 4 (Representation barrier)**.** *Let $C \subseteq \mathcal{X}$ be convex and suppose $F_C \le E(x)$ for every $x \in C$.*

1. *If $x_1, \dots, x_T \in C$, every policy in $\mathrm{conv}\{x_1, \dots, x_T\}$ has exploitability at least F_C.*
2. *If $E(q) \le \beta$ and $p \in C$, then $d(q \to p) \ge F_C - \beta$.*
3. *Under the hypotheses of Theorem 1, if every played policy x_t lies in C, then*

$$F_C - \frac{R_T^m}{T} - \bar{\gamma}_T \le \frac{1}{T}\sum_{t=1}^{T}(m_t - x_t)^\top A y_t.$$

Proof. Convexity of C places the complete archive hull inside C, which proves the first statement. Proposition 2 gives $F_C \le E(p) \le E(q) + d(q \to p) \le \beta + d(q \to p)$, which proves the second. For the third, averaging the class floor gives $F_C \le T^{-1}\sum_t E(x_t)$. Combining this inequality with the first bound in Theorem 1 and rearranging proves the result. □

Applying the proposition to a model interface additionally requires a proof of exact class membership.

C.5 Convex policy aggregation with a one-sided loss term

Let $P = [p_1, \dots, p_n]$ contain retained realization plans, let $w \in \Delta_n$, and let s be a fixed initializer. A family of security– realization trade-offs is

$$\min_{w \in \Delta_n} E(Pw) + \lambda d(Pw \to s), \qquad \lambda \ge 0.$$

With epigraph variables e, r, the full problem is

$$\begin{aligned} \min_{w,e,r} \quad & e + \lambda r \\ \text{s.t.} \quad & e \ge \ell_z(Pw) && \forall z \in Z, \\ & r \ge \ell_z(s) - \ell_z(Pw) && \forall z \in Z, \\ & r \ge 0, \quad w \ge 0, \quad \mathbf{1}^\top w = 1. \end{aligned}$$

The method in Section 4 uses the security endpoint $\lambda = 0$; varying λ traces the security–realization Pareto frontier.

Proposition 5 (Finite exact separation)**.** *Constraint generation that adds an exact best-response maximizer for each violated epigraph terminates after finitely many additions and returns a global optimum. At termination, feasible primal and dual solutions with matching objectives constitute an independently checkable optimality witness.*

Proof. The opponent has a finite set Z of pure realization plans, so the full LP has finitely many threat constraints. A failed exact separation step returns a violated constraint; it cannot already be in

the restricted master, whose candidate satisfies all active constraints. Thus the added maximizers are distinct and there are at most $|Z|$ additions.

At termination the restricted-master candidate is feasible for every full constraint. Every full-feasible point is also restricted-feasible, so restricted optimality makes the candidate objective no larger than any full-feasible objective. The candidate is therefore globally optimal. LP strong duality supplies a feasible dual point with the same objective. Conversely, weak duality shows directly that any feasible primal–dual pair with equal objectives establishes optimality of both; complementary slackness is an equivalent way to check this equality. □

C.6 Rational exploitability bound for a serialized policy

Floating-point scoring alone does not prove that serialized action probabilities form exact simplexes or that a solver's objective is exact. The deployment reconstruction represented by the proof record parses every decimal as a rational number, normalizes each information-set vector over the rationals, and multiplies conditional probabilities through the sequence-form tree to obtain an exactly feasible plan $\widehat{x}$. Let x_{raw} denote the unnormalized rational product used only to measure the displacement introduced by that deterministic simplex reconstruction.

Proposition 6 (Sound rational exploitability interval)**.** *Suppose a proof record contains: (i) an exactly feasible rational plan $\widehat{x}$; (ii) its exact pure best-response payoff b; (iii) feasible rational primal and dual game witnesses satisfying $v_L \le v^* \le v_U$; and (iv) a rational bound*

$$\Lambda \ge \max_z |\ell_z(x_{\mathrm{raw}}) - \ell_z(\widehat{x})|.$$

For seat 0, the exploitability of the reconstructed deployment policy lies in

$$[v_L - b,\ v_U - b].$$

and the verifier records the conservative serialization envelope

$$[v_L - b - \Lambda,\ v_U - b + \Lambda],$$

For seat 1 the corresponding intervals are $[b - v_U,\ b - v_L]$ and $[b - v_U - \Lambda,\ b - v_L + \Lambda]$. If the conservative envelope's upper endpoint is below the smallest admissible $|v|$, the reconstructed policy is strictly sub-value.

Proof. Exact dynamic programming over the finite information-set forest returns the optimal rational pure response to $\widehat{x}$, so b is exact for the reconstructed plan. Weak duality gives $v_L \le v^* \le v_U$. For seat 0, $E(\widehat{x}) = v^* - b$, and subtracting b gives $v_L - b \le E(\widehat{x}) \le v_U - b$; for seat 1, $E(\widehat{x}) = b - v^*$ gives $b - v_U \le E(\widehat{x}) \le b - v_L$.

The raw product need not be a feasible realization plan, but its finite affine threat envelope is still defined. Let z_r attain $E(x_{\mathrm{raw}})$. Since $\ell_{z_r}(\widehat{x}) \le E(\widehat{x})$,

$$E(x_{\mathrm{raw}}) - E(\widehat{x}) \le \ell_{z_r}(x_{\mathrm{raw}}) - \ell_{z_r}(\widehat{x}) \le \Lambda.$$

Choosing instead a threat z' attaining $E(\widehat{x})$ gives $E(\widehat{x}) - E(x_{\mathrm{raw}}) \le \Lambda$. Hence $|E(x_{\mathrm{raw}}) - E(\widehat{x})| \le \Lambda$. Expanding both endpoints of the reconstructed-policy interval gives the conservative serialization envelope used by the verifier. The final comparison is uniform over every game value consistent with the rational witnesses. □

The verifier replays the realization-flow equalities, game-value witnesses, pure best response, payoff decomposition, reconstruction displacement, and file hashes using rational arithmetic. A floating-point LP solver may propose a witness, but it is not part of the proof.

D The Blind Floor: Proof and Computation

Setup. Fix seat i; write $\mathcal{I}_i$ for its information sets, E_i for its sequence-to-flow incidence matrix, and e_i for the root-flow vector. Thus $Q_i = \{x \ge 0 : E_i x = e_i\}$ is its sequence-form realization polytope. Let A be the chance-weighted payoff matrix and Q_{opp} the opponent's polytope (Koller et al., 1996; von Stengel, 1996). The exploitability of a plan x is Definition 1 with the minimum over Q_{opp}: $\mathrm{Expl}(x) = v^* - \min_{y \in Q_{\mathrm{opp}}} x^\top A y$ for seat 0 (mirrored for seat 1). A behavioral policy is *card-blind* when it assigns identical action distributions to information sets that differ only in the private observation.

Proposition 7 (Card-blind floor)**.** *Suppose the private observation occurs once before the player's first action, and deleting it preserves public observation–action histories, action sets, and quotient predecessors. The resulting quotient has perfect recall, and a linear lift L maps its realization polytope $\widetilde{Q}$ exactly onto the card-blind realization plans. Their optimal exploitability is*

$$\min_{\pi \text{ blind}} \mathrm{Expl}(\pi) = v^* - \max_{\tilde{x}\in\widetilde{Q}} \min_{y\in Q_{\mathrm{opp}}} (L\tilde{x})^\top A y.$$

Quotient construction. Let Γ denote the original game and let φ delete the private observation from information-set labels, partitioning $\mathcal{I}_i$ into blind classes. By assumption (verified structurally from the serialized sequence-form tables for every reported card- or type-blind floor), the private observation occurs once, before the player's first action; deleting it uniformly preserves the player's public observation–action history; all members of a blind class share one action set; and the parents of a class's members lie in a single quotient class. The quotient $\tilde{\Gamma}$ is therefore a well-defined extensive structure whose sequences are (blind class, action) pairs.

Proof of Proposition 7. (1) Perfect recall of the quotient. In Γ, seat i's private observation is made once, at the root. A seat-i path of $\tilde{\Gamma}$ records exactly the public observation/action sequence of the corresponding Γ-path. Since Γ has perfect recall and φ removes the root-private symbol uniformly from every set, two quotient nodes in one blind class still agree on the full prior sequence of quotient sets and actions. Hence $\tilde{\Gamma}$ has perfect recall for seat i.

(2) Blind plans are the lifted quotient polytope. Define a card-blind realization plan to mean a plan induced by at least one card-blind behavioral policy; this avoids assigning behavioral probabilities at unreachable information sets directly from realization coordinates. Pullback along φ maps a quotient behavioral policy $\tilde{\pi}$ to the original policy that uses $\tilde{\pi}(\varphi(I))$ at every I. Conversely, because every blind class has a common action set, every card-blind π factors uniquely through φ. Thus pullback is a bijection between quotient behavioral policies and card-blind behavioral policies.

By the perfect-recall sequence-form realization theorem applied to $\tilde{\Gamma}$, the quotient realization plans induced by its behavioral policies are exactly $\tilde{Q} = \{\tilde{x} \geq 0 : \tilde{E}\tilde{x} = \tilde{e}\}$. Extend φ to own-action sequences by replacing each (I, a) with $(\varphi(I), a)$. Define L coordinatewise by $(L\tilde{x})_\sigma = \tilde{x}_{\varphi(\sigma)}$. Along any original path, the realization weight is the product of the policy factors on its own-action history; pullback makes these exactly the factors on the quotient history. Hence realization commutes with pullback. The forward direction gives $L\tilde{Q} \subseteq Q_i$ and only card-blind plans. For the reverse direction, factor an inducing card-blind policy through φ and apply the same commutation identity. Therefore $L\tilde{Q}$ is exactly the set of card-blind realization plans; since L is linear, it is a polytope.

(3) One LP; rationality. Therefore $\min_{\pi \text{ blind}} \mathrm{Expl}(\pi) = v^* - \max_{\tilde{x}\in\tilde{Q}} \min_{y\in Q_{\mathrm{opp}}} (L\tilde{x})^\top A y$. Dualizing the inner minimization over the opponent's flow constraints (standard sequence-form maximin) turns the max–min into a single linear program in $(\tilde{x}, q)$, where q contains the dual continuation values for the opponent's flow constraints. The blind and opponent realization polytopes are nonempty compact rational polytopes (uniform blind play witnesses nonemptiness), so continuity of the bilinear payoff gives a finite attained max–min value. Put the equivalent sequence-form LP in standard form, splitting each free continuation value into positive and negative parts. A rational LP with a finite attained optimum has an optimal basic feasible solution; because all flow, payoff, and lifting data are rational, that solution and the optimal value are rational. □

Computation and verification. We implement the quotient LP directly from the game dumps and solve it with HiGHS. The exact-rational verifier uses HiGHS only to propose realization plans, rounds through behavioral probabilities to obtain exactly feasible rational plans, and evaluates exact rational best responses. These primal–dual witnesses enclose both the full and quotient values, and interval arithmetic encloses their difference and its normalization by $|v^*|$; the value interval is required to exclude zero: the verified value interval $[v_L, v_U]$ maps through absolute value to $[a, b] = [v_L, v_U]$ or $[a, b] = [-v_U, -v_L]$, according to the sign of the endpoints, and if the nonnegative floor is enclosed by $[f_L, f_U]$, division monotonicity gives the sound normalized interval $[f_L/b, f_U/a]$. The exact Kuhn intervals contain the analytic rationals $5/18$ (seat 0) and $7/18$ (seat 1). The Leduc seat-0 floor is 0.9143935759 in native payoff units. The canonical river-board floors are $18.10 \times |v|$ and $20.10 \times |v|$. The same structural and rational-witness audit covers negotiation and encloses its two type-blind floors at $8.5 \times |v|$ and $4.0 \times |v|$.

Interpretation. The floor depends only on the game quotient; no model or training run enters the computation. It constrains policies that ignore private information and can be compared with the exploitability of policies extracted from a trained model under the declared interface.

E Training and Evaluation Details

This appendix specifies the training comparisons, numerical settings, game instances, policy-extraction interface, and tabular solver baselines.

E.1 Protocols for the matched comparisons

The opponent comparison uses Qwen3-4B on river seat 0 with seeds 2026–2028, fresh rank-16 LoRA over all modules, 100 rounds, 25 inner passes per round, learning rate 10^{-3}, temperature 0.25, and a 0.02 uniform component in each requested target. Both arms use the mirror-descent update. The self-copy arm re-extracts the opposite-seat policy each round. Exact best responses evaluate both arms.

The update comparison uses Leduc seat 0 and the same seeds. Each arm starts from the terminal retained checkpoint of that seed's preceding search, with a fresh optimizer and regret state. All use rank-16 LoRA over all modules, 100 rounds, five inner passes, learning rate 10^{-4}, current-policy reach weights, and no uniform target component. Temperature is 0.25 for updates that use it. Acceptance checks are disabled in both the opponent and update comparisons.

Training and evaluation use the same native renderer. Candidates are retained at rounds 25, 50, 75, and 100, with selection ties resolved toward the earlier round. An inner pass accumulates gradients over all information sets before one optimizer step. Thus the opponent and update comparisons receive 2,500 and 500 passes, respectively. Budgets are matched within each comparison. These finite-budget results do not establish convergence.

E.2 Numerical settings

The reporting configuration in Section 5 also applies to strategic training and exploitability evaluation in the capability-constrained extension. Bfloat16 is used for generation of some historical source archives; training and final-window evaluation in the negotiation target comparison; and full capability-benchmark evaluation of both the adapted and unmodified models. Saved negotiation checkpoints are evaluated under the main reporting configuration. The independent distillation-bound audit in Table 16 also includes stored policies from earlier experiments under their recorded numerical configurations. It checks the bound on those policies and is separate from the current training comparisons. Configuration records retain the numerical settings for each source. Sequence-form optimization and rational verification use their own solver and arithmetic settings.

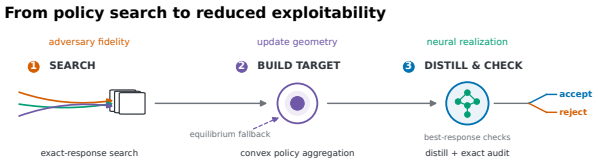

Figure 4: **From policy search to verified LLM behavior.**

E.3 Hardware and software environment

The Qwen3-4B fixed-target distillation runs and additional matched comparisons each use one NVIDIA A100-SXM4 GPU with 80 GB of device memory. Table 17 lists the recorded software environment. Numerical precision follows Appendix E.2; the additional matched comparisons disable TensorFloat-32 (TF32) matrix multiplication.

Table 17: **Hardware and software configuration.** The HiGHS version listed here is from CPU re-evaluation; the GPU-run records identify that backend through the SciPy version.

Component	Configuration
GPU per training run	1 NVIDIA A100-SXM4, 80 GB
Python	3.12.13
CUDA runtime	12.6
PyTorch	`2.12.1+cu126`
Transformers	5.13.0
PEFT	0.19.1
NumPy	2.3.5
SciPy	1.18.0
HiGHS (CPU re-evaluation)	1.12.0

Model and tokenizer. Training uses `Qwen/Qwen3-4B`. Both the model weights and tokenizer are loaded from the same pinned revision, `1cfa9a7208912126459214e8b04321603b3df60c`. Saved-model evaluation uses that same base-model and tokenizer snapshot. The external-policy audit uses the separate systems and extraction interfaces described in Appendix B.1.

Optimization and verification. Sequence-form best responses and archive-mixture linear programs use 64-bit floating-point arrays on the CPU with SciPy's `linprog` and its HiGHS backend. Best-response solves retain the solver defaults. Archive-mixture solves set primal and dual feasibility tolerances to 10^{-9} and the interior-point optimality tolerance to 10^{-10}. For the complete poker cohort, the archive constraint-generation procedure uses an optimality-gap tolerance of 2×10^{-6} in units of $|v|$. The rational verifier checks feasibility and payoff witnesses using Python's exact `Fraction` arithmetic, independently of floating-point LP termination tolerances.

E.4 Method and evaluation map

Tables 18 and 19 give the complete mapping from measured error sources to method choices and evaluation runs.

Table 18: Method choices and their roles. RM$^+$ supplies a policy-search process; the remaining choices turn its model-expressed policies into one reproducible neural policy.

problem	method choice	operation	resulting property
adversary fidelity	exact best response	expose the current policy's largest loss	training and stable sub-value outcomes use the complete opponent set
update geometry	RM$^+$ policy archive	retain every extracted policy rather than only the final target	later target construction can reuse complementary model-expressed behavior
target quality	convex policy aggregation; exact-Nash fallback	minimize exact exploitability over the archive hull; use Nash when the hull is insufficient	the selected target is fixed and strictly sub-value
neural realization	nearest retained initializer; fixed-target distillation	minimize the initial policy change, then fit one stationary target	excess distillation loss bounds added worst-case strategic loss
saved-policy evaluation	exact candidate checks; matched checkpoint reload	reject updates above the activated exploitability ceiling and restore the preceding state	the reported exploitability belongs to the saved policy

Table 19: The evaluation separates consequence, mechanism, design choices, and complete-method performance. Comparisons hold the stated settings fixed within each pair; different studies are not pooled. The fixed-target cohorts refine retained search checkpoints.

question	population	comparison	primary measurement
evaluation consequence	eight canonical-Kuhn systems; both seats	common fixed pool, exact Nash, and policy-specific exact best response	payoff relative to game value
opponent feedback	six river seat-0 runs; three seeds	exact response versus self-copy under mirror descent	selected/final exploitability and trajectories
update and target	15 Leduc seat-0 runs; three seeds	replacement, mirror descent, moving RM^+, frozen first RM^+ and fixed archive	selected/final/P90 exploitability and full-window stability
target source	12 game–seat–seed settings; 24 trained policies	mixture-only, Nash-only and the derived combined rule	stable sub-value outcomes
aggregation and initialization	13 mixture–best-policy pairs; nine initializer pairs	target aggregation; nearest versus terminal checkpoint	paired exploitability and stable outcomes
complete-method performance	18 poker runs; six negotiation runs per target	full method across games, seats, and three seeds	selected and final exploitability, final-window 90th percentile, and stable sub-value outcomes
functional preservation	one selected Kuhn adapter; 25,403 examples	unmodified model and capability-constrained adapter under one evaluation stack	security exploitability plus full capability benchmarks

E.5 GAME SPECIFICATIONS

Table 20 gives the exact sequence-form dimensions. Poker counts are per seat; negotiation lists the two asymmetric seats separately. The river endgame uses board A♠ K♠ Q♢ J♣ 9♡, effective stack 10, a three-raise cap, and fold, check-call, pot-sized bet-raise, and all-in actions. Each seat's range contains the 45 two-card combinations from the ten highest non-board cards; chance is uniform over all disjoint four-card deals. Hand strength is evaluated with Drevo & Hendley (2023).

Table 20: **Exact game instances.** Each dimension is information sets/sequences/flow rows.

game	$\lvert v\rvert$	dimensions	seats	nonzero payoff matrix entries
Kuhn	$1/18$	6/13/7	symmetric	30
Leduc	.0856064	144/337/145	symmetric	966
river endgame	.0428303	270/721/271	symmetric	25,272
negotiation	$4/45$	30/64/31; 9/46/10	asymmetric	171

Negotiation game. Two players divide a fixed \$10 budget. At the start, each player independently receives a strength drawn uniformly from $\{\text{low}, \text{medium}, \text{high}\}$. Each player observes their own strength and the public offer history, but not the other player's strength.

Player 0 makes an opening offer. Player 1 may accept, walk away, or make one counteroffer. Following a counteroffer, Player 0 must accept or walk away; no further offer is permitted. Every offer specifies a share $s \in \{5, 7, 8\}$ for Player 0, with Player 1 receiving $10 - s$, regardless of which player makes the offer.

Acceptance implements the proposed allocation. Walking away instead triggers an allocation based on the private strengths: the stronger player receives \$9 and the weaker player \$1; equal strengths give Player 0 \$4 and Player 1 \$6. The player who walks away then transfers \$1 of their allocation to the other player. This transfer preserves the total budget and makes withdrawal costly. Table 21 lists the resulting allocations.

Table 21: **Walking away changes the allocation but preserves the budget.** Each entry gives (Player 0's share, Player 1's share), including the \$1 transfer from the player who leaves.

Strength comparison	Player 0 walks	Player 1 walks
Player 0 stronger	(\$8, \$2)	(\$10, \$0)
Equal strengths	(\$3, \$7)	(\$5, \$5)
Player 1 stronger	(\$0, \$10)	(\$2, \$8)

For every terminal allocation, let s_0 be Player 0's dollar share. Utilities are $u_0 = s_0 - 5$ and $u_1 = -u_0$, making the game zero-sum. The equilibrium value for Player 0 is $4/45$. Players 0 and 1 have 30 and 9 information sets and 64 and 46 sequences, respectively. The model receives its private strength, public history, and available actions at each decision. The type-blind floors are $8.5 \times |v|$ and $4.0 \times |v|$ for Players 0 and 1; the rational verifier encloses these values.

Negotiation training protocol. We compare archive-mixture and equilibrium targets for Qwen3-4B in both seats with seeds 2030–2032, giving six runs per target. For each seat and seed, both targets use the same saved initializer and acceptance rule. Each run uses rank-16 LoRA over all modules, 100 distillation rounds, five inner passes per round, learning rate 10^{-4}, and no uniform target component. Candidate checkpoints are evaluated at rounds 25, 50, 75, and 100. Numerical settings are given in Appendix E.2.

E.6 Policy extraction contract

The saved-policy extractor uses one deterministic system prompt, user prompt, and ordered legal-action map for every information set. Extraction applies the model's declared chat template, tokenizes each complete legal action string without additional special tokens, and sums its next-token log probabilities over the full continuation. No end-of-sequence probability is added. A softmax over only these complete legal strings, in game-data order, produces the behavioral policy sent to the sequence-form oracle. The renderer validator checks prompt determinism and injectivity, exact legal-action coverage, one-to-one action strings, and inverse parsing before model loading. The declared game-definition file, renderer text, tokenizer revision, action map, and numerical configuration bind the reported exploitability to the evaluated policy.

E.7 Scope of policy evaluation

Each measurement concerns the policy extracted under the declared game and model interface. Changing the prompt renderer, legal-action strings, or numerical configuration can change that policy, so the reported guarantee does not automatically transfer to a different interface. Probabilities are normalized over the declared legal actions; the guarantee therefore concerns this constrained-action policy.

The reported selected checkpoint is chosen from the prescribed saved grid. An unsaved trajectory minimum is not substituted for that checkpoint. The same distinction applies when interpreting final-window measurements and the additional verification of the recorded action probabilities.

E.8 Update-map controls

The direct replacement target $\mathrm{softmax}(Q/\eta)$, with temperature $\eta > 0$, implements a regularized best response. Its fixed points are η-quantal-response equilibria rather than Nash equilibria, and the measured runs cycle or diverge. The proximal target

$$t(a \mid I) \propto \pi(a \mid I)^{1-\lambda} \mu(a \mid I)^{\lambda} \exp((Q(I,a) - V(I))/\eta)$$

is a Hedge, or multiplicative-weights-update (MWU), step. Here $\lambda \in [0, 1]$ controls interpolation toward the optional reference policy μ. It supplies a useful matched comparison but retains a fixed-step exploitability floor. A uniform target mixture restores support for sharp policies in both maps.

E.9 Standard regret variants under exact-response feedback

Common CFR accelerations are not uniformly faster when each opponent is the current exact best response. That response can switch between pure plans after a small learner change, which removes the low-variation feedback used by predictive methods. Table 22 compares eight matched exact tabular trajectories on river seat 1. Signed regret matching crosses below one game value first. Discounting and prediction remain effective solvers, but their usual ordering does not transfer automatically from self-play CFR to this feedback process.

Table 22: Standard regret variants under switching exact-response feedback. One exact tabular river-seat-1 diagnostic runs each method for 2,000 rounds with the same uniform support mixture. Values use the policy after removing that mixture. Lower is better.

update	feedback	best $\times\lvert v\rvert$	best round	first < 1
signed RM	current best response	0.253	1275	125
RM$^+$	current best response	0.466	1925	475
DCFR (1.5)	current best response	0.458	1850	375
DCFR (2,.5)	current best response	0.426	1525	450
PCFR$^+$	current best response	0.557	1700	575
linear CFR	current best response	0.709	1875	1050
PCFR$^+$	response to running average	0.384	1875	1250
RM$^+$	response to running average	0.555	1825	1050

The tabular comparison isolates the update rule from neural training. The method uses RM$^+$ to collect expressed policies; target construction and fixed-target distillation determine the returned model policy.

F Capability Preservation and Exploitability

This appendix specifies the capability constraints and reports exploitability and general-task performance for the same adapter as separate outcomes.

F.1 Capability-constrained training

The capability path starts from unmodified Qwen3-4B and uses fresh rank-16 attention-only low-rank adaptation (LoRA). For each protected segment, the accept/reject rule projects the realized strategic proposal into a quadratic-program cone defined by five protected functions: chat reverse Kullback–Leibler (KL) divergence, reference likelihood, instruction following, multiple-choice knowledge, and reasoning. Chat KL is limited to 0.05; each protected loss is limited to 0.02 above the unmodified-model value and 10^{-4} above the preceding accepted policy. The rule tests the actual interpolated candidates, accepts the largest coefficient that passes all functional constraints and exact sub-value security, and restores the preceding adapter after rejection. The minimum tested coefficient is $1/1024$.

The Kuhn continuation accepts 20 distinct nonzero updates. Every accepted policy is strictly sub-value and passes all five protected decisions; the terminal adapter is $0.258522 \times |v|$, and the maximum over the 20-update tail is $0.999968 \times |v|$. The full evaluation retains 59 sample files and 25,403 unique examples. The same terminal adapter scores MMLU 0.683592, normalized HellaSwag 0.686118, and GSM8K flexible exact match 0.850644, compared with unmodified-model scores 0.683, 0.685, and 0.850 under the same stack. The benchmark suite is run once after the strategic terminal rule selects the adapter, so it does not participate in checkpoint selection.

The matched negotiation continuation accepts five protected updates, each inside the safe set, and reaches $0.999824 \times |v|$. It then abstains: even the minimum tested coefficient produces $1.001086 \times$

$|v|$, although every capability decision still passes. This localizes the boundary to finite safe-set controllability, and it produces no negotiation capability result.

G Best-Iterate Guarantees and the Regret Decomposition

This appendix bounds the best played policy, states when checkpoint saving realizes that selector, and separates classical regret from distillation error. Throughout, our seat is the maximizer (seat 0; the other seat is symmetric); we refine the sets $\mathcal{X}$ and $\mathcal{Y}$ of Section 3 to the two sequence-form polytopes (von Stengel, 1996; Koller et al., 1996), and keep the requested plan m_t and expressed plan x_t of that section. The adversary is the exact best response, $y_t \in \arg\min_{y\in\mathcal{Y}} x_t^\top A\, y$. The search emits a raw behavioral target z_{t+1}; its support mixture is $m_{t+1} = (1-\alpha_t)z_{t+1} + \alpha_t u$ in behavioral-policy space and is distilled into x_{t+1}, where u is the uniform behavioral policy. In the algebra below, z_t and m_t denote the induced realization plans; thus m_t is aligned with x_t. Write $e_t = \mathrm{Expl}(x_t) = v^* - x_t^\top A\, y_t \geq 0$ for the round-t exploitability. All horizon averages and iterate minima below use $T \geq 1$.

Proposition 8 (min-iterate bound)**.** *For $T \geq 1$ and any sequence $x_1, \ldots, x_T \in \mathcal{X}$ trained against exact best responses y_t, let $R_T^x = \max_{x\in\mathcal{X}} \sum_t x^\top A\, y_t - \sum_t x_t^\top A\, y_t$ be the external regret of the played sequence against the adversary's sequence. Then*

$$\frac{1}{T}\sum_{t=1}^{T} e_t \;\leq\; \frac{R_T^x}{T}, \qquad \textit{hence} \qquad \min_{t\leq T} \mathrm{Expl}(x_t) \;\leq\; \frac{R_T^x}{T}.$$

Proof. Finite-game sequence-form minimax and compactness provide a Nash plan x^* for our seat; it guarantees $x^{*\top} A\, y \geq v^*$ for every opponent plan y. Then $R_T^x \geq \sum_t x^{*\top} A\, y_t - \sum_t x_t^\top A\, y_t \geq Tv^* - \sum_t (v^* - e_t) = \sum_t e_t$. The min is at most the mean. □

Because exact best responses evaluate every round, no-regret analysis bounds the mean played-iterate exploitability rather than an average policy, and saving the best checkpoint realizes the minimum selector; periodic saving restricts it to the checkpoint grid. Regret matching and regret matching$^+$ control their emitted sequence through Blackwell approachability and counterfactual-regret decomposition (Zinkevich et al., 2007; Tammelin, 2014; Farina et al., 2020). The expressed and requested sequences can differ, so R_T^x is computed from the aligned log of both. The decoupled exact-best-response form is counterfactual regret minimization with best responses (CFR-BR) (Johanson et al., 2012). A fixed-step mirror-descent or multiplicative-weights update admits a sequence-agnostic guarantee: with at most K actions and counterfactual-value range bounded by G, its average-regret bound has the form $O((\log K)/(\eta T) + \eta G^2)$, where η is the step size. A constant step size therefore leaves a non-vanishing $O(\eta)$ term, producing a persistent floor while last iterates can oscillate (Lockhart et al., 2019; Hennes et al., 2019; Mertikopoulos et al., 2017). Diminishing step sizes restore vanishing tabular regret; near the current policy, vanishing distillation gradients reduce the realized update (§3).

Proof of Corollary 1. Set $\bar{\gamma}_T = 0$ in Theorem 1. For every $T \geq 1$,

$$0 \leq \min_{1\leq t\leq T} \mathrm{Expl}(x_t) \leq R_T^m/T + u_A\bar{\varepsilon}_T \leq C/\sqrt{T} + u_A\bar{\varepsilon}_T.$$

The right-hand side converges to zero because u_A and C are fixed and $\bar{\varepsilon}_T \to 0$. The squeeze theorem gives the claim.

Proposition 9 (Regret decomposition)**.** *Let R_T^z, R_T^m, R_T^x denote external regret for the aligned raw-target, support-mixed-target, and played sequences, respectively. Then*

$$R_T^x = R_T^z + \sum_t (z_t - m_t)^\top A y_t + \sum_t (m_t - x_t)^\top A y_t,$$

and $T^{-1}\sum_t e_t \leq R_T^x/T$. Equivalently, taking m_t as the declared update target gives the three-factor form in Theorem 1.

Proof. Add and subtract $\sum_t z_t^\top A y_t$ and $\sum_t m_t^\top A y_t$ in the definition of R_T^x; the comparator term is shared. Then apply Proposition 8. □

Here R_T^z measures the raw target sequence, $z_t \to m_t$ the deliberate support-mixture step, and $m_t \to x_t$ the model-distillation error. The logs record m_t and x_t; z_t is recoverable when the declared support mixture is invertible. We measure the realized term in sequence-form payoff space and audit its signed payoff discrepancy directly. For a fixed target, Theorem 2 also supplies an a priori bound from the *excess* reach-weighted cross-entropy. The raw loss alone is not the right quantity because it includes target entropy; moving targets additionally change that entropy and the reach weights from round to round.

Fidelity and exact slack. If the adversary is an approximate best response with gap γ_t, the first step of Proposition 8 reads $e_t \le v^* - x_t^\top A\, y_t + \gamma_t$, and the same telescoping yields Theorem 1 by composing with Proposition 9 and bounding the realization term by Hölder's inequality (ℓ_1/ℓ_∞): $(m_t - x_t)^\top A\, y_t \le \|m_t - x_t\|_1 \|A\, y_t\|_\infty \le u_A\, \varepsilon_t$. Moreover the proof of Proposition 8 discards exactly one term, and that term is measurable. Writing $\bar{y}_T$ for the average adversary plan ($\mathcal{Y}$ is convex) and using $\max_x \sum_t x^\top A\, y_t = T \max_x x^\top A\, \bar{y}_T$,

$$\frac{1}{T}\sum_{t=1}^{T} e_t \;=\; \frac{R_T^x}{T} \;-\; \Big(\max_{x\in\mathcal{X}} x^\top A\, \bar{y}_T - v^*\Big) \;=\; \frac{R_T^x}{T} \;-\; \mathrm{Expl}_{\mathrm{opp}}(\bar{y}_T),$$

an *identity*: the slack is the average response's opponent-seat exploitability. The evaluation log verifies both decompositions and the bound.