REGULARIZED POLICY GRADIENT WITH LEARNED MIXTURES OF GAUSSIANS FOR GAMES WITH CONTINUOUS ACTIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Most successes of superhuman game-playing algorithms are in games with discrete actions, yet in auctions, robotics, sports, or trading, actions are nearly continuous. Prior techniques either rely on expert-designed discretizations or are sample inefficient. We present a scalable policy-gradient algorithm for large sequential games with continuous or mixed discrete and continuous actions. It combines *magnetic mirror descent* with a mixture of Gaussians reparametrization, trained via self-play. We show that it approximates equilibrium in games where gradient descent fails. In sequential games, it outperforms *neural fictitious self-play* and matches or outperforms the final strategies of *policy space response oracles* with 3.5–5.5$\times$ fewer samples. In heads-up no-limit Texas hold'em, it performs on par with Slumbot.

1 INTRODUCTION

Games can model many situations where multiple agents are trying to achieve possibly different goals over a long horizon. Many successes in this domain focus on games with discrete action spaces, like Chess, Go, or Stratego (Silver et al., 2018; Schrittwieser et al., 2020; Perolat et al., 2022; Sokota et al., 2025; Zhang & Sandholm, 2026). However, some domains, like robotics or economics, use continuous or nearly-continuous actions (Todorov et al., 2012; Haarnoja et al., 2018; Vickrey, 1961; Zheng et al., 2022).

In single-player and cooperative multi-player reinforcement learning, many studies have demonstrated the effectiveness of policy-gradient algorithms, even for continuous action problems (Lillicrap et al., 2016; Haarnoja et al., 2018; Yu et al., 2022). However, these works often rely on the fact that, in those environments, an optimal pure (i.e., deterministic) strategy exists. It is well known that adversarial games with hidden information or simultaneous moves lack pure optimal strategies and instead require mixing among several actions to prevent the player from becoming predictable (Von Neumann & Morgenstern, 1947; Blackwell & Girshick, 1979; Roberson, 2006). Moreover, mixing can be beneficial even in fully observable physical systems. Consider a penalty kick in football or a tennis serve; players need to commit to their decision before observing what their opponent does. If players commit to the same movement all the time, they will become highly exploitable.

Only recent developments in regularized policy-gradient algorithms have demonstrated how to converge to mixed strategies using scalable neural policy representations, without resorting to impractical and costly policy averaging (Perolat et al., 2021; Sokota et al., 2023). These algorithms then achieved superhuman performance in very large games, such as Stratego (Perolat et al., 2022; Sokota et al., 2025) and Generals.io (Straka et al., 2026), relying on their discrete action spaces.

The common approach to continuous action spaces is to discretize and then solve the resulting discretized game (Kroer & Sandholm, 2015; Tavakoli et al., 2018; Tang & Agrawal, 2020). However, this disregards the relationships among discrete actions, and the algorithm must learn them from gameplay. Moreover, the solution quality depends on the chosen discretization, which often requires expert knowledge (Kroer & Sandholm, 2015). Alternatively, some approaches are directly applicable to continuous action spaces, but they either cannot sufficiently model mixed equilibria (Schulman et al., 2017), are not easily extended to large sequential games (Hsieh et al., 2019;

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.

Domingo-Enrich et al., 2020; Martin & Sandholm, 2023; 2025), or are sample inefficient because they train an ensemble of strategies (Heinrich & Silver, 2016; Lanctot et al., 2017).

In this work, we introduce Magnetic Mixture Policy Optimization, an algorithm for approximating Nash equilibria in large sequential zero-sum games with continuous actions. It parametrizes the strategy for each decision by a mixture of Gaussians. The training jointly learns the means and standard deviations of several Gaussians, as well as the distribution over them. Recent advancements in regularized policy gradients for games were necessary to enable this method, because the mixture strategy over the fixed Gaussian distributions is itself a matrix game, where standard policy-gradient algorithms may not converge. We show that the regularization can be extended even to training the Gaussian parameters, and that it converges in games, where standard gradient descent does not.

We evaluate our method across several games of varying size and type. Our method is applicable to games that mix discrete and continuous actions, as we show in several Poker games, where we treat bets as continuous actions. We also demonstrate that this method is scalable to large games by applying it in full-size heads-up no-limit Texas hold'em. We show that, using our method in self-play on a single GPU, we trained a strategy that is on par with latest Slumbot, an improved version of the winner of the 2018 annual computer poker competition.

2 Background

In this work we consider two-player zero-sum games represented by a triplet $(\mathcal{A}_1, \mathcal{A}_2, u)$, $\mathcal{A}_i \subseteq \mathbb{R}^{N_i}$ is a compact non-empty action space of the player $i \in \{1, 2\}$ and $u : \mathcal{A}_1 \times \mathcal{A}_2 \to \mathbb{R}$ is a continuous payoff function of player i, that is bounded and Borel measurable. We will use $u_1 = u$ and $u_2 = -u$ In the game both players choose simultaneously $a_i \in \mathcal{A}_i$. Player 1 is trying to select an action such that it maximizes the payoff, whereas Player 2 is minimizing it.

$\mathcal{P}(\mathcal{A}_i)$ is a Borel probability measure in a_i, which we call a mixed strategy. The expected payoff is then

$$U_i(\pi_1, \pi_2) = \mathbb{E}_{\pi_1 \otimes \pi_2}[u_i] = \int_{\mathcal{A}_1} \int_{\mathcal{A}_2} u_i(a, b)\, d\pi_2(a)\, d\pi_1(b), \quad (\pi_1, \pi_2) \in \mathcal{P}(\mathcal{A}_1) \times \mathcal{P}(\mathcal{A}_2)$$

Best response to a strategy π_{-i} is defined as $\pi_i^{BR} = \arg\max_{\pi_i \in \mathcal{P}(\mathcal{A}_i)} U_i(\pi_i, \pi_{-i})$. We use the traditional game-theory notation of $-i$ denotes the other player than i. Since U_i is affine, there exists a pure strategy that is a best response. Nash equilibrium is a strategy pair $\pi^* = (\pi_1^*, \pi_2^*)$, such that it is a saddle point of U_1, where neither can improve by unilaterally deviating from its strategy

$$U_1(\pi_1, \pi_2^*) \leq U_1(\pi_1^*, \pi_2^*) \leq U_1(\pi_1^*, \pi_2) \quad \forall \pi_1 \in \mathcal{P}(\mathcal{A}_1), \pi_2 \in \mathcal{P}(\mathcal{A}_2)$$

We also use NashConv $\mathcal{E}$, sometimes called exploitability, to compare different baselines, and it serves as a distance from Nash equilibrium

$$\mathcal{E}(\pi) = \max_{\pi_1'} U_1(\pi_1', \pi_2) - \min_{\pi_2'} U_1(\pi_1, \pi_2')$$

3 Magnetic Mixture Policy Optimization

Gradient descent-style algorithms in continuous action games often serve to find a pure (deterministic) strategy. As such, most of those algorithms are limited to settings with pure Nash equilibria. A common approach to finding these mixed equilibria is to discretize the action spaces of both players and then compute a mixed strategy in the resulting matrix game. However, this approach will always have an unavoidable worst-case discretization error. Moreover, in an online setting with function approximators, the finer abstraction increases both the number of required samples and the overall model size.

In this work, we focus on a discretization that is optimized during training. Some prior work on *generative adversarial networks* has already used discretization, with K pure actions in the action

space (Hsieh et al., 2019; Domingo-Enrich et al., 2020). Then they optimized the value of those pure actions using gradient descent on the exact payoff gradient and also the distribution over them. However, these techniques were not extended into sequential settings. We propose using a Gaussian distribution instead of pure actions, allowing us to easily estimate the gradient and apply entropy regularization. Moreover, in section 3.2, we show that when we add secondary KL-divergence loss term, it enables convergence even in some games where standard gradient descent fails.

3.1 CATEGORICAL HEAD

First, let us assume that oracle provides K actions a^k, and the task is only to find a distribution over those. We do not impose constraints on the parametrization of the actions, except that each pair of actions has a uniquely defined expected value. This then results in a matrix game, where each player has K actions, and the utility given to the players is the expected value when the action is played.

The standard gradient descent methods are known to cycle around the equilibrium point and, as such, may not converge in the last iterate. Several techniques have been developed that converge to the Nash equilibrium in the last iterate (Korpelevich, 1976; Popov, 1980; Daskalakis et al., 2018). In this work, we focus on KL-regularization-based methods, such as *regularized Nash dynamics (RNaD)* and *magnetic mirror descent (MMD)*, as they relate to our use of KL-regularization in Gaussian reparametrization (Perolat et al., 2021; Sokota et al., 2023).

Consider a strategy π_i of player i, which is a categorical distribution over the pure actions a_i^k. With the magnet regularization, the goal of each player is to find a solution to the following optimization problem

$$\pi_i' = \arg\max_{\pi_i} \sum_{k=1}^{K} \pi_i(a_i^k) q_i(a_i^k) - \eta D_{\mathrm{KL}}(\pi_i, \pi_i^M) \tag{1}$$

$q_i(a_i^k)$ is a value player i receives after playing action a_i^k, and π_i^M is a magnet strategy, which can be arbitrary in the game. It has been proven by prior work that this regularized objective has a unique fixed point for fixed magnet regularization η and strategy π_i^M, which is a quantal response equilibrium (Sokota et al., 2023). It has also been shown that periodically updating the π_i^M or annealing η results in convergence to the Nash equilibrium (Perolat et al., 2021).

We are not making any improvements to these algorithms for discrete action games; we merely use them as an effective algorithm to approximate the Nash equilibrium in the matrix game. Other modifications of gradient descent that converge to the Nash equilibrium, such as optimism or extra-gradient (Korpelevich, 1976; Popov, 1980), are also viable. We used the magnet regularization for its compatibility with function approximators in large sequential games, which has been demonstrated in Stratego and Generals.io (Perolat et al., 2022; Sokota et al., 2025; Straka et al., 2026).

3.2 GAUSSIAN HEAD

In single-player settings, it is common to find strategies in continuous action spaces by parametrizing the strategy as a Gaussian distribution defined by its mean μ and a log of standard deviation $\log(\sigma)$ (Haarnoja et al., 2018). During training, it samples from this parametrized Gaussian to estimate the gradient. In addition to gradient estimation, this reparametrization enables entropy regularization, which has been shown to improve exploration in discrete-action settings. Gradient descent is known not to converge in some simple continuous action games, such as $f(x, y) = xy$ (Martin & Sandholm, 2023). We show in experiments that this prevails even when the strategy is parametrized by a Gaussian. In this section, we focus only on strategies parametrized by a single Gaussian and show that adding the magnet KL regularization improves the convergence properties of gradient descent.

Consider a strategy π_i parametrized by the Gaussian mean μ_i and standard deviation σ_i. Throughout the paper, when the mean μ, standard deviation σ, and strategy π share the same subscripts and superscripts, they represent the same strategy. The expected utility of the strategy is then

$$U_i^G(\pi) = \mathbb{E}_{a_1 \sim \mathcal{N}(\mu_1, \sigma_1^2), a_2 \sim \mathcal{N}(\mu_2, \sigma_2^2)} u_i(a_1, a_2)$$

As with discrete games, gradient descent does not converge because the gradient field is not strongly monotonic. Adding a sufficiently strong regularizer to the field then makes the game monotonic, and therefore the gradient descent will converge to a fixed point.

Consider a magnet strategy π^M. We will use $\pi := (\pi_1, \pi_2)$. The gradient dynamics on this reparametrized strategy with the KL regularization are

$$V^R(\pi) = V(\pi) + \eta R(\pi) = \begin{pmatrix} -\nabla_{\pi_1} U_1^G(\pi) \\ \nabla_{\pi_2} U_1^G(\pi) \end{pmatrix} + \eta \begin{pmatrix} \nabla_{\pi_1} D_{\text{KL}}(\pi_1, \pi_1^M) \\ \nabla_{\pi_2} D_{\text{KL}}(\pi_2, \pi_2^M) \end{pmatrix}$$

Assumption 1. *$V(\pi)$ is L_V-Lipschitz continuous on the Gaussian strategy π,*

$$\|V(\pi) - V(\pi')\| \leq L_V \|\pi - \pi'\| \qquad \forall \pi, \pi'$$

Theorem 1. *Let assumption 1 hold and $\eta > \frac{L_V}{\delta_R}$. If $\alpha < \frac{2\delta_M}{L_M^2}$ then the Projected Gradient Descent Ascent (PGDA) converges to unique solution π^* of variational inequalities VI(V^R, Π) and for each strategy π^k in iteration k following holds*

$$\|\pi^k - \pi^*\|^2 \leq r(\eta)^k \|\pi^0 - \pi^*\|^2 \qquad r(\eta) = 1 - 2\alpha\delta_M + \alpha^2 L_M^2 \tag{2}$$

Proof to Theorem 1 follows a similar structure to that of MMD (Sokota et al., 2023). Similarly to that result, the solution of the regularized problem is unique, but this solution is not the Nash equilibrium of the original game. Contrary to the MMD, where the solution is a Quantal Response Equilibrium, the Gaussian reparametrization does not have this guarantee, because the space of feasible strategies is limited by hyperparameter $\sigma_{\min}$, which we use because standard deviation has to be $\sigma > 0$, such that the KL-divergence is defined..

Even if the fixed point is not a Nash equilibrium, several approaches can ensure convergence, such as annealing the regularization or updating the magnet. In the remainder of this work, we use the magnet replacement strategy, which was first applied in *Regularized Nash Dynamics* (Perolat et al., 2021; 2022), where the previous fixed point replaces the magnet. If we assume only a convex-concave games, which have a pure Nash equilibrium (Sion, 1958), then the sequence of the fixed point converges to an ϵ-equilibrium, where ϵ is bounded by the $\sigma_{\min}$, which is a tunable hyperparameter.

Theorem 2. *In a convex-concave game with payoff u that has continuous second derivative. Assume only strategies $\Pi^{\sigma_{\min}} \subseteq \Pi$, which use $\sigma = \sigma_{\min}$ and let $\sigma_{\min} > 0$, $\eta > L_V \sigma_{\min}^2$ and $M_i = \sup |\frac{\partial^2 u}{\partial a_i^2}|$. Let $\pi^{M,t+1} = \Psi(\pi^{M,t})$ be the exact solution for the fixed magnet at iteration t from arbitrary $\pi^{M,0} \in \Pi^{\sigma_{\min}}$. Then the sequence of $\pi^{M,t}$ converges to a fixed solution π^* with exploitability at most*

$$\mathcal{E}(\pi^*) \leq \frac{M_1 + M_2}{2} \sigma_{\min}^2 \tag{3}$$

The proofs of Theorems 1 and 2 are in Appendix A. Proof of Theorem 2 first shows that the fixed point is the magnet itself only when the magnet is a Nash equilibrium. Then, it shows that the fixed point for a specific magnet is closer to the Nash equilibrium than the magnet itself, in terms of geometric distance. However, this does not say that each strategy in the sequence is less exploitable than the previous one. But, it says that the sequence will eventually converge.

3.3 Mixture of Gaussians

The prior section focused on a single Gaussian, but in simultaneous-move or imperfect-information games, the equilibria are often not pure, requiring randomization among several actions. Consider a game where equilibrium mixes between two real values, -1 and 1, uniformly. Using a single Gaussian can never approximate such a strategy. In this section, we propose an algorithm, Magnetic Mixture Policy Optimization (MMPO), that parametrizes the strategy as a Gaussian mixture. It uses the KL divergence between the current and magnet strategies as a secondary loss term. Since the KL divergence between a mixture of Gaussians does not have a closed-form solution, we use an upper bound, which is a sum of the KL divergence between the categorical distribution and the weighted sum of all the Gaussian KL divergences. The final loss is then combined by two PPO-like losses with magnet regularization, one for the categorical distribution and the second for the Gaussian distribution (Schulman et al., 2017; Sokota et al., 2023).

MMPO parametrizes the strategy space of player i by K Gaussian distributions represented by $\mu_{i,k}, \sigma_{i,k}$ and a categorical distribution over them $w_{i,k}$

$$\pi_i = \sum_{k=1}^{K} w_{i,k} \cdot \mathcal{N}(\mu_{i,k}, \sigma_{i,k}^2) \tag{4}$$

$$\sum_{k=1}^{K} w_{i,k} = 1 \qquad w_{i,k} \geq 0, \sigma_{i,k} \geq \sigma_{\min} \quad \forall k \in \{1, \ldots, K\} \tag{5}$$

The expected utility of this strategy profile is then

$$U_i^G(\pi) = \sum_{k=1}^{K} \sum_{l=1}^{K} w_{1,k} w_{2,l} \mathbb{E}_{a_1 \sim \mathcal{N}(\mu_{1,k}, \sigma_{1,k}^2), a_2 \sim \mathcal{N}(\mu_{2,l}, \sigma_{2,l}^2)} u_i(a_1, a_2)$$

We store the strategy π_θ implicitly in the weights of the neural network θ. Consider that the algorithm samples action a^t, from a (w^t, μ^t, σ^t) in state s^t. We will overload the notation $\mathcal{N}^t := \mathcal{N}(\mu^t, \sigma^{2,t})$. The loss is then combined from the loss for the categorical head $\mathcal{L}_w$ and for the Gaussian components $\mathcal{L}_\mathcal{N}$

$$\begin{aligned}
\rho_w &= \min\Big(\frac{\pi_\theta(s^t, w^t)}{\pi_{\text{old}}(s^t, w^t)} A(s^t, a^t), \text{clip}\big(\frac{\pi_\theta(s^t, w^t)}{\pi_{\text{old}}(s^t, w^t)}, 1 - \epsilon_w, 1 + \epsilon_w\big) A(s^t, a^t)\Big) \\
\rho_\mathcal{N} &= \min\Big(\frac{\pi_\theta(a^t | s^t, \mathcal{N}^t)}{\pi_{\text{old}}(a^t | s^t, \mathcal{N}^t)} A(s^t, a^t), \text{clip}\big(\frac{\pi_\theta(a^t | s^t, \mathcal{N}^t)}{\pi_{\text{old}}(a^t | s^t, \mathcal{N}^t)}, 1 - \epsilon_\mathcal{N}, 1 + \epsilon_\mathcal{N}\big) A(s^t, a^t)\Big) \\
\mathcal{L}_w &= -\mathbb{E}_t \rho_w + \eta D_{\text{KL}}(\pi_\theta(s^t, \cdot), \pi^M(s^t, \cdot)) - \alpha_w \mathcal{H}(\pi_\theta(s^t, \cdot)) \\
\mathcal{L}_\mathcal{N} &= -\mathbb{E}_t \rho_\mathcal{N} + \eta D_{\text{KL}}(\mathcal{N}^t, \mathcal{N}^M) - \alpha_\mathcal{N} \mathcal{H}(\mathcal{N}^t) \\
\mathcal{L}_\pi &= \mathcal{L}_w + \mathcal{L}_\mathcal{N}
\end{aligned} \tag{6}$$

both $\mathcal{L}_w$ and $\mathcal{L}_\mathcal{N}$ include the same terms. PPO clipped loss, KL-divergence regularization to a magnet, and entropy loss. We also train a critic, with the generalized advantage estimate (Schulman et al., 2015). The entire training is done in self-play, using only the game simulator.

Increasing the number of Gaussian components K broadens the set of strategies that can be modeled, but it also increases the number of samples required for training. Despite that, one advantage of this model is that it can learn to ignore certain components if they are not useful for the current decision.

MMPO is also applicable to games that combine discrete and continuous actions, such as some continuous Poker games that combine the fold and call actions with continuous betting. The only change to the MMPO is that for the discrete actions, there will be no corresponding Gaussian μ and σ.

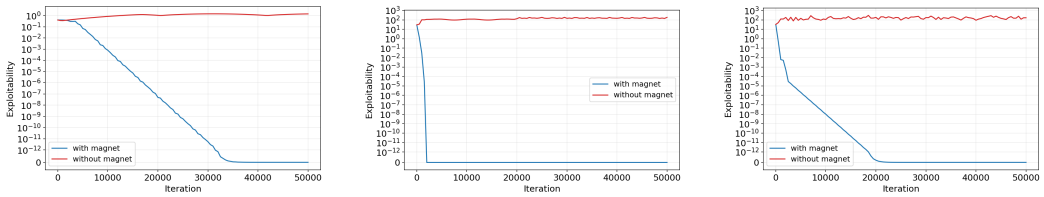
4 Experiments

We have evaluated MMPO across several game settings. We first evaluate the effect of the magnet regularization in isolation. Then we compare our algorithm to several baselines that approximate Nash equilibrium in large games with parametrized strategies, in both one-shot and sequential settings. Lastly, we demonstrate that the approach is applicable even in large games, as we use it in Heads-up no-limit Texas Hold 'em,

4.1 Convergence of the regularized Gaussian head

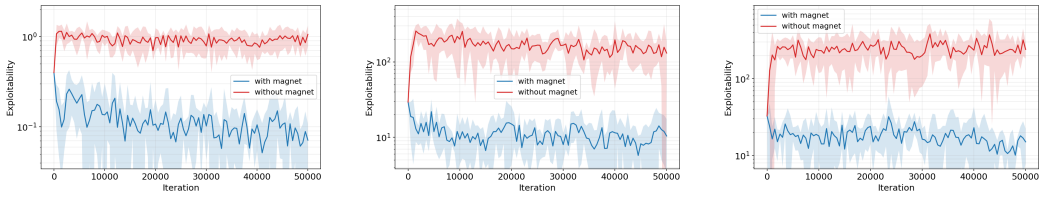
This experiment validates Theorem 1 and explores MMPO's behavior in very simple games. We consider 3 games, with rules in Appendix B, in which there is a pure equilibrium, but gradient descent fails to converge to it. In Figure 1, we show a comparison of runs across the three games, either with usual gradient descent or with regularized projected gradient descent. For the exploitability computation, we used $\sigma = 0$. With the additional regularization, the strategies indeed converge to the Nash equilibrium.

In Figure 2, we show the same comparison, but a neural network now parameterizes the strategy, and the updates are estimated via on-policy sampling. Because of the additional noise from the training, the strategy does not converge to the exact Nash equilibrium, but the regularization improves performance by at least an order of magnitude. In the rotational game, the exploitability remains quite high, but that is due to the payoff function of the game, where the exploitability is high, even when strategies are geometrically very close to the Nash equilibrium. In Section D.1, we show the same graphs when gradient descent is applied to tabular representations of the strategy, with an update using on-policy sampling. The behavior is between the two presented here.



(a) Continuous Matching Pennies (b) Rotational game, dim=2 (c) Rotational game, dim=3

Figure 1: Exploitability in one-shot games with pure Nash equilibria using different gradient descent with or without magnet.



(a) Continuous Matching Pennies (b) Rotational game, dim=2 (c) Rotational game, dim=3

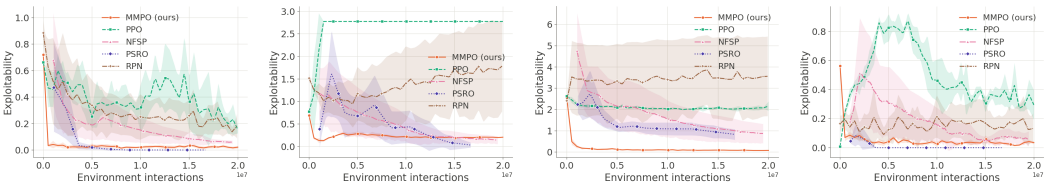
Figure 2: Exploitability in one-shot games with pure Nash equilibria, using PPO with and without magnet regularization

4.2 One-shot games

We compare our methods with several neural baselines on one-shot games. Some baselines we use in this work use vastly different training, so to keep the comparison fair, we compare them in number of environment interactions required. In section D.2, we show the same comparison using the wall-time required. We do not compare with tabular baselines because our work aims to enable efficient training in large games; as such, we do not claim that MMPO should be used in games where tabular techniques are applicable. We believe that in these tractable games, the tabular techniques would outperform our method because they contain less noise and therefore can yield more precise gradient estimates.

The baselines we have used are Proximal Policy Optimization (PPO) (Schulman et al., 2017), Neural Fictitious Self-play (NFSP) (Heinrich & Silver, 2016), Policy Space Response Oracles (PSRO) (Lanctot et al., 2017), and Randomized Policy Networks (RPN) (Martin & Sandholm, 2023). We provide more details about the baselines and their hyperparameter setting in Appendix C.

The results of this experiment are in Figure 3. We have used some games that contain non-finite support equilibria. The finite mixture model cannot approximate an arbitrary strategy. It is limited by the number of Gaussians. PSRO adds a single Gaussian strategy to the mix at each iteration, so it gradually mixes among more strategies than the mixture model. This can be seen in the Glicksberg-Gross game, where PSRO converges closer to the equilibrium. In the Appendix, we present an ablation study on the same games, varying the number of Gaussian components to show this effect in detail.



(a) Glicksberg-Gross game (b) Circle game (c) Two-point game (d) Matching Pennies

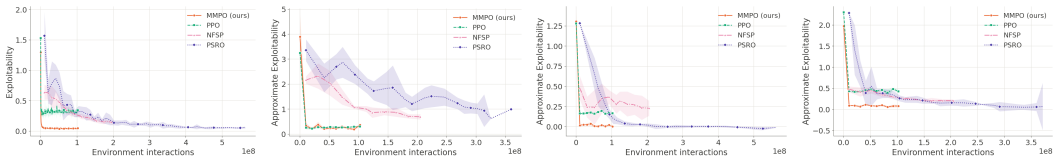
Figure 3: Exploitability with 95% confidence intervals of different baselines in one-shot games

4.3 SEQUENTIAL GAMES

We also evaluated MMPO in sequential games, which is its intended setting. We have used four different games: Kuhn Poker, Leduc Poker, and Sequential Blotto with either 3 or 5 turns. We provide more details on each game in the Appendix. In Figure 4, we show exact exploitability in Kuhn Poker and approximate exploitability computed via PPO for other games, as a function of environment interactions and wall time.

In all games except Leduc, PPO cannot sufficiently model the Nash equilibrium because it uses only a single Gaussian. MMPO converges slightly slower than PPO, but it always converges close to the equilibrium. With the increased training time, PSRO converges closer to the Nash equilibrium than MMPO, which is because PSRO adds a single Gaussian strategy to the mix at every iteration, so eventually it contains more Gaussians than MMPO. However, MMPO converges much quicker to strong strategies both in terms of wall-time and environment interactions, which is crucial, especially in large games.

The approximate exploitabilities of some strategies in Blotto are negative because we compute a best response using PPO, so the resulting value is only a lower bound on the true exploitability.



(a) Kuhn Poker (b) Leduc Poker (c) Blotto 3 (d) Blotto 5

Figure 4: Approximate exploitability with 95% confidence intervals of different baselines in sequential games

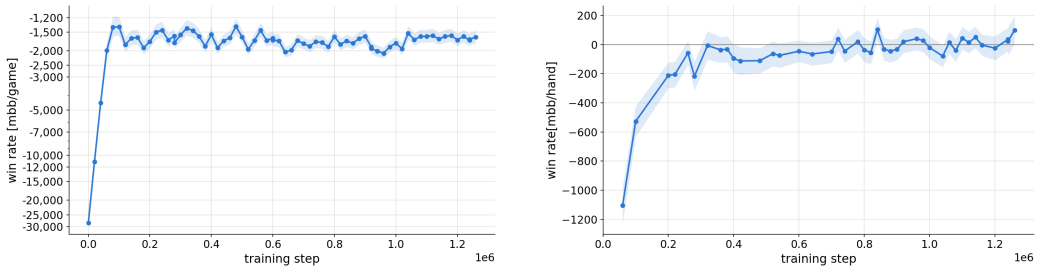
4.4 HEADS-UP NO LIMIT TEXAS HOLD'EM

In the last experiment, we evaluate the scalability of our approach by applying it to Heads-up no-limit Texas Hold'em. An advantage of mixture training is that the algorithm explores parts of the game that would not be visited with static discretization. If a discretized bot is matched against a bot trained with a different discretization, it may encounter out-of-distribution game states. This problem has been observed in Poker bots in the past, and at first, it was solved by interpolating between the closest legal states and later by subgame solving. MMPO avoids both of these as it naturally explores different bets during the training.

Recently, subgame solving has been combined with policy-gradient algorithms (Kubicek et al., 2026). It is possible to combine our training with this subgame-solving approach. We decided not to apply it here because our goal is not to create the strongest Poker bot, but to evaluate the policy gradient approach with our reparametrization.

We trained the strategy network for 4 days and during training we used two evaluation metrics, the first is local best response Lisý & Bowling (2017), which is a Poker-specific evaluation tool that is commonly used as a lower bound for exploitability in Texas Hold'em, the second is head-to-head performance against Slumbot, winner of the 2018 annual computer poker competition (Jackson, 2013), which is the strongest available bot which does not use any subgame solving. Both are shown in Figure 5. We have also used the last checkpoint for more matches against Slumbot, and after playing 900000 hands, the average win-rate was 13.5 ± 36.7 mbb/hand. Based on the information provided by the Slumbot, MMPO scored 28.2 mbb/hand better than Slumbot would score against itself with the same cards. Local best response value of the final checkpoint is -1622.7 ± 158.9.

Based on these results, our approach, trained via self-play for few days with minimal domain knowledge, can stay on par with a bot developed over five years specifically for the Annual Computer Poker Competition. We use no subgame solving, card abstraction, or fixed bet abstraction; the only poker-specific choices are expressing bets as a fraction of the pot, a common practice in Poker AI, and initializing the Gaussian means to cover the interval $[\frac{1}{3}, 3]$ of the pot, from which they are free to move during training (Appendix E).



(a) Local best response

(b) Slumbot

Figure 5: Head-to-head performance with 95% confidence intervals against different methods during the training.

5 Related Work

Counterfactual regret minimization (CFR) is a tabular algorithm, which minimizes counterfactual regret at each decision point for both players (Zinkevich et al., 2007; Tammelin et al., 2015). It has had tremendous success in Texas Hold'em, especially in its application with subgame solving (Burch et al., 2014; Brown & Sandholm, 2018; Moravčík et al., 2017). The subgame-solving version of CFR was extended to function-approximation settings with Deep CFR, and its modern extensions, including ReBeL or Dream (Brown et al., 2019; 2020; Steinberger et al., 2020). However, these extensions rely on discrete action space, and to the best of our knowledge, scalable counterfactual regret minimization for continuous action spaces has not yet been developed.

Fictitious Self-play and *Double oracle* are tabular algorithms in which each player maintains an ensemble of strategies and then mixes among them (Brown, 1951; Heinrich et al., 2015; McMahan et al., 2003; Bošanský et al., 2013). The new strategies are computed as a best response to the current ensemble. These algorithms are easily combined with deep learning, because the best response can be trained via reinforcement learning (Heinrich & Silver, 2016; Lanctot et al., 2017). Also, they can be easily used for games with continuous action spaces by modifying the algorithm to find the best response. AlphaStar has used a League system, in which different types of agents were used in the ensemble, with some trying to exploit specific weaknesses of other strategies (Vinyals et al., 2019). However, these algorithms are often sample-inefficient because training the best response is the bottleneck, and they typically require hundreds of best responses.

Policy-gradient algorithms have been central to the successes in single-player reinforcement learning. Stochastic value gradients (SVG) parametrize the strategy as a Gaussian and then train its mean and standard deviation directly from observed traces (Heess et al., 2015). This same reparametrization trick was later used in proximal policy optimization (PPO) or soft actor critic (SAC) (Schulman et al., 2017; Haarnoja et al., 2018). Prior work has already introduced a mixture of Gaussians as a reparametrization of the strategy, but those focused on increasing the variability of trained strategies and did not take adversarial opponents into account (Baram et al., 2021; Ren et al., 2021; Islam & Huber, 2025). Multi-agent PPO (MAPPO) has extended PPO into a multi-agent setting, but it primarily focused on cooperative settings (Yu et al., 2022). Some extensions have been applied in adversarial settings, but they use only a single Gaussian to parametrize the strategy (Li et al., 2019; Rahman & Xue, 2022). PPO and MAPPO have achieved surprisingly strong performance in some zero-sum games, exemplified by OpenAI Five (OpenAI et al., 2019; Rudolph et al., 2026) or AlphaHoldem (Zhao et al., 2022). Still, they do not provide any convergence guarantees and may converge to highly exploitable strategies when used in self-play (Sokota et al., 2023). Magnetic mirror descent has shown that in games with discrete actions, using the KL-regularization with respect to a magnet strategy guarantees convergence to a Quantal Response Equilibrium (Sokota et al., 2023). Regularized Nash dynamics have shown that, by decreasing the regularization strength or by moving the magnet, these algorithms converge to a Nash equilibrium (Perolat et al., 2021; Sokota et al., 2023; Kalogiannis & Farina, 2026). Our work falls into this category of algorithms, as MMPO parametrizes the strategy as a mixture of Gaussian and categorical distributions, with both components trained via a KL-regularized policy-gradient algorithm.

Action abstraction shrinks down the action space of the game, by either removing some actions or by discretizing the action space. A naive approach to discretization is to use a uniform grid for the actions. This approach is often quite effective, but it scales poorly in extensive-form games because errors can propagate. However, it has been shown that if the game utilities are Lipschitz, then the error of this uniform abstraction can be bounded (Kroer & Sandholm, 2015). Abstractions were a major part of Poker solvers, reducing the size of the game tree and making standard approaches applicable (Jackson, 2013; Brown & Sandholm, 2018; Moravčík et al., 2017). Several techniques have been developed to dynamically adjust the abstraction in Poker bets, but they require the game itself to be tractable for running CFR (Hawkin et al., 2012; Brown & Sandholm, 2014). A different perspective on double oracle algorithms is that the pure best responses can be seen as a discretization (Adam et al., 2021). Our approach can be seen as finding an action abstraction while also solving the game using this abstraction. And since MMPO does not explicitly construct the game, it is not limited by the size of the (abstracted) game tree.

6 Conclusion

We have extended the regularized policy-gradient techniques from discrete-action games to continuous-action games with algorithm Magnetic Mixture Policy Optimization (MMPO). Many real-world adversarial situations in economics, robotics, or sports are continuous, and as such, it is important to develop scalable and efficient algorithms for these settings. MMPO parametrizes the strategy as a mixture of Gaussians, enabling it to be used even in games that combine discrete and continuous actions. We show that even with a single Gaussian, KL regularization enables convergence in games where plain gradient descent diverges. We compare our method across several domains and baselines, demonstrating that it is both efficient and robust. We also scaled MMPO to Heads-up No Limit Texas Hold'em, where we show that training in self-play without any subgame-solving and with minimal domain-specific knowledge manages to stay on par with a strong Poker bot.

AI USE STATEMENT

In this work, we used these AI tools for these tasks:

- Experiments: LLM-assisted programming, scheduling experiments, managing the results of the experiments, such as plotting and summarization. Models: Claude Opus 4.8, Claude Sonnet 5, Claude Opus 5, Claude Fable 5 through Claude Code. GPT5.6-Sol, GPT5.6-Terra, GPT6-Astra through OpenAI Codex. Composer 2.5 and Cursor Grok 4.6 through Cursor
- Help with proofs: Fleshing out details and formalizing the proofs in the paper. Models: Claude Opus 4.8 and Claude Opus 5 through Claude Code.
- Exploration of the related literature. Models: Gemini 3.6 Flash and Gemini 3.6 Thinking. Claude Opus 4.8 and Claude Opus 5 through Claude Code.
- Writing: Improving formulation and overall flow of the paper. Models: Gemini 3.6 Flash, Claude Opus 5 through Claude Code.

We have not used AI tools to generate the main ideas in the paper, draft the sections of the paper, produce data for experiments, or generate citations. We have reviewed all AI-assisted work by testing the generated implementation and verifying the generated parts of the proofs. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REFERENCES

Lukáš Adam, Rostislav Horčík, Tomáš Kasl, and Tomáš Kroupa. Double oracle algorithm for computing equilibria in continuous games. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 5070–5077, 2021.

Nir Baram, Guy Tennenholtz, and Shie Mannor. Maximum entropy reinforcement learning with mixture policies. *arXiv preprint arXiv:2103.10176*, 2021.

David A Blackwell and Meyer A Girshick. *Theory of games and statistical decisions*. Courier Corporation, 1979.

Branislav Bošanský, Christopher Kiekintveld, Viliam Lisý, Jiři Čermák, and Michal Pěchouček. Double-oracle algorithm for computing an exact nash equilibrium in zero-sum extensive-form games. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, pp. 335–342, 2013.

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Yash Katariya, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018. URL `http://github.com/jax-ml/jax`.

George W Brown. Iterative solution of games by fictitious play. *Act. Anal. Prod Allocation*, 13(1): 374, 1951.

Noam Brown and Tuomas Sandholm. Regret transfer and parameter optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 28, 2014.

Noam Brown and Tuomas Sandholm. Superhuman ai for heads-up no-limit poker: Libratus beats top professionals. *Science*, 359(6374):418–424, 2018. doi: 10.1126/science.aao1733. URL `https://www.science.org/doi/abs/10.1126/science.aao1733`.

Noam Brown, Adam Lerer, Sam Gross, and Tuomas Sandholm. Deep counterfactual regret minimization. In *International conference on machine learning*, pp. 793–802. PMLR, 2019.

Noam Brown, Anton Bakhtin, Adam Lerer, and Qucheng Gong. Combining deep reinforcement learning and search for imperfect-information games. *Advances in neural information processing systems*, 33:17057–17069, 2020.

Neil Burch, Michael Johanson, and Michael Bowling. Solving imperfect information games using decomposition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 28, 2014.

Constantinos Daskalakis, Andrew Ilyas, Vasilis Syrgkanis, and Haoyang Zeng. Training GANs with optimism. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=SJJySbbAZ.

Carles Domingo-Enrich, Samy Jelassi, Arthur Mensch, Grant Rotskoff, and Joan Bruna. A mean-field analysis of two-player zero-sum games. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pp. 1861–1870. Pmlr, 2018.

John Hawkin, Robert Holte, and Duane Szafron. Using sliding windows to generate action abstractions in extensive-form games. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 26, pp. 1924–1930, 2012.

Nicolas Heess, Gregory Wayne, David Silver, Timothy Lillicrap, Tom Erez, and Yuval Tassa. Learning continuous control policies by stochastic value gradients. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper_files/paper/2015/file/148510031349642de5ca0c544f31b2ef-Paper.pdf.

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

Johannes Heinrich, Marc Lanctot, and David Silver. Fictitious self-play in extensive-form games. In Francis Bach and David Blei (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 805–813, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/heinrich15.html.

Daniel Hennes, Dustin Morrill, Shayegan Omidshafiei, Rémi Munos, Julien Perolat, Marc Lanctot, Audrunas Gruslys, Jean-Baptiste Lespiau, Paavo Parmas, Edgar Duéñez-Guzmán, et al. Neural replicator dynamics: Multiagent learning via hedging policy gradients. In *Proceedings of the 19th international conference on autonomous agents and multiagent systems*, pp. 492–501, 2020.

Ya-Ping Hsieh, Chen Liu, and Volkan Cevher. Finding mixed Nash equilibria of generative adversarial networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2810–2819. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/hsieh19b.html.

S M Mazharul Islam and Manfred Huber. Categorical policies: Multimodal policy learning and exploration in continuous control. In *2025 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 5236–5241, 2025. doi: 10.1109/SMC58881.2025.11342495.

Eric Griffin Jackson. Slumbot nl: Solving large games with counterfactual regret minimization using sampling and distributed processing. In *AAAI Conference on Artificial Intelligence*, 2013. URL https://api.semanticscholar.org/CorpusID:16497469.

Fivos Kalogiannis and Gabriele Farina. Policy gradient methods converge globally in imperfect-information extensive-form games. *Advances in Neural Information Processing Systems*, 38: 29162–29224, 2026.

Galina M Korpelevich. The extragradient method for finding saddle points and other problems. *Matecon*, 12:747–756, 1976.

Christian Kroer and Tuomas Sandholm. Discretization of continuous action spaces in extensive-form games. In *Proceedings of the 2015 international conference on autonomous agents and multiagent systems*, pp. 47–56, 2015.

Ondrej Kubicek, Viliam Lisy, and Tuomas Sandholm. Test-time reinforcement learning in imperfect information games, 2026. URL https://arxiv.org/abs/2608.30635.

Marc Lanctot, Vinicius Zambaldi, Audrunas Gruslys, Angeliki Lazaridou, Karl Tuyls, Julien Pérolat, David Silver, and Thore Graepel. A unified game-theoretic approach to multiagent reinforcement learning. *Advances in neural information processing systems*, 30, 2017.

Shihui Li, Yi Wu, Xinyue Cui, Honghua Dong, Fei Fang, and Stuart Russell. Robust multi-agent reinforcement learning via minimax deep deterministic policy gradient. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 4213–4220, 2019.

Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2016. URL https://arxiv.org/abs/1509.02971.

Viliam Lisý and Michael Bowling. Eqilibrium approximation quality of current no-limit poker bots. In *AAAI Workshops*, 2017.

Carlos Martin and Tuomas Sandholm. Finding mixed-strategy equilibria of continuous-action games without gradients using randomized policy networks. In Edith Elkind (ed.), *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*, pp. 2844–2852. International Joint Conferences on Artificial Intelligence Organization, 8 2023. doi: 10.24963/ijcai.2023/317. URL https://doi.org/10.24963/ijcai.2023/317. Main Track.

Carlos Martin and Tuomas Sandholm. Joint-perturbation simultaneous pseudo-gradient. In James Kwok (ed.), *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI-25*, pp. 3996–4004. International Joint Conferences on Artificial Intelligence Organization, 8 2025. doi: 10.24963/ijcai.2025/445. URL https://doi.org/10.24963/ijcai.2025/445. Main Track.

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-03)*, pp. 536–543, 2003.

Matej Moravčík, Martin Schmid, Neil Burch, Viliam Lisý, Dustin Morrill, Nolan Bard, Trevor Davis, Kevin Waugh, Michael Johanson, and Michael Bowling. Deepstack: Expert-level artificial intelligence in heads-up no-limit poker. *Science*, 356(6337):508–513, 2017. doi: 10.1126/science.aam6960. URL https://www.science.org/doi/abs/10.1126/science.aam6960.

OpenAI, Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Debiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, Rafal Józefowicz, Scott Gray, Catherine Olsson, Jakub Pachocki, Michael Petrov, Henrique P. d. O. Pinto, Jonathan Raiman, Tim Salimans, Jeremy Schlatter, Jonas Schneider, Szymon Sidor, Ilya Sutskever, Jie Tang, Filip Wolski, and Susan Zhang. Dota 2 with large scale deep reinforcement learning, 2019. URL https://arxiv.org/abs/1912.06680.

Julien Perolat, Remi Munos, Jean-Baptiste Lespiau, Shayegan Omidshafiei, Mark Rowland, Pedro Ortega, Neil Burch, Thomas Anthony, David Balduzzi, Bart De Vylder, et al. From poincaré recurrence to convergence in imperfect information games: Finding equilibrium via regularization. In *International Conference on Machine Learning*, pp. 8525–8535. PMLR, 2021.

Julien Perolat, Bart De Vylder, Daniel Hennes, Eugene Tarassov, Florian Strub, Vincent de Boer, Paul Muller, Jerome T. Connor, Neil Burch, Thomas Anthony, Stephen McAleer, Romuald Elie, Sarah H. Cen, Zhe Wang, Audrunas Gruslys, Aleksandra Malysheva, Mina Khan, Sherjil Ozair, Finbarr Timbers, Toby Pohlen, Tom Eccles, Mark Rowland, Marc Lanctot, Jean-Baptiste Lespiau, Bilal Piot, Shayegan Omidshafiei, Edward Lockhart, Laurent Sifre, Nathalie Beauguerlange,

Remi Munos, David Silver, Satinder Singh, Demis Hassabis, and Karl Tuyls. Mastering the game of stratego with model-free multiagent reinforcement learning. *Science*, 378(6623):990–996, 2022. doi: 10.1126/science.add4679. URL https://www.science.org/doi/abs/10.1126/science.add4679.

Leonid Denisovich Popov. A modification of the arrow-hurwicz method for search of saddle points. *Mathematical notes of the Academy of Sciences of the USSR*, 28(5):845–848, 1980.

Marc-Antoine Provost, Nejc Ilenic, Christopher Solinas, and Philippe Beardsell. Gto wizard benchmark, 2026. URL https://arxiv.org/abs/2603.23660.

Md Masudur Rahman and Yexiang Xue. Robust policy optimization in deep reinforcement learning, 2022. URL https://arxiv.org/abs/2212.07536.

Jie Ren, Yewen Li, Zihan Ding, Wei Pan, and Hao Dong. Probabilistic mixture-of-experts for efficient deep reinforcement learning. *arXiv preprint arXiv:2104.09122*, 2021.

Brian Roberson. The colonel blotto game. *Economic Theory*, 29(1):1–24, 2006.

Max Rudolph, Nathan Lichtle, Sobhan Mohammadpour, Alexandre M Bayen, Zico Kolter, Amy Zhang, Gabriele Farina, Eugene Vinitsky, and Samuel Sokota. Reevaluating policy gradient methods for imperfect-information games. In *International Conference on Learning Representations*, volume 2026, pp. 156908–156943, 2026.

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, et al. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609, 2020.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015.

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

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, Timothy Lillicrap, Karen Simonyan, and Demis Hassabis. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144, 2018. doi: 10.1126/science.aar6404. URL https://www.science.org/doi/abs/10.1126/science.aar6404.

Maurice Sion. On general minimax theorems. *Pacific Journal of Mathematics*, 8(1):171–176, 1958. URL https://projecteuclid.org/journals/pacific-journal-of-mathematics/volume-8/issue-1/On-general-minimax-theorems/pjm/1103040253.full.

Samuel Sokota, Ryan D'Orazio, J Zico Kolter, Nicolas Loizou, Marc Lanctot, Ioannis Mitliagkas, Noam Brown, and Christian Kroer. A unified approach to reinforcement learning, quantal response equilibria, and two-player zero-sum games. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=DpE5UYUQzZH.

Samuel Sokota, Eugene Vinitsky, Hengyuan Hu, J. Zico Kolter, and Gabriele Farina. Superhuman ai for stratego using self-play reinforcement learning and test-time search, 2025. URL https://arxiv.org/abs/2511.07312.

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

Matej Straka, Viliam Lisý, and Martin Schmid. Superhuman AI for generals.io using self-play reinforcement learning. *arXiv preprint arXiv:2606.23348*, 2026.

Oskari Tammelin, Neil Burch, Michael Johanson, and Michael Bowling. Solving heads-up limit texas hold'em. In *IJCAI*, volume 15, pp. 645–652, 2015.

Yunhao Tang and Shipra Agrawal. Discretizing continuous action space for on-policy optimization. In *Proceedings of the aaai conference on artificial intelligence*, volume 34, pp. 5981–5988, 2020.

Arash Tavakoli, Fabio Pardo, and Petar Kormushev. Action branching architectures for deep reinforcement learning. In *Proceedings of the aaai conference on artificial intelligence*, volume 32, 2018.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033. IEEE, 2012. doi: 10.1109/IROS.2012.6386109.

William Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *The Journal of finance*, 16(1):8–37, 1961.

Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *nature*, 575(7782):350–354, 2019.

John Von Neumann and Oskar Morgenstern. Theory of games and economic behavior, 2nd rev. 1947.

Chao Yu, Akash Velu, Eugene Vinitsky, Jiaxuan Gao, Yu Wang, Alexandre Bayen, and Yi Wu. The surprising effectiveness of ppo in cooperative multi-agent games. *Advances in neural information processing systems*, 35:24611–24624, 2022.

Brian Zhang and Tuomas Sandholm. General search techniques without common knowledge for imperfect-information games, and application to superhuman fog of war chess. In *International Conference on Learning Representations*, volume 2026, pp. 93536–93563, 2026.

Enmin Zhao, Renye Yan, Jinqiu Li, Kai Li, and Junliang Xing. Alphaholdem: High-performance artificial intelligence for heads-up no-limit poker via end-to-end reinforcement learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pp. 4689–4697, 2022.

Stephan Zheng, Alexander Trott, Sunil Srinivasa, David C. Parkes, and Richard Socher. The ai economist: Taxation policy design via two-level deep multiagent reinforcement learning. *Science Advances*, 8(18):eabk2607, 2022. doi: 10.1126/sciadv.abk2607. URL `https://www.science.org/doi/abs/10.1126/sciadv.abk2607`.

Martin Zinkevich, Michael Johanson, Michael Bowling, and Carmelo Piccione. Regret minimization in games with incomplete information. *Advances in neural information processing systems*, 20, 2007.

A PROOFS

Fact 1. *1. KL divergence between two univariate Gaussians p, q with corresponding means μ_p, μ_q and standard deviations σ_p, σ_q is*

$$D_{KL}(p,q) = \log(\frac{\sigma_q}{\sigma_p}) + \frac{\sigma_p^2 + (\mu_p - \mu_q)^2}{2\sigma_q^2} - \frac{1}{2}$$

2. First order derivatives of KL divergence between two univariate Gaussians p, q wrt. μ_p and σ_p are

$$\frac{\partial}{\partial \mu_p} D_{KL}(p,q) = \frac{\mu_p - \mu_q}{\sigma_q^2}$$
$$\frac{\partial}{\partial \sigma_p} D_{KL}(p,q) = \frac{\sigma_p}{\sigma_q^2} - \frac{1}{\sigma_p}$$

3. Second order derivatives of KL divergence between two univariate Gaussians p, q wrt. μ_p and σ_p are

$$\frac{\partial^2}{\partial \mu_p^2} D_{KL}(p,q) = \frac{1}{\sigma_q^2}$$
$$\frac{\partial^2}{\partial \sigma_p^2} D_{KL}(p,q) = \frac{1}{\sigma_q^2} + \frac{1}{\sigma_p^2}$$
$$\frac{\partial^2}{\partial \mu_p \partial \sigma_p} D_{KL}(p,q) = 0$$

Lemma 1. *Under assumption 1, for all π, π' the non-monotonicity of V is bounded*

$$\langle V(\pi) - V(\pi'), \pi - \pi' \rangle \geq -L_V \|\pi - \pi'\|^2$$

Proof. The assumption 1 states that the function is L_V-Lipschitz continuous. By using Cauchy-Schwarz inequality $\langle u, v \rangle \leq \|u\|\|v\|$

$$\langle V(\pi) - V(\pi'), \pi - \pi' \rangle \geq -\|V(\pi) - V(\pi')\| \cdot \|\pi - \pi'\| \geq -L_V \|\pi - \pi'\|^2 \tag{7}$$

□

Lemma 2. *Assume a strategy space Π, where $\sigma_1, \sigma_2 \geq \sigma_{min} > 0$. The regularizer $R(\pi)$ is a continuously differentiable, δ_R-strongly monotone and L_R-Lipschitz on π for a magnet strategy $\pi^M \in \Pi$ with*

$$\delta_R = \min(\frac{1}{\sigma_1^{2,M}}, \frac{1}{\sigma_2^{2,M}}) > 0 \tag{8}$$

$$L_R = \frac{1}{\sigma_{min}^2} + \max(\frac{1}{\sigma_1^{2,M}}, \frac{1}{\sigma_2^{2,M}}) < \infty \tag{9}$$

Proof. Using the fact that

$$R(\pi) = \begin{pmatrix} \frac{\partial}{\partial \mu_1} D_{\mathrm{KL}}(\pi, \pi^M) \\ \frac{\partial}{\partial \sigma_1} D_{\mathrm{KL}}(\pi, \pi^M) \\ \frac{\partial}{\partial \mu_2} D_{\mathrm{KL}}(\pi, \pi^M) \\ \frac{\partial}{\partial \sigma_2} D_{\mathrm{KL}}(\pi, \pi^M) \end{pmatrix} = \begin{pmatrix} \frac{\mu_1 - \mu_1^M}{\sigma_1^{2,M}} \\ \frac{\sigma_1}{\sigma_1^{2,M}} - \frac{1}{\sigma_1} \\ \frac{\mu_2 - \mu_2^M}{\sigma_2^{2,M}} \\ \frac{\sigma_2}{\sigma_2^{2,M}} - \frac{1}{\sigma_2} \end{pmatrix} \tag{10}$$

The Jacobian of $R(\pi)$ is

$$J_R(\pi) = \mathrm{diag}\Big(\frac{1}{\sigma_1^{2,M}}, \frac{1}{\sigma_1^2} + \frac{1}{\sigma_1^{2,M}}, \frac{1}{\sigma_2^{2,M}}, \frac{1}{\sigma_2^2} + \frac{1}{\sigma_2^{2,M}}\Big) \tag{11}$$

The Jacobian $J_R(\pi)$ is a diagonal matrix, and each diagonal entry is positive, because $\sigma_1, \sigma_1^M, \sigma_2, \sigma_2^M > 0$.

It holds that $\delta_R \leq \frac{1}{\sigma_i^{2,M}} \leq \frac{1}{\sigma_i^{2,M}} + \frac{1}{\sigma_i^2} \leq L_R$ for $i \in \{1, 2\}$ and all strategies π. Out of this follows $\delta_R \mathbf{I} \preceq J_R(\pi) \preceq L_R \mathbf{I}$, where $\preceq$ is an Loewner ordering $\mathbf{A} \preceq \mathbf{B} \Leftrightarrow x^\top(\mathbf{B} - \mathbf{A})x \geq 0$, for all x. In other words that $(\mathbf{B} - \mathbf{A})$ is positive semi-definite.

Strategy space Π is convex, therefore if $\pi, \pi' \in \Pi$ then $\pi' + t(\pi - \pi'); t \in [0, 1]$ is also in Π. From the fundamental theorem of calculus follows

$$\langle R(\pi) - R(\pi'), \pi - \pi' \rangle = \int_0^1 (\pi - \pi')^\top J_R(\pi' + t(\pi - \pi'))(\pi - \pi') dt \geq \delta_R \|\pi - \pi'\|^2 \tag{12}$$

$$\begin{aligned} \|R(\pi) - R(\pi')\| &= \| \int_0^1 J_R(\pi' + t(\pi - \pi'))(\pi - \pi') dt \| \\ &\leq \int_0^1 \|J_R(\pi' + t(\pi - \pi'))(\pi - \pi')\| dt \\ &\leq \int_0^1 \|J_R(\pi' + t(\pi - \pi'))\|_{\text{op}} \|(\pi - \pi')\| dt \\ &\leq \int_0^1 L_R \|(\pi - \pi')\| dt = L_R \|(\pi - \pi')\| \end{aligned} \tag{13}$$

□

Lemma 3. *Under assumption 1 and $\eta > \frac{L_V}{\delta_R}$ the V^R is δ_M-strongly monotone and L_M-Lipschitz on strategy space Π, where $\sigma_1, \sigma_2 \geq \sigma_{min} > 0$ with*

$$\delta_M = \eta \delta_R - L_V > 0 \tag{14}$$

$$L_M = \eta L_R + L_V < \infty \tag{15}$$

$$L_M \geq \delta_M \tag{16}$$

Proof.

$$\begin{aligned} \langle V^R(\pi) - V^R(\pi'), \pi - \pi' \rangle &= \langle V(\pi) + \eta R(\pi) - V(\pi') - \eta R(\pi'), \pi - \pi' \rangle \\ &= \langle V(\pi) - V(\pi'), \pi - \pi' \rangle + \eta \langle R(\pi) - R(\pi'), \pi - \pi' \rangle \\ &\geq -L_V \|\pi - \pi'\|^2 + \eta \delta_R \|\pi - \pi'\|^2 \\ &= (\eta \delta_R - L_V) \|\pi - \pi'\|^2 = \delta_M \|\pi - \pi'\|^2 \end{aligned} \tag{17}$$

$\delta_M > 0$ based on the assumption $\eta > \frac{L_V}{\delta_R}$.

$$\begin{aligned} \|V^R(\pi) - V^R(\pi')\| &= \|V(\pi) + \eta R(\pi) - V(\pi') - \eta R(\pi')\| \\ &\leq \|V(\pi) - V(\pi')\| + \eta \|R(\pi) - R(\pi')\| \\ &\leq L_V \|\pi - \pi'\| + \eta L_R \|(\pi - \pi')\| \\ &= (L_V + \eta L_R) \|\pi - \pi'\| = L_M \|\pi - \pi'\| \end{aligned} \tag{18}$$

The fact that $L_M > \delta_M$ follows from lemma 2, specifically $L_R > \delta_R$ and from the fact that $L_V > 0$. □

Fact 2. *Let Π be nonempty, closed and convex. The Euclidean projection of any π^+ is defined as*

$$P_\Pi(\pi^+) = \arg\min_{\pi \in \Pi} ||\pi - \pi^+\| \tag{19}$$

Then for any π^+

1. *The $P_\Pi(\pi^+)$ is unique*

2. *Feasible* $\pi^\circ \in \Pi$ *satisfies* $\pi^\circ = P_\Pi(\pi^+)$ *if and only if*

$$\langle \pi^+ - \pi^\circ, \pi - \pi^\circ \rangle \leq 0 \qquad \forall \pi \in \Pi \tag{20}$$

3. P_Π *is non-expansive for any two such* $\pi^+, \hat{\pi}^+$

$$\|P_\Pi(\pi^+) - P_\Pi(\hat{\pi}^+)\| \leq \|\pi^+ - \hat{\pi}^+\| \tag{21}$$

Lemma 4. *Let assumption 1 hold and* $\eta > \frac{L_V}{\delta_R}$. *Let* L_M, δ_M *be the constants from lemma 3 and* $0 < \alpha < \frac{2\delta_M}{L_M^2}$. *Define the Projected Gradient Ascent Descent operator as*

$$T(\pi) = P_\Pi(\pi - \alpha V^R(\pi)) \tag{22}$$

Then T *is a contraction*

$$\|T(\pi) - T(\pi')\|^2 \leq r(\alpha)\|\pi - \pi'\|^2 \tag{23}$$

Proof.

$$\begin{aligned}\|T(\pi) - T(\pi')\|^2 &= \|P_\Pi(\pi - \alpha V^R(\pi)) - P_\Pi(\pi' - \alpha V^R(\pi'))\|^2 \\ &\leq \|\pi - \pi' - \alpha(V^R(\pi) - V^R(\pi'))\|^2 \\ &= \|\pi - \pi'\|^2 - 2\alpha\langle \pi - \pi', V^R(\pi) - V^R(\pi')\rangle + \alpha^2\|V^R(\pi) - V^R(\pi')\|^2\end{aligned} \tag{24}$$

The first inequality follows from the fact that Π is closed and convex, which makes P_Π non-expansive. We will use the δ_M-strong monotonicity and L_M-Lipschitz continuity to get

$$-2\alpha\langle \pi - \pi', V^R(\pi) - V^R(\pi')\rangle \leq -2\alpha\delta_M\|\pi - \pi'\|^2 \tag{25}$$

$$\alpha^2\|(V^R(\pi) - V^R(\pi'))\|^2 \leq \alpha^2 L_M^2\|\pi - \pi'\|^2 \tag{26}$$

This gets the following inequality

$$\|T(\pi) - T(\pi')\|^2 \leq (1 - 2\alpha\delta_M + \alpha^2 L_M^2)\|\pi - \pi'\|^2 \tag{27}$$

Then the PGDA is a contraction when $0 < 1 - 2\alpha\delta_M + \alpha^2 L_M^2 < 1$, which holds for learning rate $0 < \alpha < \frac{2\delta_M}{L_M^2}$. □

Proposition 1. *Under assumption 1 the problem VI*(V^R, Π) *has a unique solution* π^* *for every* $\eta > \frac{L_V}{\delta_R}$ *and* $\pi^M \in \Pi$. *For every* $\alpha > 0$ *the* π^* *is a fixed point*

$$\pi^* = T(\pi^*) \tag{28}$$

If in addition π^* *is in interior of* Π, *then* $V^R(\pi^*) = 0$

Proof. We start by showing that the fixed points π^* of T are solutions of the VI(V^R, Π)

$$\pi^* = T(\pi^*) \Leftrightarrow \langle V^R(\pi^*), \pi - \pi^*\rangle \geq 0 \tag{29}$$

We will use fact 2. The unprojected step is $\pi^+ = \pi^* - \alpha V^R(\pi^*)$ and the feasible point is $\pi^* = \pi^\circ = P_\Pi(\pi^+)$. From fact 2 we know that it also needs to hold

$$\langle \pi^+ - \pi^\circ, \pi - \pi^\circ\rangle = \langle \pi^* - \alpha V^R(\pi^*) - \pi^*, \pi - \pi^\circ\rangle = \langle -\alpha V^R(\pi^*), \pi - \pi^\circ\rangle \leq 0 \tag{30}$$

This means that the $\langle \alpha V^R(\pi^*), \pi - \pi^\circ\rangle \geq 0$.

Now we will show that fixed point exists. First the $0 < \alpha < \frac{2\delta_M}{L_M^2}$ is a non-empty because of lemma 3. Let's fix any α from that feasible set. Based on lemma 4, we know that T is a contraction and the Π is closed and nonempty. Through Banach fixed-point theorem, there is a fixed point for the fixed α. This fixed point is π^*, when using eq. (29) from left to right at that same α it is a solution of the VI(V^R, Π). Then using the $eq.$ (29) from right to left, this same fixed point is a solution for any feasible α, not only the one used to find this point.

The uniqueness of the fixed point comes from the strong monotonicity. □

Theorem 1. *Let assumption 1 hold and $\eta > \frac{L_V}{\delta_R}$. If $\alpha < \frac{2\delta_M}{L_M^2}$ then the Projected Gradient Descent Ascent (PGDA) converges to unique solution π^* of variational inequalities VI(V^R, Π) and for each strategy π^k in iteration k following holds*

$$\|\pi^k - \pi^*\|^2 \leq r(\eta)^k \|\pi^0 - \pi^*\|^2 \qquad r(\eta) = 1 - 2\alpha\delta_M + \alpha^2 L_M^2 \tag{2}$$

Proof. The strategy computed at iteration $k+1$ is

$$\pi^{k+1} = T(\pi^k) \tag{31}$$

By proposition 1, there is a unique solution $\pi^* = T(\pi^*)$ for the same α. Now let us use $\pi' = \pi^*$ in lemma 4

$$\|\pi^{k+1} - \pi^*\|^2 = \|T(\pi^k) - T(\pi^*)\|^2 \leq (1 - 2\alpha\delta_M + \alpha^2 L_M^2)\|\pi^k - \pi^*\|^2 \tag{32}$$

Iterating this procedure over k steps results in

$$\|\pi^k - \pi^*\|^2 \leq r(\eta)^k \|\pi^0 - \pi^*\|^2 \tag{33}$$

□

Proposition 2. *For a convex-concave game, for strategy $\pi^M \in \Pi^{\sigma_{\min}}$ it holds that*

$$\pi^M = \Psi(\pi^M) \Leftrightarrow \langle V(\pi^M), \pi - \pi^M \rangle \geq 0 \quad \forall \pi \in \Pi^{\sigma_{\min}} \tag{34}$$

Proof. Since $\sigma_i = \sigma_i^M$, we know that the $\frac{\partial}{\partial \sigma_i} D_{\mathrm{KL}}(\pi, \pi^M) = 0$, as such it is enough to work with the μ. Since the magnet is a fixed point of Ψ, we know that

$$\begin{aligned} \langle V^R(\pi^M), \pi - \pi^M \rangle &\geq 0 \quad \forall \pi \in \Pi^{\sigma_{\min}} \\ \langle V(\pi^M) + \eta R(\pi^M), \pi - \pi^M \rangle &\geq 0 \quad \forall \pi \in \Pi^{\sigma_{\min}} \\ \langle V(\pi^M), \pi - \pi^M \rangle &\geq 0 \quad \forall \pi \in \Pi^{\sigma_{\min}} \end{aligned} \tag{35}$$

which comes from the fact that $R(\pi^M) = 0$ for fixed point. Conversely, the same approach shows that any solution of $\langle V(\pi^M), \pi - \pi^M \rangle \geq 0$ solves also the VI$(V(\pi^M) + \eta R, \Pi)$ □

Lemma 5. *Let $\pi^M \in \Pi^{\sigma_{\min}}$ and $\pi^+ = \Psi(\pi^M)$ it holds that*

$$\langle V(\pi^+), \pi^+ - \pi \rangle \leq \frac{\eta}{2\sigma_{\min}^2} \left(\|\mu - \mu^M\|^2 - \|\mu - \mu^+\|^2 - \|\mu^+ - \mu^M\|^2 \right) \tag{36}$$

Moreover, for π^* holds

$$\|\mu^* - \mu^+\|^2 \leq \|\mu^* - \mu^M\|^2 - \|\mu^+ - \mu^M\|^2 \tag{37}$$

Proof. We know that π^+ is a fixed point of VI$(V^R, \Pi^{\sigma_{\min}})$, so it holds

$$\langle V^R(\pi^+), \pi - \pi^+ \rangle \geq 0 \quad \forall \pi \in \Pi^{\sigma_{\min}} \tag{38}$$

Let us focus only on the μ part of the inequality, because the σ is fixed. We use $V(\mu)$ for only the mean part of the field

$$\begin{aligned} \langle V^R(\mu^+), \mu - \mu^+ \rangle &= \langle V(\mu^+) + \eta R(\mu^+), \mu - \mu^+ \rangle \\ &= \langle V(\mu^+) + \frac{\eta}{\sigma_{\min}^2}(\mu^+ - \mu^M), \mu - \mu^+ \rangle \\ &\geq 0 \end{aligned} \tag{39}$$

We can further reorder this into

$$\langle V(\pi^+), \pi^+ - \pi \rangle = \langle V(\mu^+), \mu^+ - \mu \rangle \leq \frac{\eta}{\sigma_{\min}^2} \langle \mu^+ - \mu^M, \mu - \mu^+ \rangle \tag{40}$$

Now we can now use following identity $2\langle q-s, p-q\rangle = \|p-s\|^2 - \|p-q\|^2 - \|q-s\|^2$ to get

$$\frac{\eta}{\sigma_{\min}^2}\langle \mu^+ - \mu^M, \mu - \mu^+\rangle = \frac{\eta}{2\sigma_{\min}^2}(\|\mu-\mu^M\|^2 - \|\mu-\mu^+\|^2 - \|\mu^+-\mu^M\|^2) \tag{41}$$

We know that for π^* it holds $\langle V(\pi^*), \pi^+ - \pi^*\rangle \geq 0$. Due to the monotonicity of the convex-concave game it also holds $\langle V(\pi) - V(\pi'), \pi - \pi'\rangle \geq 0$. Therefore

$$\begin{aligned} \langle V(\pi^+) - V(\pi^*), \pi^+ - \pi^*\rangle = \langle V(\pi^+), \pi^+ - \pi^*\rangle - \langle V(\pi^*), \pi^+ - \pi^*\rangle \geq 0 \\ \langle V(\pi^+), \pi^+ - \pi^*\rangle \geq \langle V(\pi^*), \pi^+ - \pi^*\rangle \geq 0 \end{aligned} \tag{42}$$

Putting it all together, we get

$$0 \leq \langle V(\pi^+), \pi^+ - \pi^*\rangle \leq \frac{\eta}{2\sigma_{\min}^2}\big(\|\mu^*-\mu^M\|^2 - \|\mu^*-\mu^+\|^2 - \|\mu^+-\mu^M\|^2\big) \tag{43}$$

Since it holds that $\eta \geq 0$ and $\sigma_{\min} \geq 0$, it has to hold that

$$\begin{aligned} \|\mu^*-\mu^M\|^2 - \|\mu^*-\mu^+\|^2 - \|\mu^+-\mu^M\|^2 \geq 0 \\ \|\mu^*-\mu^M\|^2 - \|\mu^+-\mu^M\|^2 \geq \|\mu^*-\mu^+\|^2 \end{aligned} \tag{44}$$

□

Theorem 3. *For a convex-concave game and $\eta > L_V\sigma_{\min}^2$, let $\pi^{M,t+1} = \Psi(\pi^{M,t})$ from arbitrary $\pi^{M,0} \in \Pi^{\sigma_{\min}}$, then it holds that*

$$\sum_{t\geq 0} \|\pi^{M,t+1} - \pi^{M,t}\|^2 \leq D^2 \tag{45}$$

where D is the diameter of the action space. Moreover, $\mu^{M,t} \to \overline{\mu}$, where $\overline{\pi} = (\overline{\mu}, \sigma_{min})$ solves variational inequalities VI$(V, \Pi^{\sigma_{min}})$

Proof. If we use the result from Lemma 5, we know that following holds for π^*

$$\|\mu^* - \mu^{M,t+1}\|^2 \leq \|\mu^* - \mu^{M,t}\|^2 - \|\mu^{M,t+1} - \mu^{M,t}\|^2 \tag{46}$$

This means that $\|\mu^* - \mu^{M,t}\|$ is non-increasing, which gets

$$\begin{aligned} \sum_{t\geq 0} \|\mu^{M,t+1} - \mu^{M,t}\|^2 &\leq \sum_{t\geq 0} \|\mu^* - \mu^{M,t}\|^2 - \|\mu^* - \mu^{M,t+1}\|^2 \\ &= \|\mu^* - \mu^{M,0}\|^2 \\ &\leq D^2 < \infty \end{aligned} \tag{47}$$

The solution set of VI$(V, \Pi^{\sigma_{\min}})$ is nonempty, because $\Pi^{\sigma_{\min}}$ is compact and convex and V is continuous. Lemma 5 holds for every solution of the variation inequality solution π^*, so the sequence $\|\pi^* - \pi^{M,t}\|$ is non-increasing for all solutions π^*. The above bound shows $\|\mu^{M,t+1} - \mu^{M,t}\|^2 \to 0$.

Since $\pi^{M,t+1}$ is a solution of VI$(V^R, \Pi^{\sigma_{\min}})$, it holds

$$\langle V(\mu^{M,t+1}) + \frac{\eta}{\sigma_{\min}^2}(\mu^{M,t+1} - \mu^{M,t}), \mu - \mu^{M,t+1}\rangle \geq 0 \tag{48}$$

Splitting the inner product and applying Cauchy-Schwarz inequality, we get

$$\begin{aligned} \langle V(\mu^{M,t+1}), \mu - \mu^{M,t+1}\rangle &\geq -\frac{\eta}{\sigma_{\min}^2}\langle \mu^{M,t+1} - \mu^{M,t}, \mu - \mu^{M,t+1}\rangle \\ &\geq -\frac{\eta}{\sigma_{\min}^2}\|\mu^{M,t+1} - \mu^{M,t}\|\|\mu - \mu^{M,t+1}\| \\ &\geq -\frac{\eta}{\sigma_{\min}^2}\|\mu^{M,t+1} - \mu^{M,t}\|D \end{aligned} \tag{49}$$

The sequence $\mu^{M,t}$ lies in a compact set, so there is subsequence that starts at time t_k and converges to fixed point $\overline{\mu}$. Since we know $\|\mu^{M,t_k+1} - \mu^{M,t_k}\| \to 0$, we know that even the subsequence

starting at μ^{M,t_k+1} will converge to the same point. So, if we use $t = t_k$, the right hand side vanishes when $k \to \infty$, so

$$\langle V(\overline{\mu}), \mu - \overline{\mu} \rangle \geq 0 \tag{50}$$

which is the solution of $\text{VI}(V, \Pi^{\sigma_{\min}})$. Applying lemma 5 on $\overline{\pi}$, the sequence $\|\overline{\mu} - \mu^{M,t}\|$ is non-increasing and it converges to 0 along wtih subsequence starting at t_k. Hence $\|\overline{\mu} - \mu^{M,t}\| \to 0$ and the whole sequence converges to $\overline{\mu}$. □

Lemma 6. *Consider a function f, which has continuous second derivative with $\sup |f''| \leq M_f$, then $|\mathbb{E}_{x\sim\mathcal{N}(\mu,\sigma^2)}[f(x)] - f(\mu)| \leq \frac{M_f\sigma^2}{2}$.*

Proof. Let $Z \sim \mathcal{N}(0,1)$ and z is a single realization of Z, then it holds

$$f(\mu + \sigma z) = f(\mu) + \sigma z f'(\mu) + \frac{1}{2}\sigma^2 z^2 f''(\xi_z) \tag{51}$$

where ξ_z is a point between μ and $\mu + \sigma z$. Expectation of this is

$$\mathbb{E}_{x\sim\mathcal{N}(\mu,\sigma^2)}(f(x)) = \mathbb{E}(f(\mu + \sigma Z)) = f(\mu) + \sigma f'(\mu)\mathbb{E}(Z) + \frac{1}{2}\sigma^2\mathbb{E}(Z^2 f''(\xi_Z)) \tag{52}$$

Because Z is Gaussian with mean 0, then $\mathbb{E}(Z) = 0$. For the second term it holds that

$$|\frac{1}{2}\sigma^2\mathbb{E}(Z^2 f''(\xi_Z))| \leq \frac{1}{2}\sigma^2\mathbb{E}(Z^2|f''(\xi_Z)|) \leq \frac{1}{2}\sigma^2 M_f\mathbb{E}(Z^2) = \frac{1}{2}\sigma^2 M_f. \tag{53}$$

So Putting this together we get

$$|\mathbb{E}_{x\sim\mathcal{N}(\mu,\sigma^2)}[f(x)] - f(\mu)| \leq |f(\mu) + \frac{1}{2}\sigma^2 M_f - f(\mu)| = \frac{1}{2}\sigma^2 M_f \tag{54}$$

□

Proposition 3. *In a convex-concave game, the exploitability for each strategy $\pi \in \Pi^{\sigma_{\min}}$ is at most*

$$\mathcal{E}(\pi) \leq \max_{\pi_1' \in \Pi_1^{\sigma_{\min}}} U(\pi_1', \pi_2) - \min_{\pi_2' \in \Pi_2^{\sigma_{\min}}} U(\pi_1, \pi_2') + \frac{1}{2}(M_1 + M_2)\sigma_{\min}^2 \tag{55}$$

Proof. Let $a_1^{\text{BR}} \in \arg\max_{a_1 \in \mathcal{A}_1} \mathbb{E}_{a_2\sim\pi_2} u(a_1, a_2)$. We can apply lemma 6 with $M_g = M_1$ to get

$$\mathbb{E}_{a_2\sim\pi_2} u(a_1^{\text{BR}}, a_2) \leq U(a_1^{\text{BR}}, \pi_2) + \frac{M_1\sigma_{\min}^2}{2} \leq \max_{\pi_1'} U(\pi_1', \pi_2) + \frac{M_1\sigma_{\min}^2}{2} \tag{56}$$

Symmetrically we could do the same process for the Player 2, to get the final bound. □

Theorem 2. *In a convex-concave game with payoff u that has continuous second derivative. Assume only strategies $\Pi^{\sigma_{\min}} \subseteq \Pi$, which use $\sigma = \sigma_{\min}$ and let $\sigma_{\min} > 0$, $\eta > L_V\sigma_{\min}^2$ and $M_i = \sup|\frac{\partial^2 u}{\partial a_i^2}|$. Let $\pi^{M,t+1} = \Psi(\pi^{M,t})$ be the exact solution for the fixed magnet at iteration t from arbitrary $\pi^{M,0} \in \Pi^{\sigma_{\min}}$. Then the sequence of $\pi^{M,t}$ converges to a fixed solution π^* with exploitability at most*

$$\mathcal{E}(\pi^*) \leq \frac{M_1 + M_2}{2}\sigma_{\min}^2 \tag{3}$$

Proof. The proof follows directly from the theorem 3, which shows that the sequence converges to a solution of $\text{VI}(V, \Pi^{\sigma_{\min}})$ and from proposition 3, which bounds the exploitability of the strategy profile based on $\sigma_{\min}$. □

B GAMES

In this section, we provide rules for all the games used in the experiments.

B.1 ONE-SHOT GAMES

Continuous Matching Pennies Both players are choosing a real value from the range [-1, 1]. This game contains a pure Nash equilibrium at $a_1 = a_2 = 0$, but the gradient descent dynamics fail to converge to it.

$$u_1(a_1, a_2) = a_1 a_2 \tag{57}$$

Rotational game Both players are choosing a real vector from $[-1, 1]^2$ or $[-1, 1]^3$. Similarly to continuous matching pennies, this game contains a pure Nash equilibrium for $\mathbf{a}_1 = \mathbf{a}_2 = \mathbf{0}$ to which the gradient descent does not converge. This game shows that the result of Theorem 2 can hold even beyond convex-concave games. Also it shows that even if the theorem is stated for single-dimension Gaussian, it holds even in multi-dimensional case.

$$u_1(\mathbf{a}_1, \mathbf{a}_2) = 20(\mathbf{a}_1 + 0.3\mathbf{a}_1^3)^\top \mathbf{A}(\mathbf{a}_2 + 0.3\mathbf{a}_2^3) - \frac{1}{16}(\|\mathbf{a}_1\|^4 - \|\mathbf{a}_2\|^4) \tag{58}$$

where $\mathbf{A} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$ for 2D case and $\mathbf{A} = \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix}$

Circle game Both players are choosing a real value from the range [0, 1]. The game is parametrized by the number of harmonics; we have used $K = 3$. For any K, this game has a Nash equilibrium of size $K + 1$, with uniformly spread points on the interval.

$$u_1(a_1, a_2) = \sum_{k=1}^{K} 2^{-(k-1)} \sin(2\pi k(a_1 - a_2)) \tag{59}$$

Glicksberg-Gross game Both players are choosing a real value from the range [0, 1]. It contains a Nash equilbrium that is $\frac{4}{\pi} \cdot \arctan(\sqrt{t})$, for $t \in [0, 1]$. Again, the mixture of Gaussians cannot converge to this equilibrium with a finite number of Gaussians. Along with the circle game, these demonstrate how the mixture of Gaussians performs in games outside the scope of the algorithm.

$$u_i(a_1, a_2) = \frac{(1 + a_1)(1 + a_2)(1 - a_1 a_2)}{(1 + a_1 a_2)^2} \tag{60}$$

Two-point game Both players are choosing a real value from the range [-2, 2]. This game has a Nash equilibrium in which both players mix between 30% playing -1 and 70% playing 1. This is the best-case scenario for the mixture of Gaussians, as it can approximate these 2 pure strategies very well.

$$\begin{aligned} u_i(a_1, a_2) = &\exp\Big(-\frac{(a_1 + 1)^2}{0.02}\Big) + \exp\Big(-\frac{(a_1 - 1)^2}{0.02}\Big) \\ &- \exp\Big(-\frac{(a_2 + 1)^2}{0.02}\Big) - \exp\Big(-\frac{(a_2 - 1)^2}{0.02}\Big) \\ &+ (a_1 - 0.4)(a_2 - 0.4) \end{aligned} \tag{61}$$

B.2 SEQUENTIAL GAMES

Continuous Kuhn Poker The game is played with the deck of 3 cards, Jack (J), Queen (Q), and King (K). The game starts with each player receiving 1 card and paying an ante of 1. Then player 1 either checks or bets between 0.25 and 2. If player 1 checks, the other player can either check or bet again. After the bet, the other player can either fold and lose their ante or call the bet. If neither player folds, both players reveal their cards, and the player with the higher value wins. The game uses the following order of cards: K > Q > J.

Continuous Leduc Hold'em The game is played with the deck of 6 cards, which contains 2 Jacks (J), 2 Queens (Q), and 2 Kings (K). The game starts with each player receiving 1 card and placing an ante of 1. The first player then either checks or bets between 0.25 and 2. After that, the players alternate by either folding, which results in a loss, calling the opponent's bet, or raising between

0.25 and 2. The game consists of 2 rounds, and in each, there can be at most 2 raises. The round ends when one of the players calls the other one, and between the first and second rounds, the public card is revealed. If neither player folds, then at the end of round 2, both show their hands, and either the player who matches the public card wins or the player with the higher-value card, with K $>$ Q $>$ J.

Sequential Blotto N A sequential game over N turns, where players are assigning resources simultaneously to the same base. The player who uses more resources "wins" that base. At the end of the turn, only the players who won the base are revealed, not how many resources they used. The player who wins more bases in the N turns wins the game.

Heads-up No-limit Texas Hold'em The game is played with the standard 52-card deck. Each player starts with 400 chips and receives 2 cards at the beginning of the game. The first player (small blind) makes an ante of 1 chip, while the second player (big blind) makes an ante of 2 chips. The game is played in 4 rounds. After the first round, 3 public cards are revealed; after the second round, 1 public card is revealed; and after the third round, the last public card is revealed. In the round, players alternate in betting until one of them calls. Either player can bet all their chips at any time, forcing the opponent to call or fold. At the end of the game, the players reveal their private cards, and the player with the highest combined value of their private and public cards wins. Even if Texas Hold'em has discrete bets, we treat them as continuous. So the algorithm samples a real-valued bet, and we round it to the nearest integer, which is then used in the game.

C Experimental details

In this section, we provide the hyperparameters for all algorithms across all experiments, along with the computational resources used for each experiment.

C.1 Proximal policy optimization

We use Proximal Policy Optimization (PPO) as one of the baselines, as an algorithm to compute the best response for evaluation, and as an oracle in PSRO and NFSP. We use Gaussian reparametrization for the PPO, where we train just a single Gaussian head through the following loss function

$$\begin{aligned}\rho_{\text{PPO}} &= \min\Big(\frac{\pi_\theta(a_k^t|s^t,\mathcal{N}_k^t)}{\pi_{\text{old}}(a_k^t|s^t,\mathcal{N}_k^t)}A(s^t,a_k^t), \text{clip}\big(\frac{\pi_\theta(a_k^t|s^t,\mathcal{N}_k^t)}{\pi_{\text{old}}(a_k^t|s^t,\mathcal{N}_k^t)}, 1-\epsilon_{\text{PPO}}, 1+\epsilon_{\text{PPO}}\big)A(s^t,a_k^t)\Big)\\ \mathcal{L}_{\text{PPO}} &= -\mathbb{E}_t\rho_{\mathcal{N}} - \mathcal{H}(\mathcal{N}_k^t)\end{aligned} \tag{62}$$

To train the critic, we use Generalized Advantage Estimate (GAE) (Schulman et al., 2015). Due to the similarity of PPO to the mixture of Gaussians, we include the hyperparameters in Table 1

C.2 Magnetic Mirror Descent

Magnetic Mirror Descent is a modification to discrete PPO that introduces the KL divergence between the current strategy and the magnet strategy into the loss. In our mixture-of-Gaussians reparametrization, we use the MMD loss to train the categorical distribution $\mathcal{L}_w$. As a baseline, we use MMD also as a solver for the discretized game. The discrete bins we use are uniformly spread through the action space. All the MMD hyperparameters are included in Table 1

C.3 Neural fictitious self-play

Neural fictitious self-play stores two strategy networks at each time, a BR strategy and an average strategy. The algorithm works in two alternating phases: in the first, it trains the best response against every strategy; then, in the second, it adds some trajectories sampled by the BR into the replay buffer, which is then used to train the average strategy. Compared to the vanilla NFSP, we use PPO to compute the best response and not DQN. We provide hyperparameters in Table 2

	One-shot games			Sequential games			
	PPO	**MMD**	**MMPO**	**PPO**	**MMD**	**MMPO**	**BR-PPO**
PPO epochs	2	2	2	1	1	1	1
Entropy weight	0.05	0.05	0.05	0.05	0.02	0.02	0.01
Magnet weight η	-	0.2	0.2	-	0.2	0.2	-
Magnet update	-	500	500	-	1000	1000	-
Minimal std σ_{min}	-	-	0.001	-	-	0.1	0.001
Components K	1	-	*	1	-	4	1
Exploration ϵ	0.0	0.0	0.0	0.2	0.2	0.2	0.0
Value loss weight	0.5	0.5	0.5	0.5	0.5	0.5	0.5
GAE lambda λ	-	-	-	0.95	0.95	0.95	0.95
V-trace clipping $\overline{\rho}$	-	-	-	2	2	2	2
V-trace clipping $\overline{c}$	-	-	-	1	1	1	1
Hidden sizes	(64,64)	(64,64)	(64,64)	(64,64)	(64,64)	(64,64)	(64,64)
Activation	GELU	GELU	GELU	GELU	GELU	GELU	GELU
Normalization	RMS	RMS	RMS	RMS	RMS	RMS	RMS
Optimizer	Adam	Adam	Adam	Adam	Adam	Adam	Adam
Learning Rate α	10^{-3}	10^{-3}	10^{-3}	10^{-3}	10^{-3}	10^{-3}	$3 \cdot 10^{-3}$
Max grad norm	0.5	0.5	0.5	0.5	0.5	0.5	0.5
Target update τ	10^{-3}	10^{-3}	10^{-3}	10^{-3}	10^{-3}	10^{-3}	10^{-3}
Batch size	256	256	256	512	512	512	512

Table 1: Hyperparameters used in experiments for different policy-gradient algorithms and also for computing the approximate exploitability. The amount of components in one-shot games depend on game, in matching pennies it is 1, and in other games it is 4.

	NFSP	**PSRO**	**NFSP**	**PSRO**
BR train steps	10^3	10^3	10^4	10^4
BR batch size	64	64	64	64
PSRO payoff samples	-	256	-	20000
NFSP average steps	400	-	1000	-
Reservoir capacity	10^6	-	10^6	-
NFSP reservoir batch	256	-	256	-
NFSP anticipatory η	0.1	-	0.1	-

Table 2: Hyperparameters used in experiments for different best-response based algorithms. Training of the best response uses PPO hyperparameters from Table 1, besides a different batch size.

C.4 Policy space response oracles

The training of Policy space response oracles is also split into two phases. In the first phase, both players train best response against the meta strategy. Then this BR is added to each player's portfolio, and the expected utility of these strategies against all other strategies in the portfolio is computed. Then the resulting matrix game is solved, and the solution is used as the new meta-strategy, to which both players compute their best responses. Unlike the NFSP, the number of strategies PSRO stores grows with the training. The hyperparameters of PSRO are in Table 2

C.5 Randomized policy networks

Unlike other techniques, randomized policy networks do not explicitly produce the strategy. Instead, they store it implicitly in the network's weights. The network takes random noise at the input and outputs the sampled action. This makes RPN similar to diffusion models, which perform a single step. Unlike diffusion models, which use supervised training, the randomized policy networks are used in reinforcement learning. Since the strategy is not parametrized in any way, the network can learn an arbitrary probability distribution given its architecture. We provide all hyperparameters in Table 3

Optimizer	Optimistic GD
Optimism	0.333
Learning rate	10^{-3}
Max grad norm	5
Batch size	256
Noise dimension	16
Activation	Mish
Normalization	RMS
Extragradient step	0.01

Table 3: Randomized policy networks hyperparameters used in one-shot games

C.6 Local best response

Local best response (LBR) is a Poker-specific technique that estimates a lower bound on the exploitability of the strategy π. Let us describe it from the perspective of evaluating Player 1's strategy, so the local best response is Player 2. When Player 2 is about to make a decision, it computes the exact probabilities for each hand of Player 1. The local best response uses action abstraction; we have used fold, call, pot, all-in abstraction. For each action, local best-response estimates its value by assuming both players would only check from this point to the end, and then pick the best one. LBR is more effective if it lets the opponent make more mistakes and therefore, we use the setting where it always checks and calls in the first two betting rounds of the game and only estimates action values and plays the best of them in turn and river.

C.7 Computational resources

All of our experiments used Python and JAX as a machine learning framework (Bradbury et al., 2018). Every experiment except Heads-up no limit Texas hold'em used a single CPU core of AMD EPYC 7543 with 16 GB of RAM and 16Gb of memory. In Figures 2 to 4, 6, 10 and 11 we report means and 95% confidence intervals computed from 5 different random seeds. Experiment in Figure 1 do not use an any randomness.

Convergence of the regularized Gaussian Head All of the experiments in this section took less than 4 hours using when ran sequentially.

One-shot games Single training run for each game took less than 10 minutes. We have used 5 baselines, 5 seeds, 4 games. Altogether this makes the experiment run 16 hours.

Sequential games Single training run for each game took less than 4 hours in fig. 4. However, the approximate exploitability computation took up to 12 hours. In total for each of the 4 baselines, 5 seeds and 4 games, the experiment took $4 \cdot 4 \cdot 5 + 3 \cdot (4 + 12) \cdot 5 \cdot 4 = 1040$ hours.

Capacity experiments For one-shot games we have used the same limit, for sequential games, we have used longer time-frame for training, when number of components and bins was 16 or greater. Similarly for those runs the best response took up to 24 hours. In total this experiment took $64 + 11 \cdot 5 \cdot 4 + 7 \cdot 5 \cdot 3 \cdot (4 + 12) + 4 \cdot 5 \cdot 3 \cdot (12 + 24) = 4124$ hours.

Heads-up no limit Texas hold'em This is the only experiment where we have used GPUs, and we have trained the strategy for 4 days on H200. Then for evaluation, each checkpoint comparison against Slumbot took roughly 120 CPU-hours, while the LBR computation took roughly 12 CPU hours per checkpoint. Altogether this gives 8976 CPU-hours for the final evalutaion.

D Additional experiments

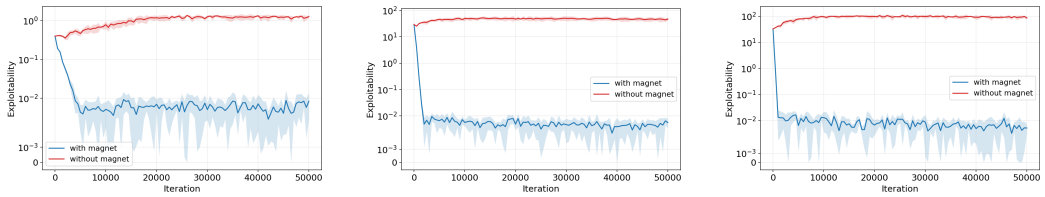
We have run two additional experiments to evaluate different aspects of the MMPO. First, we show that the MMPO converges in the same games as in section 4.1 when the neural network does not parametrize the strategy but the gradients are estimated from the samples. Second, we show how the

strategy's exploitability varies with the number of Gaussians in the model. A different perspective on MMPO is that it trains the abstraction during training. We also compare MMPO with uniform discretization of the action space using the same number of components.

D.1 Regularized Gaussian head

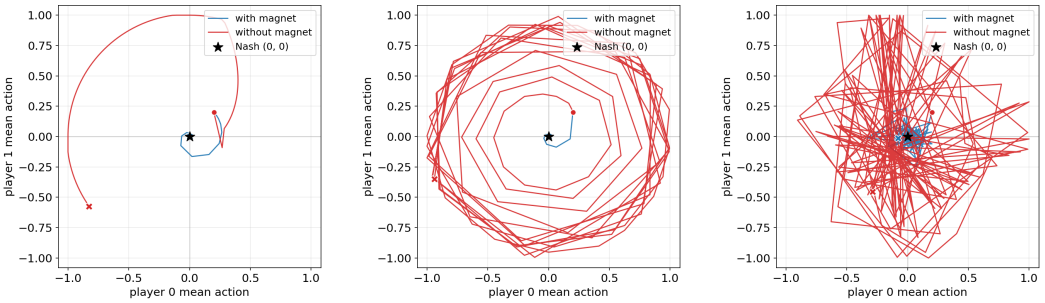
We have run the same experiment as in Section 4.1, but now we do not store the strategy in the neural network, but we use samples to estimate the gradient. This serves as a middle step between the Figure 1 and Figure 2. We have used a learning rate of 0.05 and a batch size of 256. The results of this experiment are in Figure 6. Even these results confirm that without a magnet, the gradient descent diverges to the most exploitable strategies. With the magnet, the strategies do not converge arbitrarily close to the Nash equilibrium because convergence is limited by noise in gradient estimation. Still, in all games, the exploitability is roughly 10^{-2}, which is several orders of magnitude better than without a magnet.

We also show how the strategies in continuous matching pennies evolve during training across the three training regimes in Figure 7. It also confirms that the strategies gradually diverge without the magnet regularization.



(a) Continuous matching pennies (b) Rotational game, dim=2 (c) Rotational game, dim=3

Figure 6: Exploitability in one-shot games with pure Nash equilibria using gradient descent with or without magnet, using only the gradient estimate.

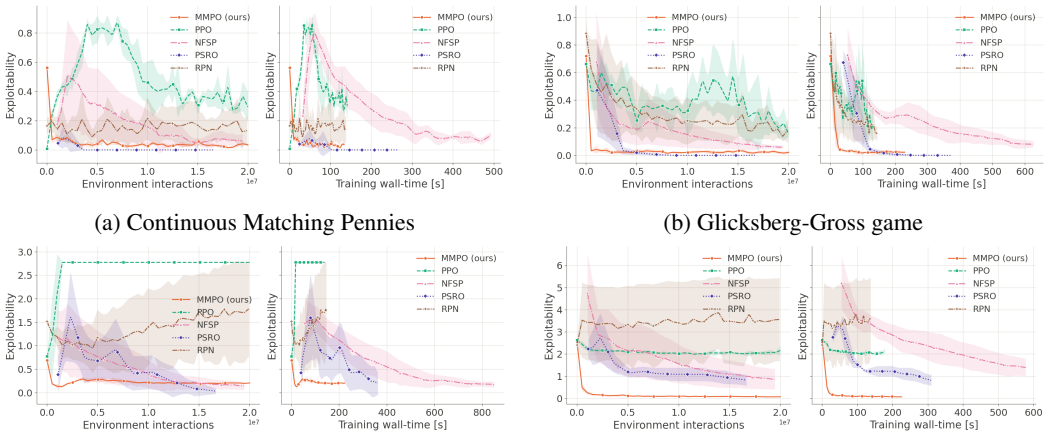


(a) Exact gradient (b) Estimated gradient (c) Estimated gradient, neural

Figure 7: The evolution of a strategy in continuous matching pennies during training for each of the experimental setting

D.2 Wall-time comparison

In Figures 3 and 4 we compared the algorithms in terms of environment interactions. Here, we also provide the comparison of the same runs, but when using wall-time for comparison. The results are in Figures 8 and 9. Both the environment interactions and wall-time agree with the relative performance of the algorithms, which show that MMPO is as fast as PPO, but converges much quicker than NFSP and PSRO.



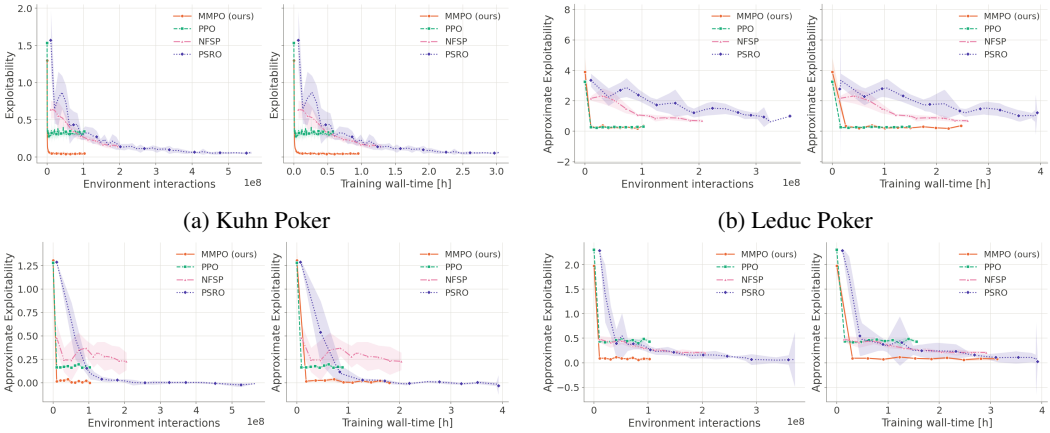
(a) Continuous Matching Pennies

(b) Glicksberg-Gross game

(c) Circle game

(d) Two-point game

Figure 8: Exploitability with 95% confidence intervals of different baselines in one-shot games



(a) Kuhn Poker

(b) Leduc Poker

(c) Sequential Blotto 3

(d) Sequential Blotto 5

Figure 9: Approximate exploitability with 95% confidence intervals of different baselines in sequential games

D.3 Effect of the discretization and amount of components in the mixture

The strategy space of MMPO is limited by the number of Gaussians used in the mixture. We have run another experiment that shows how the final exploitability changes when we use different numbers of components. We also compared it with the alternative approach that uniformly discretizes the action space and then uses MMD in the discretized game. We have used the same games as in sections 4.2 and 4.3.

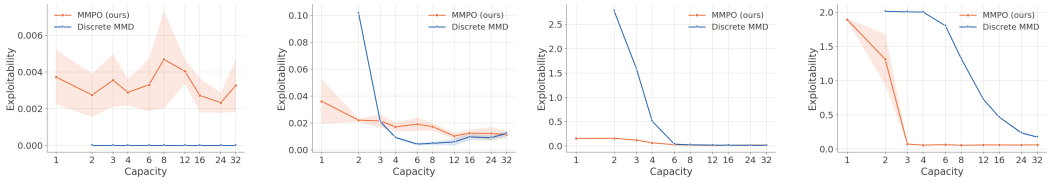
The results for one-shot games are in Figure 10. The discretization works well in matching pennies, because even if the game contains a pure Nash equilibrium, any strategy that has a mean of 0 is also a Nash equilibrium, so a uniform discretization that is symmetrical around 0 can always find a Nash equilibrium. In contrast, the noise from the Gaussian sampling in MMPO limits how close to equilibrium the MMPO can converge, regardless of the number of components.

In the Glicksberg-Gross game, the discretized version performs the best with 6 bins, and adding more bins degrades the performance. Similar behavior can be observed in some sequential games in Figure 11. This happens because of entropy regularization, which slightly skews the distribution toward a uniform distribution. This highlights another advantage of the MMPO, because even if the categorical distribution is skewed toward playing all actions with some nonzero probability, the

Gaussian can shift to a place where this probability is better utilized. This means that MMPO is less sensitive to the entropy setting.

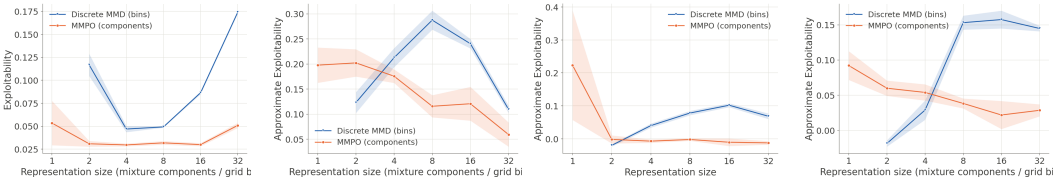
A two-point game presents the worst-case scenario for the discretization: when the two pure actions of the equilibrium are not in the discretization, the discretized version will never converge arbitrarily close, as we can see when we increase the number of discrete bins. On the other hand, the MMPO can reliably converge with three Gaussian components. Two components should be sufficient for convergence, but MMPO is sensitive to their initialization. Specifically, in this game where the two pure actions are -1 and 1, if all components are initialized greater than 1, then only very strong exploration would be able to move one of the components to -1. This is not specific to our method, but to any method that uses gradient descent to move pure actions. In practice, it may be possible to detect these collapsed actions, reinitialize the μ and σ for one of them, and resume training, but we decided to avoid this to keep the setting as simple as possible.

Even if adding more components broadens the set of strategies the algorithm can converge to, it also slows training, as more samples are required to learn more actions. As such, the optimal amount of components can differ based on the available budget.



(a) Matching Pennies (b) Glicksberg-Gross (c) Circle game (d) Two-point game

Figure 10: Final exploitability of MMPO and MMD on discretized action space based on the amount of Gaussians or discrete components used.



(a) Kuhn Poker (b) Leduc Poker (c) Blotto 3 (d) Blotto 5

Figure 11: Final exploitability of MMPO and MMD on discretized action space based on the amount of Gaussians or discrete components used.

E HEADS-UP NO LIMIT TEXAS HOLD'EM DETAILS

In this section, we detail several changes to the MMPO we made specifically for the Texas hold 'em experiment and include additional results from that experiment.

E.1 EXPERIMENTAL DETAIL

Action abstraction We used these discrete actions: Fold, Call/Check, and All-in. Then we treat the betting as a single continuous action. However, we do not use the raw bet-size; the bet action is a fraction of the pot, which was used in prior Poker AIs Brown & Sandholm (2018); Moravčík et al. (2017). We use 5 Gaussian components in this continuous action, and we do not initialize them randomly; instead, we initialize them so that their means cover the interval $[\frac{1}{3}, 3]$ geometrically. If the mean of the action is beyond the legal betting range, we clip it. Besides that, we are not predicting the mean μ directly. We predict $\log(\mu)$, which we then exponentiate. In the game, bets are discrete, so after we sample the real value, we round the bet to the nearest legal amount. During training, we use this rounded bet rather than the sampled one.

PPO epochs	1
Categorical magnet weight η_w	0.2
Gaussian magnet weight $\eta_{\mathcal{N}}$	0.02
Magnet update	20000
Minimal std σ_{min}	0.05
Components K	5
Exploration ϵ	0.01
Value loss weight	1.0
GAE lambda λ	0.95
V-trace clipping $\overline{\rho}$	1
V-trace clipping $\overline{c}$	1
Hidden sizes	(1024,1024,1024,1024)
Activation	GELU
Normalization	Layer
Optimizer	Adamw
Weight decay	0.002
Max grad norm	1.0
Target update τ	$4 \cdot 10^{-4}$
Batch size	4096
Max game length	40

Table 4: Hyperparameters used in training of heads-up no limit Texas hold'em

Action Clipping We apply a trick similar to Neural replicator dynamics that stops learning certain discrete actions when their logits are too far apart (Hennes et al., 2020). Specifically, when an action logit is β smaller than the highest logit, only the gradients that will increase the probability of such actions are used, and vice versa when the logit is higher than the smallest logit. We applied this trick because in one of our runs, the logits were getting too small, which made them unrecoverable when the action turned out to be valuable. We also use sigmoid for the $\log(\sigma)$ rescaled to values between [-3, -0.7]

Exploration Early training runs showed that later checkpoints improved their performance against older checkpoints, but their performance against Slumbot was degrading substantially. We found out that it was because some bets were not being explored, and the strategy for those bets was degrading. We have made several changes to avoid this problem. The neural replicator dynamics' maximal-difference logits are one such change, and the second change is that, with 1%, we sample the random categorical action. We use an importance-sampling correction to ensure that all estimates are unbiased. In imperfect-information games, it is not enough to apply an importance-sampling correction to the future values in V-trace; it is also necessary to correct past decisions. To avoid the correction becoming too high or too low, we clip it to the range [0.1, 10]. We do not use any exploration for the Gaussian head, besides the samples from the Gaussian itself.

Hyperparameter specifics We use two training schedules. First is cosine decay of the learning rate from 10^{-3} with $T = 1.5 \cdot 10^6$ and $\alpha = 0.06$. We also used an exponential decay of entropy from 0.03 to 0.003 over $2 \cdot 10^6$ iterations. We also did not use the same magnet and entropy regularization coefficients for the Gaussian and categorical heads. In the categorical head, we use the KL divergence coefficient $\eta_w = 0.2$ and the entropy coefficient α with exponential decay. For Gaussian, we used the KL divergence coefficient $\eta_{\mathcal{N}} = 0.02$, and scaled the entropy by 0.5. We decreased these coefficients for the Gaussian head because, in our early experiments, the Gaussian means were not learning due to the magnet loss outweighing the advantage gradient. All other hyperparameters are in Table 4

E.2 Additional results

In Figures 12 and 13 we show the initial bets of small blind and big blind after small blind checked, respectively. The probabilities are averages across all possible suits for that combination. The values in the main diagonal correspond to the player holding a pair; the lower-left triangle corresponds to the player holding 2 offsuit cards; and the upper-right triangle corresponds to the player holding

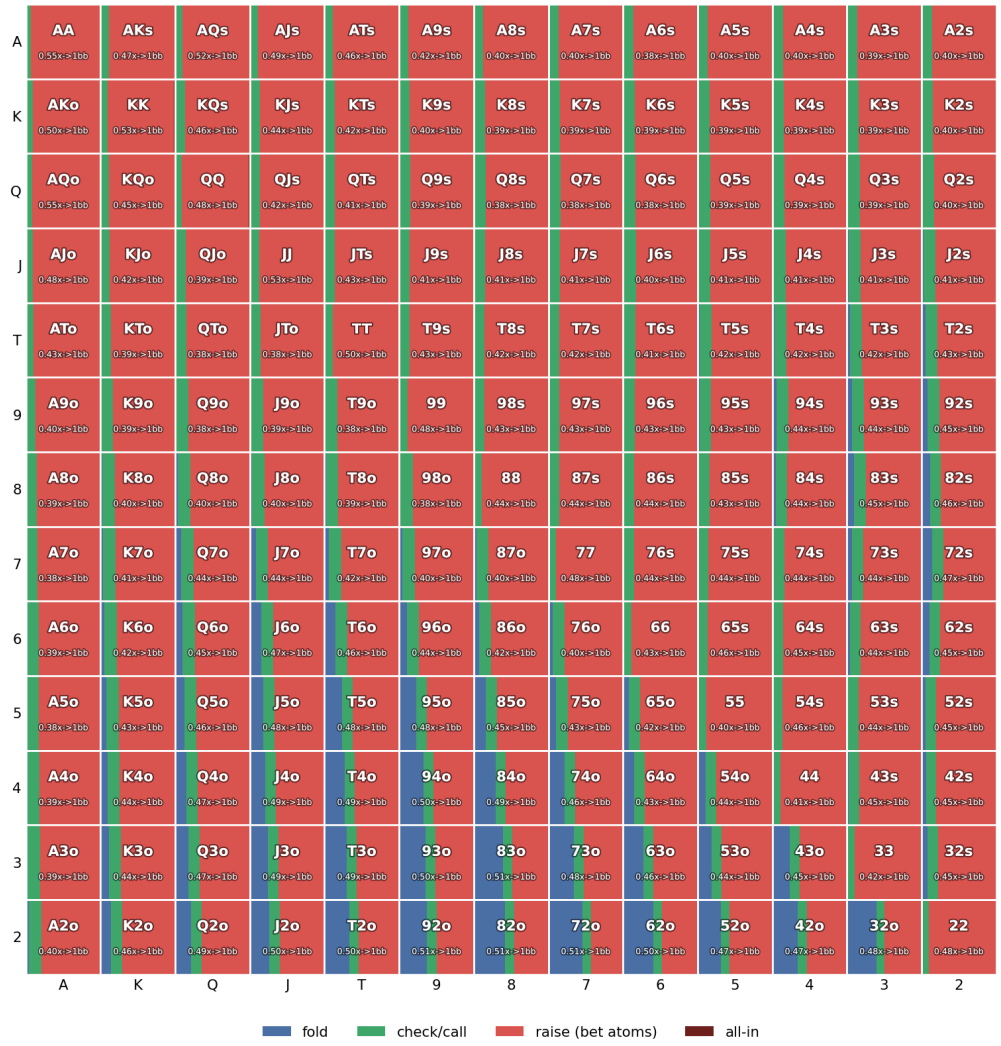


Figure 12: Small blind initial bet with different cards averaged over all suits.

cards in the same suit. Interestingly, these two tables differ from the tables provided by the authors of the GTO wizard AI (Provost et al., 2026). For the Small blind bet, the GTO wizard's bot never folds with suited cards, and with off-suited cards, it folds much more aggressively. And for the Big blind bet, MMPO never folds, whereas GTO wizard folds quite often, especially when holding 2 and 3 offsuit.

In Figure 14, we show the average win rate against older checkpoints and weighted outcomes against Slumbot for checkpoints produced during training. Even if the Slumbot performance does not seem to improve substantially in later iterations, the network itself is still improving against older checkpoints. This highlights the strategy non-transitivity in imperfect information games. The improvement in gains for later checkpoints seems to be mainly from improving how much the player loses at showdown. On the other hand, rewards from folding remain roughly the same in later stages.

In Figures 15 and 16, we show all the bets as fractions of the pot and the pot sizes from self-play of the final checkpoint, which shows that, unlike abstraction bots, the MMPO does not have blind spots on some decisions, which would be caused by static discretization.

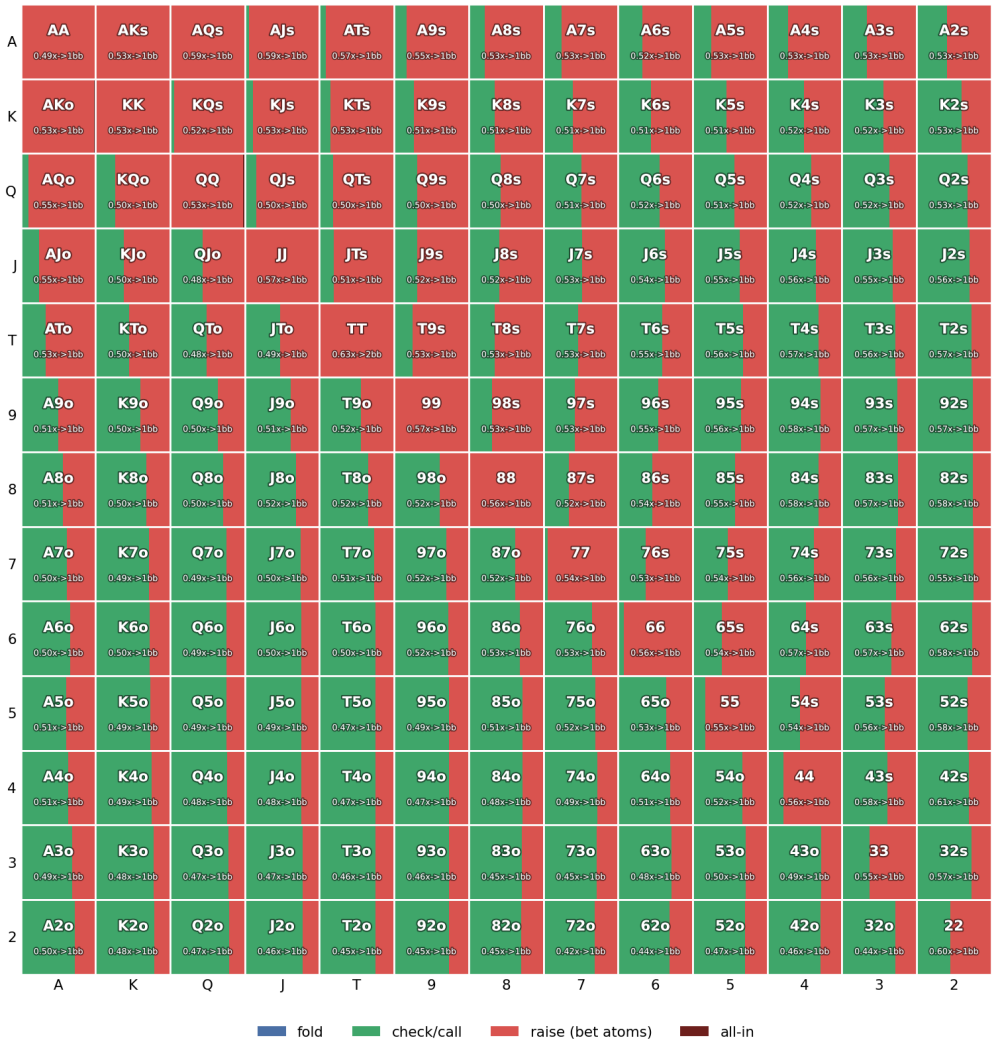


Figure 13: Big blind initial bet after check from the small blind with different cards averaged over all suits.

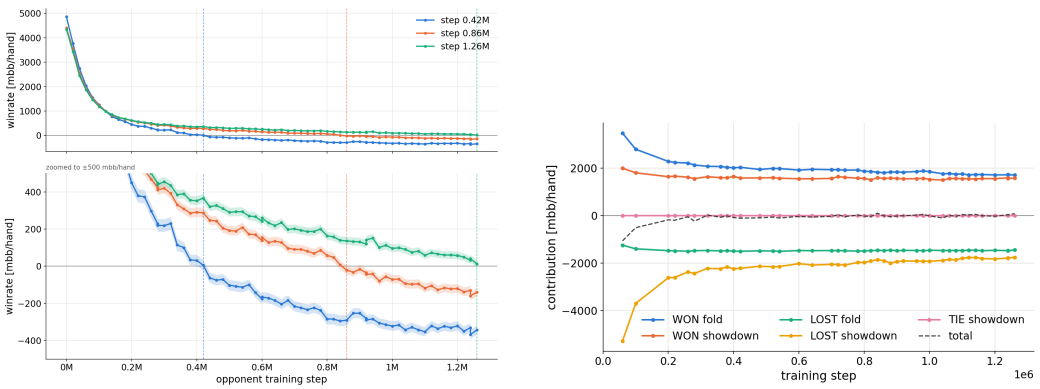


Figure 14: (left) Average win-rate of different checkpoints against all other checkpoints. (right) Weighted outcomes based on different results of the hand against Slumbot.

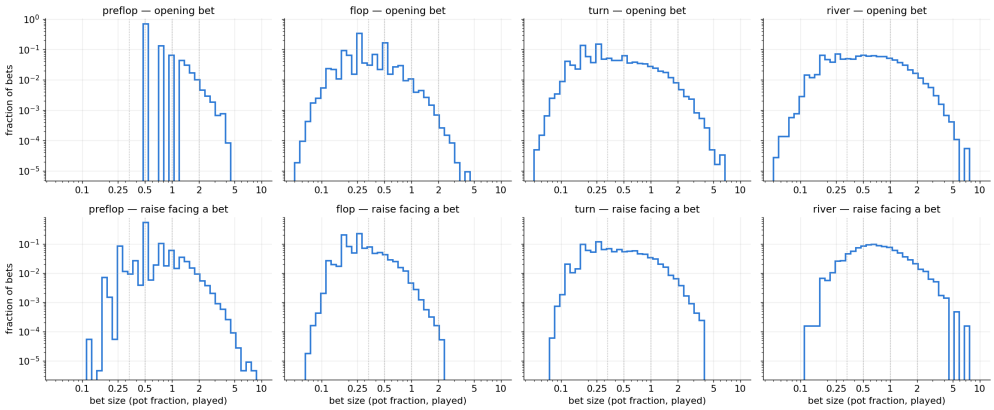


Figure 15: All bets made by the final trained checkpoint in self-play over million hands

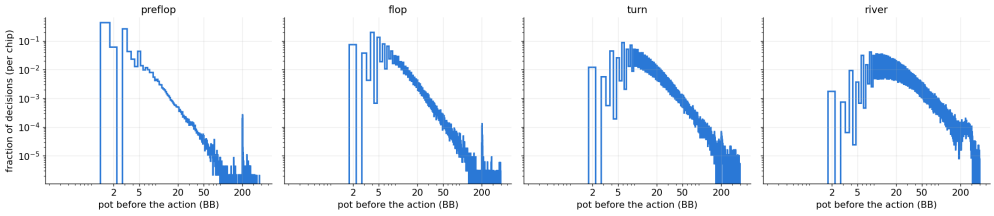


Figure 16: All pots, which were encountered by the final trained checkpoint in self-play over million hands